

# POLITECNICO DI TORINO

Master's Degree in ICT For Smart Societies



Master's Degree Thesis

## Frontend Module for the Management of Museums and Events

Supervisors

Prof. Renato FERRERO

Candidate

OBAID UR REHMAN

Academic Year 2023/2024

# Acknowledgements

At the end of this academic journey, I would like to thank my family, who has always been by my side, my friends, the colleagues I met here at university and all those who stood by me. Thanks also go to the polytechnic, to the professors who taught me so much, to my supervisor for the opportunity given to me.

## **Abstract**

This project aims to develop a SaaS-compliant frontend module to enhance the management and user experience of museums and events within an existing platform. This process involves needs analysis, interactive UI design, frontend module development using modern technologies, integration with the SaaS platform, robust authorization management, implementation of advanced functionalities, testing, and comprehensive documentation. An AI Chat, powered by GPT-3.5 turbo fine-tuning model will be created to increase user satisfaction and engagement. The expected outcome is a seamlessly integrated frontend module that simplifies operational processes and provides an engaging and user-friendly experience for museums and events, thus improving efficiency and user satisfaction. Ongoing support, including post-implementation assistance and regular updates, will be prioritized to ensure sustained compatibility and continual enhancement within the evolving SaaS platform environment.

# Table of Contents

<b>List of Figures</b>	IV
<b>1 Introduction</b>	1
1.1 Market . . . . .	2
1.2 List of Services . . . . .	3
1.3 New Services . . . . .	4
1.4 Related Work . . . . .	5
1.5 Methodology . . . . .	10
<b>2 Planning</b>	12
2.1 Goals . . . . .	12
2.2 Challenges . . . . .	13
<b>3 Analysis</b>	17
3.1 Direct Communication . . . . .	19
3.2 Information Gathering . . . . .	19
3.3 Needs and Goals Analysis . . . . .	20
3.4 Reporting and Documentation . . . . .	20
<b>4 Design</b>	22
4.1 Design Phase Purpose . . . . .	22
4.2 Cross-Device Optimization . . . . .	24
4.2.1 Compatibility Considerations . . . . .	29
<b>5 Enabling Tools/Technologies</b>	30
5.1 Front-End . . . . .	30
5.1.1 JavaScript . . . . .	31
5.1.2 Django . . . . .	32
5.1.3 Rosetta . . . . .	34
5.2 Back-End . . . . .	35
5.2.1 Node.js . . . . .	36

5.2.2	GitHub	36
5.2.3	FireBase	38
5.2.4	MongoDB	40
5.3	Other Tools	41
5.3.1	Visual Studio Debugger	41
5.3.2	Reverse Proxy NGNIX	42
5.3.3	Chrome Debugger	43
5.3.4	Flutter	43
<b>6</b>	<b>Implementation</b>	<b>45</b>
6.1	Frontend	46
6.1.1	Platform/Server	46
6.1.2	Frontend website for End-User	46
6.1.3	Frontend website for Operator	47
6.1.4	Help-Desk	47
6.2	Application Functionalities (Frontend)	47
6.2.1	Main/Home page	47
6.2.2	Operations/Functions	49
6.2.3	Login/Sign-up	49
6.2.4	Google-Login	52
6.2.5	Cart-Checkout	54
6.2.6	Deleting-Item	57
6.2.7	Payment	59
6.2.8	Pay through QR-Code	60
6.2.9	ChatAI	61
6.2.10	Analytics	65
6.2.11	Scanner	67
6.2.12	Detail	71
6.2.13	DESK	75
6.2.14	Flutter	76
<b>7</b>	<b>Testing</b>	<b>78</b>
7.1	Unit Testing	78
7.2	Acceptance Testing	79
7.3	Regression Testing	79
7.4	System	79
7.4.1	Performance Testing	81
7.4.2	Security Testing	83
7.5	Usability Testing	84

<b>8</b>	<b>Conclusion</b>	89
8.1	Technical Development . . . . .	89
8.2	Functional Development . . . . .	90
8.3	User Development . . . . .	90
8.4	Future Development . . . . .	91
<b>9</b>	<b>Bibliography</b>	92

# List of Figures

1.1	SDLC . . . . .	11
2.1	Gantt Chart . . . . .	15
4.1	Work-Flow . . . . .	23
4.2	Design Compatibility . . . . .	26
4.3	Main Page . . . . .	27
4.4	AI Chat Design . . . . .	28
5.1	Rosetta Home . . . . .	35
5.2	Firebase Create User . . . . .	38
5.3	Firebase Anonymous User . . . . .	39
5.4	Chrome Debugger . . . . .	43
6.1	Login / Google-Login . . . . .	54
6.2	Cart Checkout . . . . .	57
6.3	Proceed to Pay / Delete All . . . . .	58
6.4	Payment through NEXI . . . . .	59
6.5	Pay with QR-Code . . . . .	61
6.6	Analytics / Collections and Accesses . . . . .	67
6.7	Scanner Page . . . . .	69
6.8	Validation . . . . .	71
6.9	Detail Page . . . . .	74
6.10	Desk Page . . . . .	76
7.1	System Testing . . . . .	80
7.2	Waterfall Chart . . . . .	81
7.3	Page Visualisation . . . . .	82
7.4	Security Testing . . . . .	84
7.5	Load Time . . . . .	86
7.6	Page Size . . . . .	87

# Chapter 1

## Introduction

Culturatela.com is an online booking system for cultural places. The goal is to develop a user-friendly ticketing frontend module for Culturatela.com, which is a comprehensive platform for effortlessly booking tickets to museums, concerts, and events across various municipalities, including Lucca, Avellino, Roma, and Ancona. These cultural hubs attract more than 1.5 million visitors every year. They are crowded places where people come to enjoy art, history, and entertainment. The booking system on Culturatela.com follows the rules set for Software as a Service (SaaS). This means it is built using the best methods for online services. It is reliable, easy to use, and can handle lots of people using it at once. This ensures that cultural places can use modern technology to serve visitors better. Culturatela.com will make it easier to manage and enhance the experience of museums and events. It achieves this by seamlessly integrating with the existing platform, providing efficient tools for both administration and user enjoyment. This special software is made for art and culture groups. It is the result of a lot of research and development, and its goal is to make managing cultural organizations easier. It uses advanced digital technology to be more efficient and save money. It is a flexible platform that can fit the needs of each organization. The software helps with planning events, managing people, selling tickets both at the venue and online, and handling communication and marketing. It also follows guidelines from the Ministry of Cultural Heritage<sup>1</sup> to digitize cultural information.

---

<sup>1</sup><https://docs.italia.it/italia/designers-italia/design-musei-docs/it/versione-corrente/index.html>



## 1.1 Market

Culturatela.com focuses on serving places with cultural importance like museums, theaters, exhibitions, and events. In Italy alone, there are more than 500 civic museums. The worldwide market for museums, historical sites, zoos, and parks increased from \$60.60 billion in 2022 to \$96.99 billion in 2023, with a growth rate of 60.1% [1]. However, the war between Russia and Ukraine has affected global economic recovery from the COVID-19 pandemic, leading to economic challenges. Despite this, the market for cultural places is expected to keep growing and reach \$119.33 billion in 2027, with a growth rate of 5.3%. [2] [3]

The cultural market in Italy is getting bigger, and digital technologies play a big part in this. People are more interested in culture, and there is a lot of promotion and appreciation for Italian art. New technologies and better tourist experiences also help it grow.

Other services on the market are not as flexible and smart as Culturatela.com, like the MidTicket and ticketing system, VivaTicket, TicketOne, etc, may offer ticketing solutions, but they often fall short in terms of flexibility and smart functionality compared to Culturatela.com.

TicketOne does not allow autonomous management of event slots, timings, or prices for the events it oversees. This restricts the immediate efficiency of its client's operations. Instead, TicketOne offers additional assistance and customer care packages, essentially requiring clients to purchase these to make any adjustments to their events. VivaTicket shares very similar characteristics with TicketOne. Mid Ticket is a smaller company compared to the first two, with a stronger orientation towards the cultural sector. However, it lacks a precise focus, as it also manages sporting events.

Culturatela.com stands out for its customizable and intelligent features, empowering event organizers to create tailored ticketing experiences that drive engagement and success. Also, Culturatela.com provides ticketing software that lets both public administrations (like local governments) and private individuals use secure payment services like PagoPA and NEXI, some of our competitors do not use PogaPA they only use NEXI. It wants to bring all the functionalities of museums in Italy onto one platform, making them work together. This way, Culturatela.com makes all Italian cultural places easy to access and buy tickets for, and it offers new digital services to engage visitors. This makes it very easy for customers to find and buy tickets for any museum they want to visit. Culturatela.com can attract more customers and grow in the market, even if there are other companies in the same market, because it provides museum administration with the big data and mail

database of previous customers in order to send emails about new events.

## 1.2 List of Services

Here is a list of services that explain the advantages of the booking system.

- **Booking System**

It helps in planning events, selling tickets, and knowing how many tickets are sold, whether at the location or online. It is like having everything organized so that managing tickets is simple and clear, whether people buy them in person or on the Internet.

- **Information Collection**

It also helps us look at information about the people who attend events so Culturatela.com can understand them better. This way, Culturatela.com can make smarter plans for advertising and promoting the events.

- **Automation**

It helps operators and museum administrators by doing many tasks automatically, which makes their daily work easier and more efficient. It also saves time and lowers the risk of making errors during the process. Prior to the introduction of the booking system, some museums relied on Excel sheets to manage booking data, like the Museum of Pietro Micca<sup>2</sup>, one of the Culturatela.com customers. Culturatela.com can provide a modern solution by allowing museums to store their booking information securely on cloud servers. This not only ensures data integrity but also facilitates efficient management and accessibility.

- **Partnership**

It enables linking tickets from various sources, even if they have different features or qualities. Opens up the possibility of forming partnerships and working together with other places in the area.

- **Rapid Response**

Use of AI Chat, which can respond to user queries rapidly and provide the easiest way to fulfill customer needs. AI Chat is available 24 hours a day, regardless of time zone or business hours.

---

<sup>2</sup><https://www.museopietromicca.it/>

- **Availability**

24/7 availability, the booking system is available round the clock, seven days a week, without any interruptions. This means that visitors can make reservations at their convenience, contributing to a more flexible and user-friendly booking experience.

- **Time Efficient**

One of the primary benefits of the booking system is the significant reduction in queuing lines at museum counters. Visitors can seamlessly book their tickets in advance, saving time and enhancing their overall cultural experience.

- **Analysis**

The desired functionalities aim to enhance these existing services further. For instance, the analytics module may be expanded to include more advanced data visualization tools and predictive analytics capabilities. The graphs module could be improved to provide customizable charts and graphs for better data interpretation. Additionally, the sidebar module could be optimized for easier navigation and access to key features.

- **Ticket Validation**

The QR code scanner module introduces a new functionality that allows for seamless ticket validation using QR codes, streamlining entry processes and improving security. Meanwhile, the API integration enables communication between the website and backend systems, facilitating data exchange and synchronization.

By combining these features, Culturatela.com aims to revolutionize the way people engage with cultural events while providing museums with a reliable and modernized platform for managing their bookings.

## 1.3 New Services

The project objective is to develop a SaaS-compliant frontend module that enhances the management and user experience of museums and events within the existing platform. The process encompasses detailed needs analysis, intuitive UI design, frontend module development using modern technologies, integration with the SaaS platform, robust authorization management, implementation of advanced functionalities, thorough testing, and comprehensive documentation. The creation of AI Chat with the help of GPT 3.5 turbo finetuning modal, which will increase user satisfaction and engagement, aims to deliver a seamlessly integrated frontend module that not only simplifies operational processes but also offers an engaging

and user-friendly experience for museums and events, contributing to improved efficiency and user satisfaction. Ongoing support, including post-implementation assistance and regular updates, is prioritized to ensure sustained compatibility and continual enhancement within the evolving SaaS platform environment. The goal of the thesis is to add some new features to the software system. These changes will make this platform even better and more user-friendly. My goal is to make the software as good as possible. I am going to explain the main modules in detail in Chapter 6.

- New way to sell tickets with automatic machines, users can purchase the tickets on site with totem machines
- QR code for payment redirections
- QR code scanner for mobile phones
- Chat AI integration to page
- Online consultation with the app
- FAQ page
- New ticket layout
- Intro js to give the administrative users a guided tour of the new functionalities in our application. It provides step-by-step interactive tooltips that highlight and explain each new feature, making it easier for them to understand and use the updates.
- Content for end users such as AI videos
- Tracking of users of Google Pixels and Meta
- Android/IOS App for displaying content like festival details and concerts.

## 1.4 Related Work

### Marketing Applied in Art Sector

This category is focused on marketing and art museums because the Culturatela.com solution is well applied in the Italian art ecosystem. One of the main objectives of the software is to deliver the right path to analyze and valorize the community and the visitors interested in the cultural field.

The "advanced museum services" research project seeks to create a prototype web-based application tailored to enhance the management and communication within cultural heritage institutions while providing cataloging and web publishing tools specifically geared towards museums, with a strong emphasis on user interaction. The primary objective is to develop a product capable of anticipating the future evolution of museum services, recognizing the influence of technological advancements on users' behaviors and expectations. Developed collaboratively by the Department of Electrics and Electronics Engineering at the University of

Cagliari and SoSeBi Srl, the project receives financial support from the Autonomous Region of Sardinia through European local development funds [4]. The project faces challenges such as needing ongoing maintenance by skilled personnel and the risk of becoming obsolete due to new technologies. A potential solution is to make the application more user-friendly, including a separate dashboard for museum organizers to manage and update posts. By anticipating future needs and collaborating with experts, the project aims to stay relevant and sustainable, enhancing the experience for both visitors and museum staff. Another study explores the digital transformation of cultural institutions and the online art market, examining the rapid digitization of museums and galleries, the surge in online art sales, and the impact of digital technologies on exhibitions, educational programs, and digital art creation [5]. Assessing the impact of digital transformation on the art market and buyer behavior can be complex due to the rapidly evolving nature of online platforms. The proposed solution could be to make museums easier to manage and more engaging for visitors by creating specialized web tools for better organization and online sharing. The collaborative effort between academia and industry ensures a robust and forward-thinking approach. A research paper completed as part of the MA in Art Law and Arts Management at the International Hellenic University examines the digital strategies of contemporary museums and their impact on audience development. Focusing on the Rijksmuseum in Amsterdam, the study provides a detailed analysis of digital methods used to engage audiences [6]. Digital technologies have transformed museums, enhancing visitor experiences and accessibility. Key practices include virtual reality tours, mixed reality exhibitions, social media engagement, digital collections, and augmented reality. These innovations enable museums to connect with global audiences, preserve exhibits, and offer interactive experiences. Examples include the British Museum's VR tours and the Cleveland Museum of Art's mixed reality exhibits. [7] Despite these benefits, challenges persist, such as the digital divide, copyright issues, financial constraints, and impacts on staff. The Archaeological Museum of Dion's traditional approach, as explained by its retired director, contrasts with the digital success of other Greek museums like the Acropolis Museum. Balancing digital and traditional methods can help museums maximize benefits while addressing these challenges. A study on gamified mobile applications, such as "MuseumStars," explores how these tools can foster engagement with museum content. Using a mixed-methods approach grounded in the uses and gratifications theory, the study analyzes user motivations and the impact of such applications on supplementing or substituting physical museum visits [8]. A mobile app developed for cultural tourism in India aims to digitize the travel industry by providing easy access to information on cultural events and festivals, allowing travelers to book tickets conveniently [9]. These digital tools face challenges in scalability, applicability to diverse museums, and sustaining long-term engagement. Implementing and maintaining them can

be resource-intensive. Offers insights into the digitization of cultural institutions and online art sales, highlighting the potential of digital technologies to enrich exhibition experiences and educational programs.

### **Improvement of Well Known Systems**

This category is focused on the improvement of an already well-known system. Culturatela.com, through a series of investigations that are constantly shared with the customer, analyses and develops strategies in order to provide better and more tailor-made solutions that aim to improve the operations and organization of the cultural facility.

A study on the use of QR Code technology in museum collection management highlights the benefits of digital databases for easy tracking and updating of collection information. QR Codes also improve visitor engagement by providing instant access to detailed object information [10]. An Android mobile application that addresses the inefficiencies of manual ticket verification by allowing online ticket purchases and using QR codes and fingerprint scans for user verification [11]. Potential challenges include technical issues with the app or QR code scanning, which could disrupt the ticketing process. The requirement for reliable Internet access and smartphone availability may exclude some users. Privacy concerns regarding the storage of personal data and fingerprints could also affect user adoption and trust in the system. Implementing QR Code technology in museum collection management enhances efficiency by providing a digital database for easy tracking and updating of collection information. It improves visitor engagement by allowing instant access to detailed object information via smartphones. QR Codes also support compliance with museum ethics and legal frameworks by providing transparent documentation of collection history. I used a QR code for ticket validation and payment for e-ticketing. The implementation of IoT-aware architectures in museums aims to enrich user experiences by utilizing recent advancements in ICT. An example is the architecture proposed for the MUST museum in Lecce, Italy, which includes indoor localization, wearable devices, and mobile applications to provide personalized content and facilitate multimedia sharing. [12] The adoption of RFID technology-based ticketing and visitor management systems in Chinese museums aims to improve efficiency and capture visitor behavior data for better decision-making. These systems support the implementation of targeted cultural value-added services. [13] Challenges for these technologies include technical complexity in development and integration, privacy and security concerns with cloud data, high initial and maintenance costs, potential user resistance to new technologies, and the need for ongoing support and updates to ensure system reliability. Specifically, RFID systems must address initial investment costs, privacy concerns

related to data collection, technological issues affecting system reliability, and potential visitor resistance to new technology. Addressing these challenges is crucial for the successful implementation and acceptance of RFID systems in museums. The proposed IoT-aware architecture aims to enhance the museum experience by providing personalized and interactive content through wearable and mobile devices. It improves accessibility with BLE technology and facilitates multimedia content sharing in the cloud, potentially increasing visitor engagement and satisfaction. Meanwhile, RFID technology-based ticketing and visitor management systems improve operational efficiency by automating processes and enhancing the visitor experience. These systems capture detailed visitor data, enabling museums to analyze trends and preferences for better decision-making and facilitate customized cultural services, thereby enhancing visitor engagement.

In this paper, the inefficient and error-prone manual registration for The Millennials' event via WhatsApp is improved with a web-based ticket booking system. This system aims to streamline ticket transactions, monitor registrations, enhance organizer responsiveness, and simplify the booking process. Developed using the waterfall model, the system employs PHP, Bootstrap, and MySQL. Evaluation shows the system significantly simplifies and improves the efficiency of the registration process. [14] In the recent past, visiting a museum's exhibits required a physical trip, but with the advent of the World Wide Web, affordable software, and easy access to technology, museums have started showcasing their collections online. This allows them to offer educational resources to those unable to visit in person. Advances in technology and social media now enable museums to reach a wider audience instantly, often through recommendations and direct links from friends or mentors. This expanded reach helps museums market to more art enthusiasts and solicit donations while displaying works in ways that best support their mission. However, this opportunity also brings competition from web-based publishers like Google, which offers virtual museum tours. By researching and interviewing various institutions, this study develops metrics to evaluate museums' web presence, considering page views, budgetary factors, and staff efforts. Despite varied results, the study finds a significant increase in website traffic and engagement as museums increasingly market through social media platforms like Facebook and Twitter. [15] There is increased competition from other web publishers like Google offering virtual tours. Developing and maintaining an online presence can be costly. Results vary between institutions, with some seeing more success than others. Museums must balance resources between physical and online spaces, and virtual experiences may lack the authenticity of in-person visits. Challenges include potential technical issues during the implementation and maintenance phases. Users may experience a learning curve, and ensuring data security is crucial for user trust and privacy protection. Because of web-based ticket booking system for The Millennials' event automates the registration process, reducing errors and inefficiencies associated

with manual methods like WhatsApp. It saves time, enhances data management, and improves organizer responsiveness. The use of PHP, Bootstrap, and MySQL ensures a robust and user-friendly system that simplifies ticket transactions and participant registration. Museums can now reach a global audience online, making art accessible to those who can not visit in person, like teachers and students. Social media helps attract a larger audience quickly and effectively. Online displays create new fundraising opportunities and allow museums to showcase collections in flexible formats. There is another "E-Ticketing system for Indian museums and heritage sites" that focuses on the adoption of smart ticketing systems within a vast network, covering over 1,015,000 km and serving over 22 million visitors daily. Leveraging digitization and smartphone technology, the system aims to streamline ticketing processes, enhance authentication, and improve efficiency, reducing paper usage and contributing to sustainability [16]. Research on e-ticketing trends in urban communities, particularly in Kuala Lumpur, reveals significant adoption among young, educated, and higher-income individuals. Convenience and ease of use are key motivating factors for online ticket purchases [17]. However, e-ticketing poses security risks and privacy concerns with online transactions, and some users, particularly older adults, may struggle with the technology. To address these issues, I implemented the NEXI channel for secure payments and provided payment via QR codes, offering a secure and easy way to purchase e-tickets. This system enhances speed and security, minimizes paper use, reduces counterfeit tickets, and offers convenience to visitors.

## **Society Requirements**

Museums play a fundamental role in society. To maintain their relevance and effectiveness, they must adapt to technological advancements and the evolving needs of their visitors. As bastions of culture and education, museums should equip visitors of all ages with tools that align with contemporary learning methods. This approach ensures that museums remain dynamic spaces for cultural enrichment and personal growth.

This study investigates how Italian museums are adapting to modern technological advancements, with a focus on digital tools and the impact of the COVID-19 pandemic on museum operations. The research highlights the progress made by museums in adopting interactive experiences and the ongoing challenges they face [18]. A study comparing the digital services used by museums and art galleries in Krakow before and during the COVID-19 pandemic found that digital tools became more accessible during the pandemic, although the number of museums offering mobile applications did not change [19]. An integrated mobile application model



proposed for cultural tourism in Indonesia combines online ticketing and funding platforms to address financial constraints and organizational challenges during the COVID-19 pandemic [20]. Challenges include the cost of implementing digital innovations, limited access to technology, and difficulties maintaining authenticity. Smaller museums struggle with these issues, and there's a risk of widening socioeconomic disparities. Not all museums offer online visit booking or have updated their digital services, limiting accessibility. The proposed solution is for museums to leverage innovative technologies to enhance interactive experiences, making visits more enjoyable and memorable. Digital tools can widen access to collections and streamline operations, keeping museums relevant. During the pandemic, digital tech allowed museums to share culture online, providing learning resources and online ticketing when physical visits were impossible.

A study utilizing a system dynamics approach investigates the challenges faced by the National Museum of Natural Science (NMNS) in Taiwan, particularly in attracting and retaining visitors. The research explores the interplay between service quality, facility availability, and visitor satisfaction, simulating the effects of various policy implementations to enhance long-term satisfaction [21]. Limited generalizability of findings beyond the specific case study museum. May lack comprehensive coverage of diverse audience segments and their digital engagement preferences. Offers a detailed analysis of a digitally advanced cultural institution's approach to audience engagement through innovative digital methods. Provides practical insights into implementing digital strategies for audience development.

## 1.5 Methodology

The software development life cycle (SDLC) describes the necessary steps for creating any software or application program. Basically, it is a methodical process that leads the software development journey from conception to deployment. This process usually includes various stages, each with its own importance. Every step plays a significant role in achieving the overall goal of producing a high-quality product, and it ensures collaboration, effectiveness, and efficiency among the team members.

The main goal of the SDLC is to provide a structured way to develop software or applications. It is all about making sure that the final product not only meets the client's needs and quality standards but also gets delivered on time and within budget. This process helps keep everything organized and ensures that the end result satisfies everyone involved.

To keep information and computer systems safe, it is important to think about security from the very beginning of a project until the end. This means considering security risks when planning a project and making sure security measures are in place as the project is developed. Throughout the project's life, security should be regularly checked and updated to keep it strong. Even when the project is finished and no longer needed, steps should be taken to safely shut it down and protect any remaining data. This way, the system stays safe from potential threats at all times. [22]

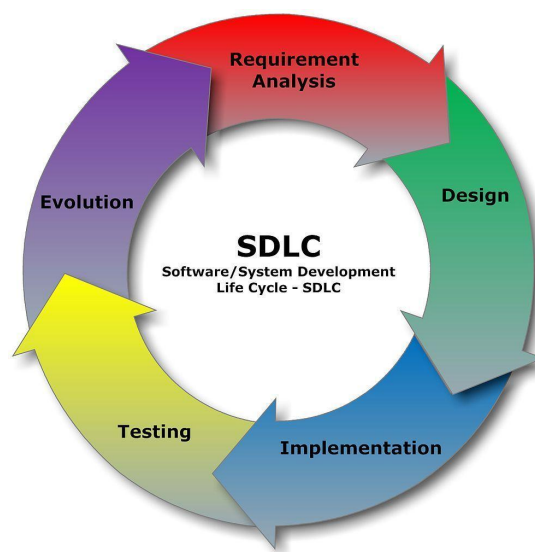


Figure 1.1: SDLC

# Chapter 2

## Planning

### 2.1 Goals

Planning is the first phase of SDLC. As I figure out what my project is about in this phase, Culturatela.com initiates the project and provides the necessary information to guide the planning phase like which modules I have to create and which I have to update. Basically, they define the project objectives, set deadlines, and allocate resources based on their requirements and expectations. Essentially, the planning phase is like creating a roadmap that guides me through the project journey, ensuring that everything runs smoothly and according to plan. It sets the foundation for the subsequent phases of the SDLC and is critical for the overall success of the project.

In the end, Culturatela.com wants to deliver a fully responsive booking system that operates seamlessly 24/7, providing uninterrupted service to customers while meeting and exceeding their expectations. Customer satisfaction is Culturatela.com's main priority, and it is committed to ensuring that every interaction with the system is smooth, efficient, and satisfactory.

End-user security is also an important priority for Culturatela.com. To address this concern, the booking system will incorporate a secure payment method using QR codes, especially in public places displayed on large screens. This approach provides an added layer of security by allowing users to make payments without displaying sensitive information such as bank details. By scanning the QR code displayed in public spaces, users can complete transactions securely and conveniently, safeguarding their personal and financial data. This initiative not only enhances user confidence but also reinforces the commitment to prioritize security and privacy throughout the booking system's operations.

## 2.2 Challenges

The project completion deadline is set for four to five months from the starting date, which is December 2023. This timeline aligns with the launch schedule for our new product line, Culturatela.com. To ensure timely delivery, I have to track progress effectively and address any potential delays proactively. This structured approach will enable me to stay on track and meet the established deadline while maintaining high standards of quality and functionality.

To successfully complete the project, I could rely on a diverse range of resources. Firstly, a mentor was assigned to me who guided me to bring the responsive booking system to life, ensuring it meets both functional and aesthetic requirements. Additionally, access to essential technological resources such as development tools, design software, hosting services, and testing environments is crucial for seamless execution. Financial resources play a vital role, including software licenses, hosting fees, and operational expenses. By securing these resources, which are provided by Culturatela.com, I can effectively navigate the development process and develop a high-quality booking system that meets the needs of both end-users and administrators.

Several potential challenges may arise during the project, I faced compatibility issues and design issues and a lack of key people involvement in the designing phase which I explained in the next paragraph, and it is essential to have plans in advance to address them effectively. One common issue could be unexpected delays due to technical difficulties or unforeseen complexities in the development process. To mitigate this, I will implement regular progress assessments and maintain open communication channels with the project manager to identify any roadblocks early on. Additionally, resource constraints or budget limitations may pose challenges to project execution. To overcome this, I will prioritize tasks based on criticality and explore cost-effective solutions without compromising quality. Furthermore, changes in project scope or requirements could lead to scope creep and project drift. To manage this, I will adhere strictly to the project scope and establish a change control process to evaluate and approve any modifications. Regular meetings with the project manager and status updates will foster a collaborative environment and ensure everyone remains on the same page throughout the project lifecycle. By proactively addressing these potential challenges, I can minimize disruptions and keep the project on track toward successful completion.

If I do not have all the things I need to complete a project, it can cause problems like delays or leaving parts of the project undone. So, it is important to make sure I have everything necessary before starting a project to avoid these issues. I faced this problem in app development because I did not have a clear design mockup for the app and it took more time to complete it. When the goals of a project are not clear, it can create confusion and misunderstandings. This happens because everyone may have different ideas about what they are supposed to be working towards. For example, some might think the goal is to finish quickly, while others might prioritize making the product perfect. Without clear goals, people may end up working on different things or not knowing what to focus on, which can slow down progress and cause frustration. So, it is really important to make sure everyone understands what the project is aiming to achieve right from the start. When key people who are important to a project do not join in the planning stage, it can cause problems. This is because they might not fully understand what is supposed to happen or what is needed. It is important to involve all the right people from the start to make sure everyone is on the same page and the project runs smoothly. I faced some problems in the designing phase because there were no clear design instructions from them like font or color scheme etc. Unexpected technical issues, such as software bugs or compatibility issues, could disrupt the development process and delay project timelines. To mitigate this risk, I will conduct thorough testing at each stage of development to identify and address any technical issues early on. Additionally, I maintain open communication with the development manager and leverage their expertise to troubleshoot and resolve any technical challenges promptly. I faced compatibility issues when using CSS classes to ensure responsiveness across different devices and browsers such as Chrome, Safari, and Firefox. Unfortunately, I had not anticipated this risk. Nevertheless, I managed to resolve the issue by adopting the Bootstrap framework. Bootstrap's robust compatibility ensures consistent display across various browsers, mitigating such concerns effectively.

Addressing problems at the start of the planning phase is crucial for ensuring the project's success. To avoid encountering such issues, I link my project to Git and do my coding after. Git allows me to create personal branches and feature branches, which means I can make changes without affecting the entire project. If something needs fixing, I can work on that specific branch without disrupting the rest of the project, ensuring smoother development and better project management. By identifying potential challenges early on, I can develop effective strategies to handle them before they escalate. This proactive approach involves carefully analyzing possible risks and issues that could arise during the project and creating plans to mitigate them. In this way, I am better equipped to handle any difficulties that may arise later in the project. This not only minimizes disruptions but also increases the project's chances of achieving its goals within the set timeline and budget. Overall, taking proactive steps during the planning phase sets a solid foundation for a smoother project journey and greater success in the end.

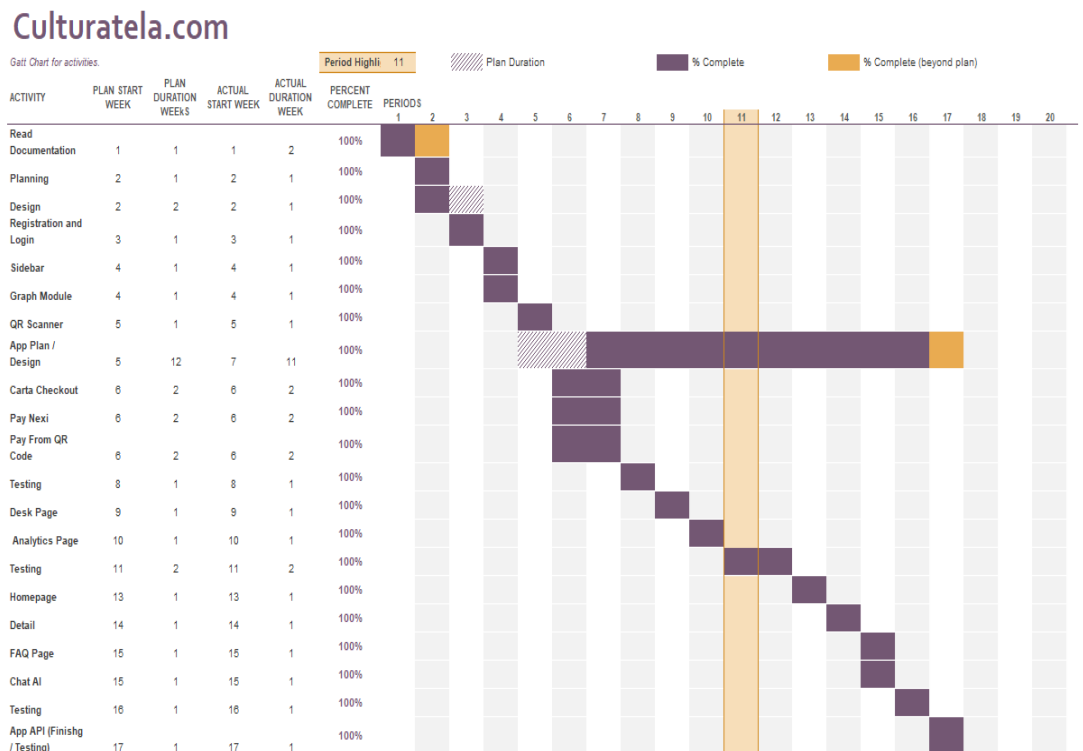


Figure 2.1: Gantt Chart

In Figure 2.1, I have carefully outlined the project's activities, detailing their start and end weeks. While most tasks were completed according to schedule, there were slight deviations for two specific activities. Reading the documentation required an additional week, and the app planning and design phase was extended by one week. This delay was due to my delayed start on this particular task. Notably, by week 11, marked by the yellow highlight, half of the project was successfully completed. In the end, even with these small delays, I finished everything I needed to do for the project in the 18 weeks I planned.

## Chapter 3

# Analysis

When creating software, it is important to know exactly what the people who will use it need and want. To do this, I talk to different people involved. I ask them lots of questions to understand how they want the software to work, what steps it should take, and how I can make sure it is safe to use. Once I have all this information, I carefully look through it to figure out exactly what the software needs to be able to do. This helps ensure that the software I create will meet the needs of the people who will use it.



Questions	Answers		Answers	
	Lucca	Sondrio	Catania	Velletri
What facility do you manage?	3 M, 1L	3 M, 1T	4 M	1 M
Which ticket system do you use?	Excel	VivaTicket	other	Excel
Did you sell tickets onsite or online?	Online	Online	Online	Onsite
Do you usually have access to your visitor's data?	Yes	No	No	Yes
Did you use the kiosk(dynamic QR link for payment) machine for the automatic selling?	No	No	No	No
Do you need a QR scan for validation at the entrance?	Yes	Yes	Yes	Yes
Do you need a reporting system?	Yes	Yes	Yes	Yes
Do you want to provide AI content?	Yes	Yes	Yes	Yes

**Table 3.1:** Question Analysis (M\* Museums L\* Library T\* Theater)

The analysis provided in Table 3.1 highlights the plans of several municipalities to implement online booking ticketing systems. In Lucca, for instance, where three museums and one library are managed, an Excel sheet serves as the database for the ticketing system. However, this manual process requires staff to input visitor data manually, which includes timestamps. Meanwhile, in Sondrio, which oversees three museums and one theater, Vivaticket is utilized as the booking system. Despite this, access to visitor data is lacking, and there is no kiosk mode for automatic ticket selling. Instead, QR code validation and a reporting system are employed. Similarly, Catania, with four museums under its management, sells tickets both online and onsite but faces challenges in accessing visitor data and implementing a kiosk mode. Meanwhile, Velletri manages one museum and relies exclusively on onsite ticket sales. Although they use an Excel sheet for booking data storage, they do have access to visitor data and a reporting system. Notably, none of the municipalities have integrated an AI chat system or AI content for museums and

theaters, nor do they utilize kiosk mode for ticket selling.

### 3.1 Direct Communication

Direct communication with stakeholders is a crucial step in understanding their needs and expectations for the software project. By engaging directly with customers which is representative of Comune di Lucca, Sordrio, Catania, and Velletri, I better understand the information about their requirements and preferences. They have expressed a need for software that streamlines ticket organization and sales. Their priority is to find a solution that offers the safest and most user-friendly experience for ticket validation and purchasing. This direct feedback provides valuable insights into the specific functionalities and features that customers prioritize, allowing me to tailor the software development process to meet their needs effectively. By actively listening to customers and incorporating their input into the project plan, I can ensure that the resulting software aligns closely with their expectations and delivers maximum value.

### 3.2 Information Gathering

During these conversations, I understand detailed information about the functionality of the software. This includes understanding the specific tasks the software needs to perform, any features it should have, and any constraints or limitations it may need to address. Such as, administrators will have the capability to edit content, manage ticket slots, and adjust the number of available tickets. Additionally, they will be able to search for and validate tickets, as well as access an overview of data and export it in CSV format. Operators, on the other hand, will require a complete flow for automatic selling, especially through kiosks(Pay through QR). Finally, for customers, the software should provide AI-generated information during their visit, enhancing their experience and engagement. These features aim to meet the specific needs of each user group while ensuring the smooth operation and effectiveness of the ticketing system.

Moreover, quick responses to user questions are essential. When users have questions, an AI chatbot will be there to help them out. This chat support will be available all day and night, so users can get assistance whenever they need it. Also, it is important to make sure that paying for tickets is safe and secure. This way, users can feel confident making payments from anywhere without worrying about their information getting into the wrong hands.

### 3.3 Needs and Goals Analysis

After gathering all the information, I conduct thorough research into the needs and objectives of the end users. Like to provide portability and enable ticket sales from anywhere, the ticketing system was designed to be accessible from various hardware devices, including smartphones, tablets, and computers. This multi-device compatibility ensures that users can conveniently purchase tickets using their preferred device, whether they are at home, on the move, or at the venue. By leveraging the versatility of mobile devices, such as smartphones, the ticketing system offers greater flexibility and convenience to users, allowing them to access and purchase tickets anytime, anywhere, enhancing the overall user experience and providing more portability. Alongside this, I also consider the primary objectives outlined by the stakeholders, such as satisfying existing customers and increasing engagement through online ticketing tools, as well as attracting new customers to visit their museums. By incorporating these additional goals into my analysis, I aim to ensure a comprehensive understanding of what the software needs to achieve for the users and the organization as a whole.

Before starting to work on the software development process, it is essential to establish a thorough understanding of all requirements. This involves clarifying any uncertainties, reconciling conflicting needs, and prioritizing features based on their significance to the end user. This preparatory phase is similar to creating a blueprint or detailed plan before starting construction on a building, ensuring that everything is carefully organized before actual work starts. By carefully defining requirements in advance, for example, analytics of all ticket sales, tracking the number of tickets remaining, and graphical representations enable more accurate management, and portability from any device, kiosk mode for ticket selling and ticket validation is the main functionality. It is a groundwork for successful software development, minimizing the risk of misunderstandings and ensuring alignment with user expectations.

### 3.4 Reporting and Documentation

Throughout this phase, I carefully analyzed all the information I had. This documentation serves as a central reference point for the entire development process, ensuring that I am aligned with and aware of the project objectives and requirements. By maintaining detailed documentation, I facilitate clear communication and collaboration with the project manager, enabling him to stay on the same page regarding what needs to be achieved. Additionally, this documentation provides a foundation for making informed decisions and adjustments as the project evolves.

It serves as a valuable resource for guiding the development process, enabling agile responses to changing circumstances, and ensuring that the project stays on track toward successful completion.

With the assistance of this dedicated platform, stakeholders will be empowered to manage events at cultural venues and libraries independently and efficiently. They can utilize specialized registration and purchase formats tailored to different event types, such as free or paid, and for various demographics like minors, families, and adults. Moreover, stakeholders benefit from the ability to track and analyze visitor data, enabling them to enhance the visitor experience and promote events and venues effectively.

Additionally, the platform will provide stakeholders with a comprehensive database of tourists, categorized by interests and demographics. It offers various financial, quantitative, and qualitative reports to support accounting and administration tasks. Furthermore, the platform facilitates collaboration with other entities in the area by allowing museums, events, and private exhibitions to be hosted on the platform. This enables the creation of "combined tickets" for multiple initiatives, fostering cross-marketing activities and simplifying access to city services for tourists.

Stakeholders want one notable collaboration facilitated by the platform: the "combined ticket" initiative with the administration of Lucca, including the Puccini Museum of Lucca. This partnership will highlight the platform's capability to foster collaboration between cultural institutions and local authorities, ultimately enhancing the cultural offerings and tourism experience within the city.

To further enrich the platform's offerings, consider integrating personalized event recommendations based on visitor preferences and past attendance. Additionally, develop an interactive event calendar for seamless event browsing and ticket purchasing. Enable social media integration to facilitate event sharing and increase visibility. Implement a feedback system for visitors to provide reviews and ratings, enhancing stakeholder insights into visitor satisfaction. Develop a mobile app companion for convenient access to event information on-the-go. Ensure accessibility features for an inclusive user experience. Finally, implement a data analytics dashboard for stakeholders to track event performance and audience engagement. These enhancements will elevate the platform's capabilities, fostering increased visitor engagement and overall success.

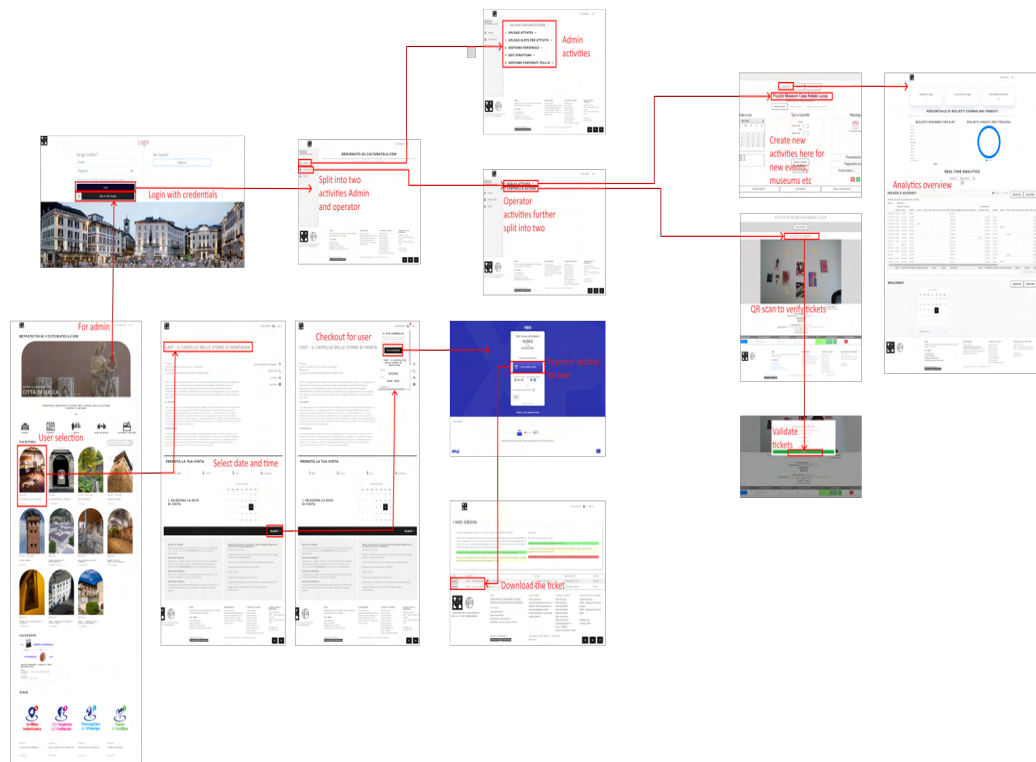
# Chapter 4

## Design

In previous phases, I gathered information and analyzed it. In this phase, I designed the whole structure of the project, how different components will interact and work together, and anticipated the impressions that end-users will have when they interact with it. It is a very critical step because what I design in this phase I have to work on later in the next phases. I have to make sure the design I will make works well on all kinds of devices, such as computers and mobile phones. In this way, people get an excellent experience no matter which kind of device they are using.

### 4.1 Design Phase Purpose

The design phase acts as a crucial bridge between conceptualization and implementation. Here, I transform the gathered information and insights into a detailed blueprint design for the project. This encompasses more than just visual design elements; it also includes considerations for functionality and user experience. By meticulously planning the project's design at this stage, I laid the foundation for a seamless and user-friendly final product. This involves defining how different components will interact, structuring navigation pathways, and ensuring that the user interface is intuitive and visually appealing. Decisions made here heavily influence subsequent development steps. By paying close attention to both form and function during the design phase, I set the stage for successful implementation and a positive user experience.



**Figure 4.1:** Work-Flow

In Figure 4.1, the workflow of the design is shown to align with user preferences, featuring a flexible search feature that enables users to explore options based on destination, events, or all available choices. The dynamic nature of the content ensures that every piece of information on the home page is retrieved through a server call. This approach makes it highly adaptable for Culturatela.com, allowing effortless updates such as adding or removing events and cities or modifying content by simply adjusting or modifying the JSON data. This flexibility streamlines the process of managing and updating information, providing an agile and user-centric experience.

Detailed process for managing and purchasing tickets for museums or events through distinct admin and user interfaces. It begins with admin logging in using their credentials, and providing access to different functionalities based on their roles. The interface then splits into two main activities: admin activities and operator activities. Admins have access to features for creating and managing activities, including adding new events or museums and inputting necessary details. They can also view an analytics overview to monitor performance metrics such as ticket sales. Operator activities are further divided into managing current activities and verifying tickets.

Operators use QR scanning to verify the tickets of users as they enter the event or museum, and the system then validates these tickets to ensure they are legitimate.

On the user side, after logging in, users can select an event or museum from the available options. They proceed by selecting a date and time for their visit, followed by the checkout process where they review their selections. The payment section allows users to enter their payment details to complete the purchase. Once the transaction is successful, users can download their tickets. This comprehensive flow ensures that all aspects of ticket management, from creation to validation, are efficiently handled, providing a seamless experience for both administrators and users.

## 4.2 Cross-Device Optimization

This may involve adapting navigation menus, adjusting touch targets, and optimizing performance to ensure a seamless experience across different platforms and devices. I designed the interfaces that are intuitive, efficient, and enjoyable to use. It involves conducting usability testing, creating user personas, and iterating on design prototypes to refine and enhance the overall user experience.

Here are the design principles followed to create the mentioned web pages.

- **Responsive Design:** The main principle behind all the designs is responsiveness. Each page is designed to adapt seamlessly to various screen sizes, ensuring an optimal viewing experience across devices, including phones, tablets, and computers, it emphasizes the action of designing each page to adapt well to different screen sizes, with a particular emphasis on providing an optimal viewing experience across various devices. This is achieved by utilizing techniques like fluid layouts and media queries.
- **User-Centric Navigation:** The main page design prioritizes user-friendly navigation by offering clear and intuitive options for exploring different categories such as museums, events, and concerts. This allows users to easily find what they are looking for and proceed with ticket purchases effortlessly.
- **Clear Call-to-Action (CTA):** Each page includes clear and prominent call-to-action elements to guide users toward their desired actions. For example, on the main page, the CTAs prompt users to explore categories and purchase tickets, while on the AI chatbot page, the CTA invites users to interact with the chatbot by typing their queries.

- **Consistent Branding and Visual Identity:** Consistency in branding elements such as colors, typography, and imagery is maintained across all pages to reinforce the brand identity and provide a cohesive user experience. This helps users easily recognize and associate the web pages with the brand.
- **Accessibility:** Accessibility considerations are integrated into the design to ensure that all users, including those with disabilities, can access and interact with the web pages effectively. This includes features such as alternative text for images, keyboard navigation support, and color contrast for readability.
- **Feedback and Interaction:** The AI chatbot page is designed to encourage user engagement through interactive features such as a text input box for typing queries and options for receiving responses in both text and voice formats. This promotes user interaction and enhances the overall user experience.
- **The functionality of the QR page** revolves around facilitating secure and efficient transactions through the use of dynamic QR codes. Users visiting the QR page are presented with a unique QR code generated for each transaction. By scanning this QR code with their smartphones, users can initiate payments swiftly without the need to manually input sensitive bank details. This process not only enhances security by reducing the risk of unauthorized access to financial information but also ensures privacy protection by preventing the exposure of bank details on public displays. Additionally, the QR page may include features such as real-time QR code generation, integration with payment gateways for seamless transactions, and user-friendly instructions for scanning the QR code. Overall, the functionality of the QR page streamlines the payment process, enhances security, and offers users a convenient and protected means of completing transactions.
- **The functionality principle underlying the collection and access page** revolves around providing users with comprehensive and efficient access to ticket sales data. Through a table-format report, tickets are categorized by type, enabling users to easily track and analyze sales performance. Key metrics, such as total revenue from online sales and box office transactions, are prominently displayed for quick reference. Additionally, users can select specific time slots from the calendar interface to access detailed lists of participants who made online purchases. To streamline the user experience, dates without online sales are initially deactivated, with a dynamic function activated to display only slots with online transactions. This functionality principle ensures that users can efficiently retrieve specific and relevant information, facilitating thorough reporting and analysis for informed decision-making.



Here are some designs with different functionalities.



**Figure 4.2:** Design Compatibility

Figure 4.2 presents the design for the main page. I am going to use a Totum which is the big screen in museums and public places to buy tickets, to make sure it looks good on different screen sizes. I will test my design on small screens, like phones, medium-sized screens, like tablets, and large screens, like computers, as well.

If people can easily navigate and use the main page no matter what device they are using, this way I can make sure everyone has a good experience when they visit our website, regardless of the device they are using.

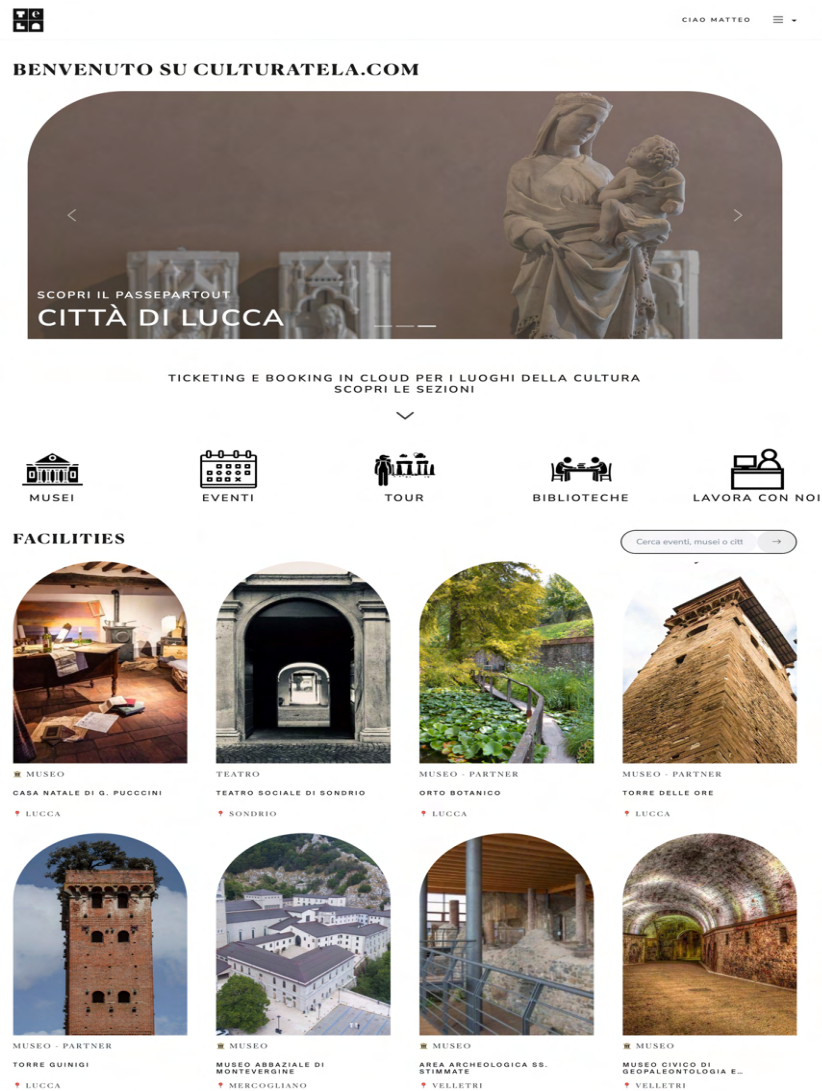
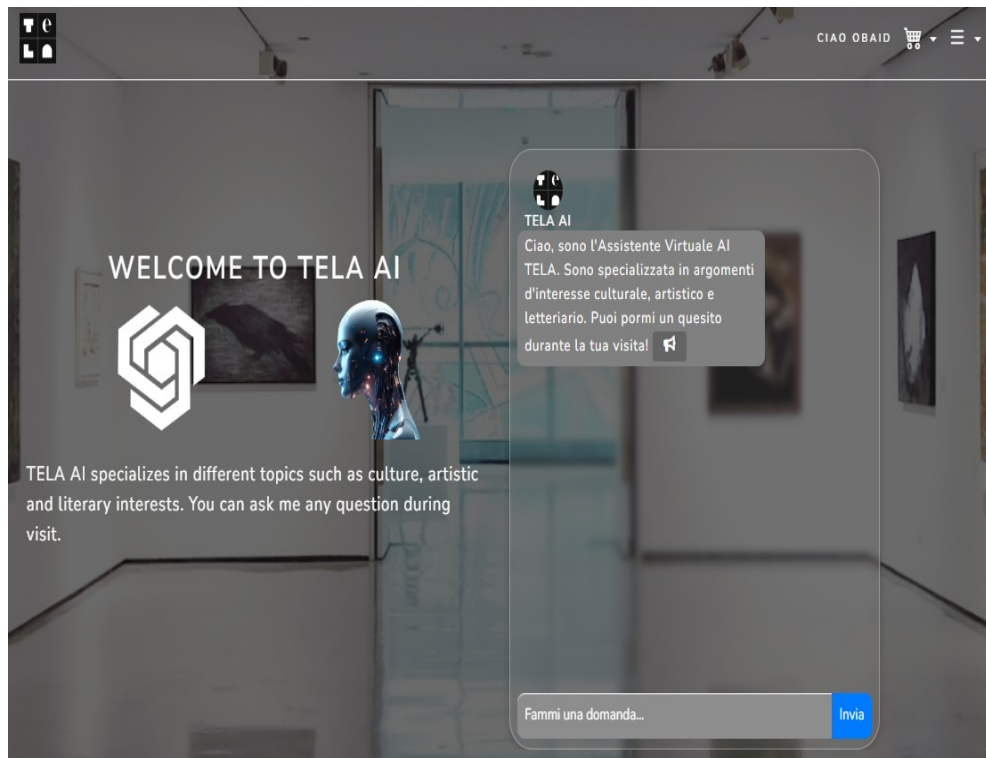


Figure 4.3: Main Page

Figure 4.3 presents the design for the main page. Customers will find options to explore different categories, such as museums, events, concerts, and more. They can click on the category they are interested in to buy tickets or learn more about what is available.

For example, if someone wants to visit a museum, they can click on the "Museums" category. This will take them to a page where they can see all the museums we offer tickets for. They can browse through the options, read descriptions, and select the museum they want to visit. Then, they can proceed to purchase tickets directly from here.

Similarly, customers interested in events or concerts can choose those categories from the main page and explore the available options. This makes it easy for customers to find and purchase tickets for the types of experiences they are interested in, all from one convenient location.



**Figure 4.4:** AI Chat Design

Figure 4.4 presents the design for the AI chatbot page. On this page, users can type their questions or queries into a text box. They can ask anything they want, and our AI bot will provide responses in both text and voice formats.

This setup allows users to interact with our AI chatbot in a way that is most convenient for them. Whether they prefer reading responses or listening to them, our chatbot provides flexibility to accommodate different user preferences.

### 4.2.1 Compatibility Considerations

During this phase, I prioritize the creation of responsive and adaptive designs that seamlessly adapt to various screen sizes, resolutions, and input methods. This guarantees a consistent and optimized user experience across desktops, laptops, tablets, and smartphones. Furthermore, cross-device optimization is a key focus area during this phase. Leveraging tools like Totum (big screens to buy tickets in museums), I verify the design's compatibility and functionality across various screen sizes, ensuring a seamless experience on phones, tablets, and computers. Through usability testing, user creation, and iterative design prototyping, I refine and enhance the overall user experience, aiming to provide interfaces that are intuitive, efficient, and enjoyable to use across different devices.

When considering compatibility, I encountered several issues and implemented various solutions to address them. Firstly, I encountered challenges with SSL (Secure Sockets Layer) compatibility, particularly with older browsers and devices. To mitigate this, I ensured that the SSL configuration was up-to-date and compatible with a wide range of browsers and devices.

Additionally, I utilized browser APIs to test compatibility across different browsers such as Firefox, Chrome, Safari, and Edge. Furthermore, I incorporated the Bootstrap framework into the project. Bootstrap is known for its cross-browser compatibility and responsive design features, making it an ideal solution to ensure consistent rendering across various devices and browsers. By utilizing Bootstrap, I could streamline the development process while ensuring broad compatibility with different devices and browsers.

## Chapter 5

# Enabling Tools/Technologies

I developed an online booking system with the primary goal of creating an interactive and secure platform for purchasing tickets to museum events. The project involves designing an engaging front-end interface to attract users, featuring intuitive navigation and a seamless ticket selection process. This includes implementing search and filter functionalities, as well as interactive calendars, to enhance user experience. Additionally, a robust backend system is crucial to ensure the security of user data and financial transactions. Tools and technologies such as user authentication, encryption of sensitive information, and adherence to industry standards for data protection are integral to safeguarding user privacy. Moreover, scalability and performance optimization are key considerations, with the system designed to handle high volumes of traffic while maintaining responsiveness across various devices and platforms. By carefully addressing these aspects, the online booking system aims to provide a convenient and reliable solution for both museum visitors and administrators alike.

The main features are as follows.

### 5.1 Front-End

To create a good and user-friendly front end for the online booking system, I plan to utilize a combination of tools and technologies. These technologies offer component-based architecture and efficient state management, facilitating the creation of dynamic and responsive user interfaces. By using these tools and technologies, I aim to create a front end that not only attracts users but also provides a seamless and enjoyable booking experience for museum visitors.

Here are the tools and technologies that are used for the frontend.

### 5.1.1 JavaScript

I used JavaScript as a programming language in my project. At first, JavaScript was created for dynamic web pages. I can add these kinds of scripts directly to the HTML of a web page, and it runs automatically whenever the page is loaded without needing any kind of special operation.

JavaScript is unique because it does not require any kind of specific setup or plan, unlike other programming languages. It does not require a special environment, just a web browser. I can run JavaScript on the server side, not just the browser, using a JavaScript engine I can run on any computer. Basically, it lacks access to lower-level memory or CPU functionalities, as its original design was intended for use in web browsers, which do not necessitate such access.

For instance, when I am employed in a web browser, it takes charge of the user interaction and facilitates communication with the web server. Here is a detailed breakdown of the various tasks that JavaScript can execute.

JavaScript can dynamically add new content to a webpage, modifying existing styles to enhance the user interface. It responds to various user actions, including pointers, mouse clicks, and keyboard inputs. JavaScript can make various requests to remote servers over the Internet using AJAX (Asynchronous JavaScript and XML) and COMET, which can enable the upload and download of files. It has the ability to access and manipulate the cookies, which allows it to store and retrieve small pieces of information on the user's device. JavaScript increases user engagement by asking questions and providing answers to users, contributing to an interactive and engaging user experience. Utilizing the local storage JavaScript can store the data on the client side, and it allows the continuity of information between user sessions.

JavaScript capabilities in web browsers are intentionally restricted to protect user privacy and protect against potential security threats. The main goal is to prevent unauthorized access to critical or sensitive information and to ensure the overall security of data. [23]

- **File Operations Limitations** JavaScript on a web page is deliberately barred from executing actions like reading, writing, and copying or manipulating files. This can prevent security breaches and unauthorized access. But some modern browsers do allow some file interactions, access is restricted and contingent on specific user-initiated actions, such as dragging and dropping files into the browser or selecting files via the input tag.
- **Isolation of Tabs/Windows** Tabs or windows in browsers are designed to

operate independently with limited communication between them. Even when JavaScript in one window tries to open another, there is isolation between them. This isolation extends to different pages originating from distinct regions, protocols, or ports, and it prevents JavaScript on one page from accessing or manipulating content on another page. This measure enhances overall security by minimizing potential cross-site scripting vulnerabilities.

- **Cross-Origin Data Collection Restrictions** JavaScript can communicate with the server from which the current page originated through the network. There are stringent limitations on collecting data from servers with different origins. Explicit consent, usually conveyed through HTTP headers, is generally required from the remote server. This safeguard prevents potential misuse of data and reinforces the principle of data integrity.
- **Limited Network Access** JavaScript can make network requests to the server from which the current page starts, but its ability to make random network requests is limited. This prevents potential abuse by ensuring that JavaScript is used responsibly and transparently, contributing to a more secure browsing experience.

These restrictions collectively contribute to a secure browsing environment, safeguarding users from potential privacy infringements and malicious activities.

## 5.1.2 Django

I use Django for the frontend. Basically, Django is a powerful and user-friendly web framework written in Python. It simplifies and speeds up the process of building web applications by providing a structured and organized development environment.

Django is designed to be beginner-friendly. Its syntax is clear, and it provides built-in tools that make common tasks straightforward. Django follows a structured architecture, dividing the application into models (which represent the data), views (which handle the user interface and user input), and templates (which define the presentation layer). Django's ORM allows developers to interact with databases using Python code rather than SQL queries. This simplifies database management and enhances code readability. Django includes a built-in, customized admin interface. It provides an easy way to manage and update your application's data without the need for additional coding.

## Security Features

Django prioritizes security, offering protection against common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Django provides a robust authentication system, simplifying the process of user login, registration, and password management.

Here is how Django simplifies these processes.

- **User Authentication:** Django's authentication system allows users to create an account and log in securely. It provides built-in views and forms for handling user authentication, making it easy to implement login functionality without needing to write custom code.
- **User Registration:** Django simplifies the process of user registration by providing built-in forms and views for handling user registration. Developers can customize these forms to collect additional user information if needed, such as email addresses or profile details.
- **Password Management:** Django includes features for managing user passwords securely. It provides mechanisms for password hashing and salting, protecting user passwords from being compromised in the event of a security breach. Additionally, Django includes built-in views and forms for password reset and password change functionality, allowing users to reset their passwords if they forget them or change them if needed.
- **User Permissions and Groups:** Django's authentication system includes support for user permissions and groups, allowing developers to define fine-grained access control rules for different parts of the application. This enables administrators to manage user permissions easily and restrict access to sensitive areas of the application based on user roles.
- **Integration with Django's Admin Interface:** Django's authentication system seamlessly integrates with Django's built-in admin interface, allowing administrators to manage user accounts, groups, and permissions through a user-friendly interface. This makes it easy to perform administrative tasks such as creating new user accounts, resetting passwords, or managing user permissions without writing any custom code.

The URL dispatcher enables developers to define patterns for URLs and map them to corresponding views or functions in their Django application. This means that when a user visits a specific URL, Django knows which view function to call to handle the request, and developers can organize their application's URL



structure in a logical and hierarchical manner. This makes it easier to understand and manage the routing of requests within the application. The URL dispatcher seamlessly integrates with Django's view system, allowing developers to specify which view function should be called to handle each URL pattern. This tight integration streamlines the development process and promotes code reusability.

Django's middleware components play a crucial role in intercepting and processing requests globally before they are passed on to the views. This functionality proves invaluable for executing tasks such as authentication, security checks, and much more. By inserting middleware into the request-response processing pipeline, developers can intercept incoming requests at various stages of the process and perform necessary actions or modifications.

Django's template system allows developers to create dynamic HTML by embedding Python-like expressions. This separation of code and presentation enhances maintainability. Django has a vibrant and supportive community. It also provides comprehensive documentation, making it easy for developers to learn and find solutions to common problems.

In summary, Django is a web framework that empowers developers to create robust and secure web applications efficiently. Its user-friendly design, structured architecture, and built-in features make it an excellent choice for both beginners and experienced developers. [24]

### **5.1.3 Rosetta**

Rosetta is a powerful translation tool designed to facilitate the translation of text into multiple languages. It operates by utilizing text enclosed within translation tags, which act as markers to identify the content intended for translation. With Rosetta, developers can easily mark specific sections of text within their codebase using these translation tags. These tags serve as indicators for the Rosetta tool, signaling which parts of the text need to be translated into various languages. Once the text is properly tagged, Rosetta can then process the marked content and generate translations for each language specified. This enables developers to efficiently localize their applications or websites for different regions and language preferences.

The screenshot shows the Rosetta Home interface with a filter set to 'PROJECT'. It displays four language sections: Italian, English, France, and German. Each section contains a table with the following data:

Language	Application	Progress	Messages	Translated	Fuzzy	Obsolete	File
Italian	App	100%	4	4	0	0	/home/webserver/telaProjectDev/app/locale/it/LC_MESSAGES/django.po
	Frontend	0%	171	0	0	0	/home/webserver/telaProjectDev/frontend/locale/it/LC_MESSAGES/django.po
	Frontend	100%	1	1	0	0	/home/webserver/telaProjectDev/frontend/locale/it/LC_MESSAGES/djangojs.po
English	App	0%	4	0	0	0	/home/webserver/telaProjectDev/app/locale/en/LC_MESSAGES/django.po
	Frontend	100%	176	176	0	1	/home/webserver/telaProjectDev/frontend/locale/en/LC_MESSAGES/django.po
	Frontend	0%	1	0	0	0	/home/webserver/telaProjectDev/frontend/locale/en/LC_MESSAGES/djangojs.po
France	App	0%	4	0	0	0	/home/webserver/telaProjectDev/app/locale/fr/LC_MESSAGES/django.po
	Frontend	100%	171	171	0	1	/home/webserver/telaProjectDev/frontend/locale/fr/LC_MESSAGES/django.po
German	Application	Progress	Messages	Translated	Fuzzy	Obsolete	File

Figure 5.1: Rosetta Home

In Figure 5.3, with Rosetta’s translation capabilities, I can streamline the localization process and ensure that the content is accessible and understandable to a global audience. I chose four languages: English, French, German, and Italian.

## 5.2 Back-End

For the backend of the online booking system, I’ll focus on building a robust and secure infrastructure to handle user authentication, data storage, and transaction processing. To achieve this, I will use a combination of server-side programming languages such as Node.js, and JavaScript (JavaScript is used for both frontend and backend), depending on the project requirements and my expertise. These languages offer scalability, performance, and versatility, making them ideal for developing backend services. I am using MangoDB to store user information, ticket details, and transaction records securely.

Here are the tools and technologies which are used for frondend.

### **5.2.1 Node.js**

I am using Node.js, which is a JavaScript runtime built on the V8 JavaScript engine. It allows developers to run JavaScript code on the server side, enabling the development of scalable and high-performance web applications. This opens up a wide range of possibilities, including creating web servers, handling HTTP requests and responses, interacting with databases, and much more. It is designed to handle large numbers of concurrent connections efficiently, making it well-suited for building real-time applications and APIs that need to handle a lot of traffic.

Node.js allows developers to use JavaScript for server-side scripting, making it possible to use the same language for both client-side and server-side development. Node.js is designed to be non-blocking and event-driven. It uses an asynchronous, non-blocking I/O model that enables handling multiple connections simultaneously without waiting for one to complete before moving on to the next. Node.js operates on a single-threaded event loop. This architecture is efficient for handling a large number of concurrent connections, making it suitable for real-time applications. Node.js comes with NPM, a package manager that allows developers to easily manage and install third-party libraries and modules. NPM is a vast ecosystem of open-source libraries. Node.js encourages the use of modular programming. Developers can create reusable modules and package them for distribution through NPM.

Node.js is cross-platform and runs on various operating systems, including Windows, macOS, and Linux, providing flexibility for developers. Node.js has a large and active community of developers. This community contributes to the continuous improvement of Node.js, provides support through forums, and shares a wealth of knowledge and resources. There are numerous libraries and frameworks built on top of Node.js to simplify common tasks. Express.js is one of the most popular frameworks for building web applications with Node.js. Node.js is well-suited for developing real-time applications like chat applications, online gaming, and collaborative tools where low-latency communication is crucial. Node.js is excellent for handling streaming data. It can be used for building applications that require real-time updates or processing large amounts of data in chunks.

Node.js has gained significant popularity in the web development community due to its performance, scalability, and versatility. It is widely used for building web servers, APIs, and various types of networked applications. [25]

### **5.2.2 GitHub**

GitHub is a web-based platform that plays a crucial role in modern software development. It provides a version control system, primarily using Git, which allows

developers to manage and track changes to their code over time. The platform is widely utilized for hosting repositories, serving as a centralized location for source code, documentation, and collaborative work.

In GitHub, a repository (repo) acts as a container for a project, housing all relevant files and preserving the entire history of changes. Repositories can be either public or private, depending on whether the code is meant for public view or restricted access. Developers can create, clone, and fork repositories to initiate and contribute to projects.

Collaboration is a key feature of GitHub, achieved through mechanisms like pull requests and issues. Pull requests enable developers to propose changes, discuss modifications, and ultimately merge them into the main codebase after reviews. Issues are employed for tracking tasks, enhancements, and bugs, providing a space for discussion and coordination.

GitHub supports branching, allowing developers to work on specific features or fixes in isolation through branches. Once changes are complete and tested, they can be seamlessly merged back into the main branch, ensuring a streamlined development process.

GitHub Actions is a feature that automates workflows such as building, testing, and deploying code. This automation enhances the efficiency of development processes and supports a DevOps approach.

GitHub Pages is another notable feature, enabling users to publish static web content directly from their repositories. It is often used for hosting project documentation, personal blogs, or simple websites.

The platform also incorporates community and social features. Developers can follow others, star repositories, and contribute to open-source projects. GitHub serves as a hub for discussions, code reviews, and collaboration.

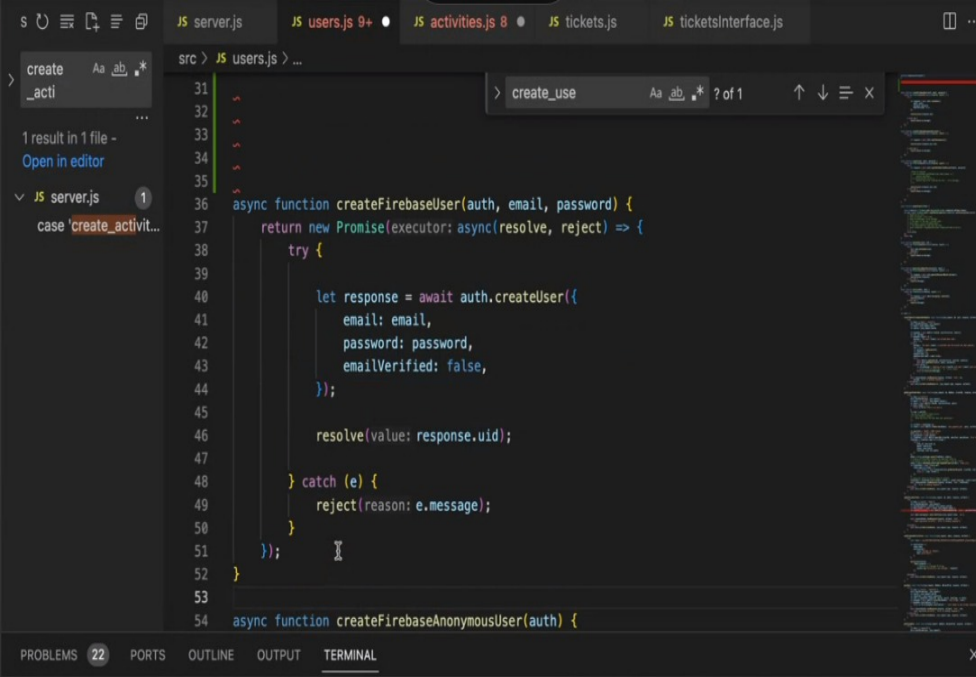
Overall, GitHub has become an integral part of the software development landscape, fostering collaboration, providing version control, and offering a platform for open-source contributions. Its user-friendly interface, powerful features, and integration capabilities make it a central hub for developers, both individual and organizational. [26]

### 5.2.3 FireBase

Firebase is a comprehensive mobile and web application development platform developed by Google. It provides a range of tools and services to help developers build and deploy applications quickly. Firebase includes features for various aspects of app development, such as authentication, real-time databases, hosting, cloud functions, storage, and more.

Firebase authentication provides a simple way to implement user authentication in applications. It supports various authentication methods, including email/password, social identity providers (like Google, Facebook, and Twitter), and anonymous authentication.

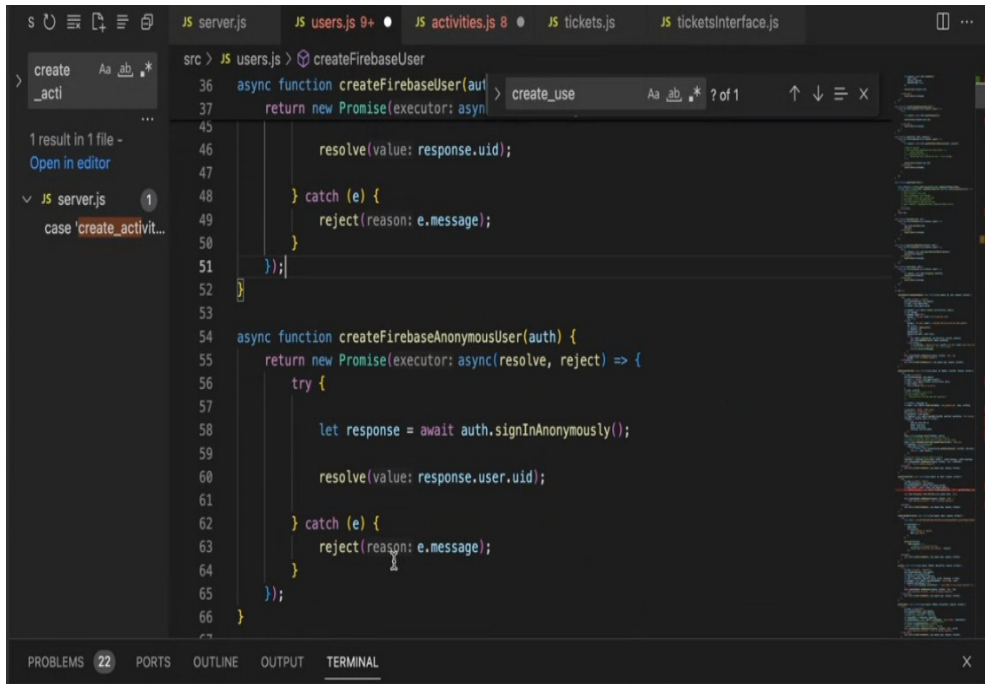
Firebase Cloud Functions allow developers to run backend code in response to events triggered by Firebase features and HTTPS requests. It enables serverless computing and is often used for tasks such as sending notifications, processing data, or integrating with third-party services. [27]

A screenshot of a code editor interface, likely Visual Studio Code, showing a JavaScript file named 'users.js'. The editor is in dark mode. The main window displays the code for an asynchronous function 'createFirebaseUser'. The function takes three arguments: 'auth', 'email', and 'password'. It returns a new Promise with an executor function. Inside the executor, there is a try-catch block. The try block contains a call to 'auth.createUser()' with the provided email and password, and 'emailVerified' set to false. The result is stored in 'response' and 'response.uid' is returned. The catch block catches any error 'e' and rejects the promise with 'e.message'. Below the main function, the start of another function 'createFirebaseAnonymousUser' is visible. The editor's interface includes a search bar at the top with the query 'create\_use', a file explorer on the left showing the project structure, and a terminal at the bottom.

**Figure 5.2:** Firebase Create User

In Figure 5.1, I have established Firebase for authentication using email and password. Firebase Authentication is a service provided by Google that allows users to sign in to the app using email addresses and passwords. This provides a convenient

and secure way for users to access your application's features and services. Firebase Authentication handles the entire authentication flow, including user registration, email verification, password resets, and sign-in.



```
src > JS users.js > createFirebaseUser
36 async function createFirebaseUser(auth) {
37   return new Promise(executor: async(resolve, reject) => {
45     resolve(value: response.uid);
46   });
47 }
48 } catch (e) {
49   reject(reason: e.message);
50 }
51 });
52 }
53 }
54 async function createFirebaseAnonymousUser(auth) {
55   return new Promise(executor: async(resolve, reject) => {
56     try {
57       let response = await auth.signInAnonymously();
58       resolve(value: response.user.uid);
59     } catch (e) {
60       reject(reason: e.message);
61     }
62   });
63 }
64 }
65 }
66 }
```

Figure 5.3: Firebase Anonymous User

In addition to setting up authentication with email and password, I have also implemented anonymous user authentication in the Firebase service. Anonymous authentication allows users to access certain features of the application without requiring them to create an account or provide any personal information.

## 5.2.4 MongoDB

I am using MongoDB, a popular NoSQL database system. MongoDB is a document-oriented database that stores data in flexible, JSON-like documents, known as BSON (Binary JSON). It has no static structure, and I stored all information, activities, and photos in MongoDB. It is designed to store, query, and process large amounts of data in a flexible, schema-less format. MongoDB is often used in modern web development, particularly for applications where flexible and scalable data storage is essential.

Here are some key aspects of MongoDB.

- **Document-Oriented:** MongoDB stores data in flexible, JSON-like BSON (Binary JSON) documents. Each document can have a different structure, allowing for more dynamic and evolving data models compared to traditional relational databases.
- **Collections and Documents:** In MongoDB, data is organized into collections, which are similar to tables in relational databases. Each collection contains documents, and each document is a set of key-value pairs.
- **Schema-less:** MongoDB is schema-less, allowing developers to insert data without a predefined structure. This flexibility makes it easier to handle dynamic and evolving data models.
- **JSON Format:** MongoDB stores data in BSON format, a binary representation of JSON. This format supports various data types, including strings, numbers, arrays, and nested documents.
- **Query Language:** MongoDB uses a rich query language that supports a wide range of queries, including filtering, sorting, and aggregation. It also supports full-text search and geospatial queries.
- **Indexes:** MongoDB allows developers to create indexes on fields to improve query performance. Indexes can be created on single fields, compound fields, and even arrays.
- **Scalability:** MongoDB is designed to scale horizontally by distributing data across multiple servers. This makes it suitable for handling large amounts of data and high-volume traffic.

MongoDB offers a robust aggregation framework designed for efficient data transformation and analysis, incorporating diverse operators and stages to manipulate and process data effectively. The platform supports replication, allowing data

mirroring across multiple servers to ensure high availability and fault tolerance. MongoDB is a sharding feature that distributes data across numerous servers or clusters, facilitating horizontal scaling for enhanced performance. With a vibrant and engaged community, MongoDB benefits from a rich ecosystem comprising drivers, libraries, and tools supporting various programming languages and platforms. MongoDB Atlas, a fully managed cloud-based database service, streamlines database management, scaling, and security, reinforcing its position as a widely adopted choice for applications such as content management systems, e-commerce platforms, and real-time analytics. The platform's adaptability and scalability make it particularly favored among developers handling dynamic and swiftly evolving data. [28]

## **5.3 Other Tools**

In addition to the core technologies for frontend and backend development, I'm using several other tools for debugging, request redirection, and other essential tasks.

### **5.3.1 Visual Studio Debugger**

Visual Studio is an integrated development environment (IDE) developed by Microsoft for software development. It provides a comprehensive set of tools and features to support the entire development lifecycle, from writing and debugging code to testing and deployment. Visual Studio includes a powerful code editor with features like syntax highlighting, IntelliSense (auto-completion), and code navigation, enhancing productivity during code development. The integrated debugger allows developers to identify and fix issues in their code efficiently. It supports features like breakpoints, watch windows, and step-by-step debugging. Visual Studio includes visual designers for creating user interfaces, database schemas, and other visual elements. These designers allow developers to design components visually and generate corresponding code.

Visual Studio supports multiple programming languages, including C#, Visual Basic, C++, F#, and more. It also provides tools for web development, mobile app development, and cloud-based applications. Visual Studio can be extended through a rich ecosystem of extensions and add-ons available in the Visual Studio Marketplace. These extensions provide additional functionality and tools. Visual Studio supports web development with features for building ASP.NET applications, HTML, CSS, and JavaScript. It also has tools for developing cloud-based applications with Azure integration. Visual Studio includes tools for collaborative development, such as version control integration with Git and Team Foundation



Server (TFS), as well as features for code reviews and collaboration within development teams.

Visual Studio offers a suite of testing tools, including unit testing and performance testing, to help ensure the quality of code. Visual Studio facilitates the build and deployment process with features like project configurations, build configurations, and deployment profiles. It supports continuous integration and continuous deployment workflows. Visual Studio supports cross-platform development for platforms such as Windows, macOS, Android, and iOS. It includes tools like Xamarin for building cross-platform mobile applications. Visual Studio is available in different editions, including the free Visual Studio Community edition, suitable for individual developers and small teams, and the Visual Studio Professional and Enterprise editions, which offer additional features for larger teams and enterprises.

Overall, Visual Studio is a versatile and widely used IDE that caters to a broad range of development needs, making it a popular choice among developers for building diverse types of applications. [29]

### **5.3.2 Reverse Proxy NGNIX**

A reverse proxy is a server that sits between client devices and a web server, forwarding client requests to the web server and returning the server's responses to clients. Nginx is a popular web server and reverse proxy server that is often used to improve the performance, security, and scalability of web applications.

Setting up a reverse proxy with Nginx involves configuring the Nginx server to act as an intermediary between client devices and a backend server. In the Nginx configuration file, a server block is defined with the specified domain, listening port (commonly 80 for HTTP), and a location directive for the root path. The critical part is the `proxy_pass` directive within the location block, where the IP address and port of the backend server are specified. Additional `proxy_set_header` directives are used to preserve important information like the client's IP address. After making these configurations, Nginx is reloaded to apply the changes. This basic setup allows Nginx to efficiently handle client requests, forwarding them to the backend server and returning the server's responses to clients. Depending on your requirements, you may also need to consider SSL/TLS configuration, logging, error handling, and security measures. Always consult the official Nginx documentation for comprehensive guidance and updates. [30]

### 5.3.3 Chrome Debugger

The Chrome Debugger is a powerful tool integrated into the Google Chrome browser that aids developers in debugging and analyzing web applications. Accessible through the browser's Developer Tools, the Chrome Debugger provides a comprehensive set of features for inspecting and troubleshooting client-side code written in languages such as JavaScript, HTML, and CSS. Developers can set breakpoints, step through code execution, and examine variables, allowing for a detailed examination of the application's behavior. It also provides real-time insights into network activity, performance profiling, and memory usage, helping to identify bottlenecks and optimize the application. With its user-friendly interface and extensive capabilities, the Chrome Debugger is an indispensable resource for web developers striving to create efficient and error-free web applications.

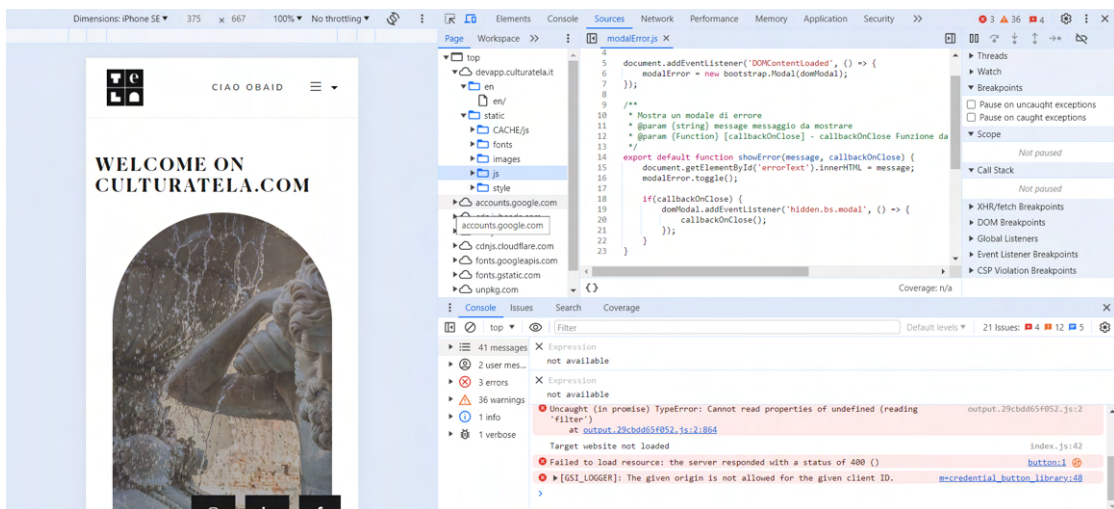


Figure 5.4: Chrome Debugger

In Figure 5.4, on the Chrome debugger, I can utilize the console to check for errors and experiment with different logics without disrupting the actual codebase. The console serves as a valuable tool for debugging and testing purposes, allowing developers to interactively evaluate expressions, execute code snippets, and log messages for analysis.

### 5.3.4 Flutter

Flutter is an open-source UI software development kit created by Google. It is designed for building natively compiled applications for mobile, web, and desktop from a single codebase. Flutter uses the Dart programming language, also developed by Google, which is known for its simplicity and efficiency.

One of Flutter's key features is its fast development cycle, which allows developers to see changes instantly with its hot reload feature. This enables rapid experimentation, iteration, and debugging, leading to faster development times.

Flutter also boasts a rich set of customizable widgets, allowing developers to create beautiful, expressive user interfaces that can adapt to various screen sizes and pixel densities. These widgets follow Material Design for Android and Cupertino for iOS, ensuring a native look and feel across platforms.

Additionally, Flutter provides access to a wide range of native device features and APIs through its extensive plugin system. This allows developers to easily integrate platform-specific functionality such as camera access, geolocation, and sensors into their Flutter apps.

Flutter offers a powerful and versatile framework for building cross-platform applications with high performance, expressive UIs, and a fast development cycle. Its growing community and strong support from Google make it an attractive choice for developers looking to build modern, responsive apps for various platforms. I use the Flutter app to show the different festival details.

# Chapter 6

## Implementation

In the implementation phase, we turn the design into actual code using programming languages. We test the code to make sure it does what the design needs it to do. This involves understanding the design, picking the right programming language, setting up a place to work on the code, and testing it. Our goal is to follow the design plan, write code that is easy to understand and manage, and make sure it works well before moving forward.

The modules that I am going to implement.

- New way to sell tickets with automatic machines, users can purchase the tickets on-site with totem machines
- QR code for payment redirections
- QR code scanner for mobile phones
- Chat AI integration to page
- Online consultation with the app
- FAQ page
- New ticket layout
- Intro js to give the administrative users a guided tour of the new functionalities in our application. It provides step-by-step interactive tooltips that highlight and explain each new feature, making it easier for them to understand and use the updates.
- Content for end users such as AI videos
- Tracking of users of Google Pixels and Meta
- Android/IOS App for displaying content like festival details and concerts.

The main components of our booking system are as follows.

## 6.1 Frontend

### 6.1.1 Platform/Server

- Registration, Login and Authentication
- Payment through NEXI, debit cards, Apple Pay, Google wallet
- Support services for the registration of end users
- Support services for the registration of operators
- High reliability/redundancy server architecture

In the platform/server domain, several essential functionalities are being developed to ensure a robust and user-friendly experience for both end users and operators. Firstly, a comprehensive registration, login, and authentication system is being implemented to enable users to create accounts securely, log in seamlessly, and access their accounts without encountering any authentication issues. Additionally, the integration of various payment methods, including NEXI, debit cards, Apple Pay, and Google Wallet, will facilitate smooth and secure transactions for users when purchasing tickets or making payments on the platform. Furthermore, dedicated support services will be provided to both end users and operators to assist them with the registration process and address any queries or concerns they may have while using the platform. To ensure uninterrupted availability and performance, a high-reliability and redundant server architecture is being deployed, mitigating the risk of downtime due to hardware failures or unexpected outages. These functionalities aim to enhance the reliability, security, and usability of the platform, thereby providing a seamless experience for all stakeholders involved.

### 6.1.2 Frontend website for End-User

- Application infrastructure with login/registration
- Ticket purchasing
- Payments through NEXI
- Print the tickets
- Cancellation/changing of tickets
- Refund of payments
- AI chat for information
- Online consultation

### 6.1.3 Frontend website for Operator

- Application infrastructure with login/registration
- Status of payments
- Status of tickets
- Verify the mechanism for validation/tickets
- Analytics online booked registry by visiting hours
- Training of AI chat with new queries
- Box office management
- Management Area
- QR-code verification
- Reporting system
- API / Web Service to enable integration

### 6.1.4 Help-Desk

- User can get help regarding serious issues by mail
- Back office support

## 6.2 Application Functionalities (Frontend)

### 6.2.1 Main/Home page

Culturatela.com is a dynamic, responsive platform. The homepage operates dynamically, tailoring its content based on user preferences, including chosen destinations or events. All elements, including images, text, and explanations, are fetched from the server whenever a service is invoked to retrieve information. From the frontend perspective, each segment is designed as a component, ensuring complete responsiveness across various devices.

- The booking system encompasses a diverse range of offerings, allowing users to make reservations for various experiences, including museums, events, tours, and study rooms. This comprehensive platform provides a seamless and convenient way for users to plan and secure their participation in cultural and educational activities. Whether it is exploring museums, attending events,

joining guided tours, or reserving study spaces, the booking system streamlines the entire process, enhancing accessibility and engagement for users across different domains.

- The operations related to ticket obliteration, control, and verification are restricted to privileged users with authorized access. These specific functionalities involve sensitive actions, and access is limited to ensure secure and controlled management of tickets. Only users with the appropriate privileges are granted access to this section, emphasizing the importance of maintaining the integrity and security of ticket-related processes.

```
1 // Query used to get the homepage.
2
3 { lucca_req:
4   {
5     type: 'get_homepage',
6
7   }
8 }
9
```

I used a request query to get the main home page. If the server response is successful, then it redirects to the homepage.

```
1 {
2   status: 'done',
3   data: [
4     {
5       title: 'Torre Guinigi',
6       type: 'Museo',
7       mainPhotoObj: { id: '5f9459eb8d5a1f35f2918b7e', type: 'image/jpeg' },
8       activityId: '5f9459eb8d5a1f35f2918b81'
9     },
10    {
11      title: 'Orto Botanico',
12      type: 'Museo',
13      mainPhotoObj: { id: '5f945a8e583da435e2c93754', type: 'image/jpeg' },
14      activityId: '5f945a8e583da435e2c93757'
15    },
16    {
17      title: 'Torre delle Ore',
18      type: 'Museo',
19      mainPhotoObj: { id: '5f945ae66f7b3835ebc4c775', type: 'image/jpeg' },
```

```

20     activityId: '5f945ae66f7b3835ebc4c777'
21   },
22   {
23     title: 'Torre Guinigi',
24     type: 'Museo',
25     mainPhotoObj: { id: '5f9459eb8d5a1f35f2918b7e', type: 'image/jpeg' },
26     activityId: '5f945d19df86a12674cec3e6'
27   },
28 ],
29 message: null
30 }
31

```

## 6.2.2 Operations/Functions

Users have the option to search for specific events or tours within the system. When they enter their desired search term, such as a museum or event name, they are directed to the search results page, where they can view all relevant matches. For instance, if a user selects "Lucca Puccini Museum," the search results will display all upcoming events hosted there.

Each event listing provides detailed information, including the number of available tickets, opening hours, and other important details. Users can easily browse through the listings, select the event they are interested in, and proceed to book and purchase tickets. This streamlined process allows users to swiftly locate their desired event and smoothly navigate through the steps to reserve and buy tickets.

## 6.2.3 Login/Sign-up

I provide users with two options to begin using the platform: signing up or logging in. For users who have not created an account yet, they can choose to log in either using their email or through Google Login. This offers flexibility, enabling users to use the method that is most convenient for them to access the platform. Whether they prefer traditional email-based login or the convenience of using their Google account, users have the freedom to choose the method that suits them best.

I used a request query to get the login page.

```

1   {
2     {
3       type: 'sign_in',
4       email : '*****@**.*',
5       pwd : '*****',

```



```
6     }
7
8
```

If the server response is successful, then it redirects to the login page.

```
1
2
3
4 data: {
5     name: 'Matteo',
6     userId: 'gGgCUC01UDM1GnBuSTWk9deeTCx2',
7     superUser: true,
8     numProd: 1,
9     fiscalCode : 'TRNMTT...',
10    deskUser: {
11        isDesk: true,
12        activities : [
13            { activityId: '5fa0973ea7435a68e7bf26d7',
14 title: 'Orto botanico'
15        },
16        {
17            activityId: '5fa09b3e20226c68d74b95a8',
18            title: 'Torre Delle Ore'
19        }
20    ],
21    },
22    message: 'User signed correctly'
```

- **userId** key shows the special ID that belongs uniquely to a user.
- **superUser** key is a simple switch – it is either true or false. If it is true, it means that the user has special permissions.
- **deskUser** key is interesting, it holds a bunch of information in a JSON format. This information might include various details about a user related to a desk.

- The "isDesk" key is a boolean, which means it can be either true or false. If it is set to false, it indicates that the user does not have checkout permissions. In this case, they are redirected to the homepage. On the other hand, if it is set to true, it means the user has cashier permissions and the mentioned information is sent or made available. Essentially, this boolean value determines whether a user has the ability to perform checkout actions or not.
- The "title" key is a string that holds the name or title of an activity. This information is meant to be displayed somewhere on the management system,

providing a clear and descriptive label for the respective activity. It helps users easily identify and understand the nature or purpose of the activity within the system.

- The "sellingDays" key is an array of JSON objects. Each individual JSON object within this array likely contains information about specific days related to selling activities. The details inside each JSON object could include various pieces of information, such as dates, availability, pricing, or any other relevant details tied to the selling aspect of the system.
- The "day" : timestamp - indicates the specific day for which time slots exist for selling. This timestamp is used to highlight and reference the days on the calendar within the system. It helps in organizing and visualizing the availability or events associated with selling on particular days.

If the "sellingDays" array is empty, it means that there are no days designated for sales. In this case, the calendar must be rendered entirely non-clickable, meaning that there are no available time slots or events for selling during the specified time period. Users interacting with the calendar will not be able to select or engage with any days since there are no sale activities on those dates.

- The "sellingSlots" key is an array of JSON objects. Each JSON object within this array represents a single time slot for today's date. These JSON objects likely contain details specific to each time slot, such as start and end times, availability, pricing, or any other relevant information pertaining to the selling activities happening during those time slots.
- slot\_start : timestamp, indicates the start time of a specific time slot. This timestamp provides information about when the slot starts, helping to organize and display the selling activities.
- slot\_end: timestamp, indicates the end time of a specific time slot. This timestamp provides information about when the slot concludes, completing the details for the selling activities during the specific time period.
- num\_tickets: it represents an integer that denotes the number of tickets that are still available for a specific time slot. The integer value can range from 0 to N, where N represents the total number of tickets available for that specific time slot. This information helps users understand the remaining availability for ticket purchases during the indicated period.

If the "sellingSlots" array is empty, it means that there are no time slots available on today's date. In this case, the system should be displaying the information,

indicating to users that there are no available time slots for selling activities. Users should be informed with a message to show that there are no selling slots available today.

The error management.

- The password is invalid -> The password entered is incorrect.
- There is no user -> The email entered does not exist.

## 6.2.4 Google-Login

The first step is to check if the user already exists in MongoDB or not. If any inaccurate or missing information is identified, we update the user information in our database. In this way, user data is correct and complete in our system.

```
1
2 {
3   "type":
4   "is_external_sign_in_complete",
5   "token": 5234iou5h08732456 ,
6   "provider": "google",
7 }
8
```

```
1
2 {
3   status: 'done',
4   data: {
5     isComplete: [bool]
6   },
7   message: "",
8   }
9
```

The value of "isComplete" helps me determine if the external provider mentioned in the request has provided all the necessary information to create the MongoDB user in the token. If "isComplete" is true, it means I have all the required details, and I can proceed with creating the user. If it is false, I may need to ask the user for additional information to ensure the completeness of the data before creating the MongoDB user in the token.

JSON query for the google login.

```
1
2 JSON - GOOGLE
3
4 {
5     "type": "google_sign_in",
6     "token": "324hjb2lg4",
7     // additional info required if not provided by the token
8     // (at the moment there is the dateOfBirth field): it must be replaced
9     // with a field arguments which will be
10    // an object containing various required fields
11    "dateOfBirth": new Date()
12 }
13
14 If the server response is successful the status will be done.
15
16 {
17     status: 'done',
18     data: {
19         name: 'Matteo',
20         userId: 'gGgCUC01UDM1GnBuSTWk9deeTCx2',
21         superUser: true,
22         numProd: 1,
23         fiscalCode: "TRNMTT",
24         deskUser: {
25             isDesk: true,
26             activities: [
27                 {
28                     activityId: '5fa0973ea7435a68e7bf26d7',
29                     title: 'Orto botanico'
30                 },
31                 {
32                     activityId: '5fa09b3e20226c68d74b95a8',
33                     title: 'Torre Delle Ore'
34                 }
35             ],
36             message: 'Google User signed correctly'
37         }
38     }
39 }
```

In Figure 6.1, Google Login is a great feature. It makes it easy for people to create an account and log in. It is also good for developers because I can use Firebase to add support for Google Sign-In. Firebase makes it simple to set up Google Sign-In without needing to do a lot of work. It is safe and handles things like managing users and keeping their information secure. Users find it easy to use, and developers find it easy to set up.

- **Install Pacakages:** First, we have to install the Google Sign-In package. This package helps us add Google authentication features to our application.
- **Import Modules and Components:** After installing the package, we import all the necessary modules and components from Google Signin. This step allows us to use the functions needed for Google login.

**Password reset function:**A password reset function allows users to regain access to their accounts when they forget their passwords. It starts with the user clicking a link to forget their password; this process involves email verification as well, where a unique link or code is sent for user identity confirmation. After clicking the link or entering the code, users are directed to a password reset page, where they can create a new password for their accounts. This functionality ensures an easy and secure way for users to recover their forgotten passwords.

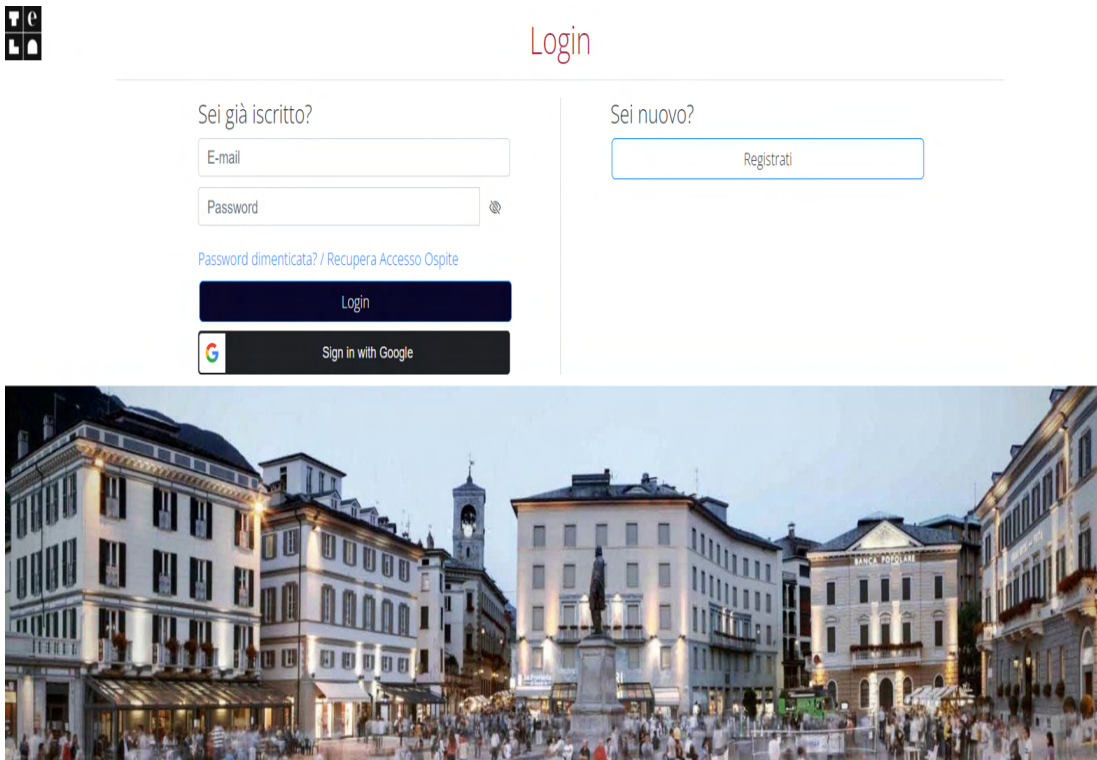


Figure 6.1: Login / Google-Login

### 6.2.5 Cart-Checkout

Once users have selected their tickets, they can add their order to the cart checkout page. In Figure 6.2, the cart checkout page marks the final step in purchasing

the tickets, offering users the option to proceed. After selecting their items and adding them to the cart, users move to the checkout to complete the transaction. During checkout, they review their chosen items, input payment details, and then proceed to the payment gateway to securely submit their payment. The checkout process is designed to be user-friendly, efficient, and secure, ensuring a seamless and satisfying end to the online shopping experience.

```
1 //Fetching data in the cart
2 //JSON Response
3
4 {
5     userId : 'N67In0k3qgUONGjJKSVeWPltTB92',
6     type: 'get_cart',
7 }
8
9 Esempio risposta server ( andata a buon fine )
10 {
11     status: 'done',
12     data: : [
13         {
14             slotId: '2404fff7-62f4-4b59-a17c-34e051abfa1f',
15             activityId: '5fa08e72f957344e1eb15238',
16             ticketsRequest: [
17                 { type: 'free', numTickets: 3 },
18                 { type: 'free', numTickets: 2 }
19             ],
20             prodId: 'd6234455-4ba7-4782-ba87-9d25b96df79f',
21             price: 500,
22             title: 'Biblioteca Civica Agorà',
23             start_slot: '2020-11-04T11:30:00.000Z',
24             end_slot: '2020-11-04T12:30:00.000Z',
25         },
26     ],
27     {
28         slotId: 'd4211036-8805-462e-bd1a-c91b28e912d9',
29         activityId: '5fa08e72f957344e1eb15238',
30         ticketsRequest: [ { type: 'free', numTickets: 2 } ],
31         prodId: 'b745b6fe-a158-447e-804d-168bcef7212f',
32         price: 500,
33         title: 'Biblioteca Civica Agorà',
34         start_slot: '2020-11-04T11:30:00.000Z',
35         end_slot: '2020-11-04T12:30:00.000Z',
36     },
37 ],
38 {
```

```
39     slotId: '2404fff7-62f4-4b59-a17c-34e051abfa1f',
40     activityId: '5fa08e72f957344e1eb15238',
41     ticketsRequest: [
42       { type: 'free', numTickets: 3 },
43       { type: 'free', numTickets: 2 }
44     ],
45     prodId: 'ca424424-c0c3-4d27-bc2c-97d4d2fe8919',
46     price: 500,
47     title: 'Biblioteca Civica Agorà',
48     start_slot: '2020-11-04T11:30:00.000Z',
49     end_slot: '2020-11-04T12:30:00.000Z',
50
51   },
52   {
53     slotId: 'd4211036-8805-462e-bd1a-c91b28e912d9',
54     activityId: '5fa08e72f957344e1eb15238',
55     ticketsRequest: [ { type: 'free', numTickets: 2 } ],
56     prodId: '42aab6e8-48ae-406a-95a3-f69e12795c66',
57     price: 500,
58     title: 'Biblioteca Civica Agorà',
59     start_slot: '2020-11-04T11:30:00.000Z',
60     end_slot: '2020-11-04T12:30:00.000Z',
61
62   }
63 ],
64 message: null
65 }
66
```

When returning products from the cart, each product is associated with a specific ID sent through the "prodId" attribute. This ID serves as a reference to remove particular products from the cart. By including the prodId the system can identify and process the removal of each specified item, and it ensures that the cart is updated according to the user's desires. This approach allows for a precise and efficient adjustment of the cart contents based on the provided prodId's.

1. expected 1 -> Error fetching the cart. Add the products back.

If you receive this error, you must empty your cart automatically by sending the associated request. This is because it means that there has been tampering with the data in the cart (the activity ID or slot ID does not exist in the databases).

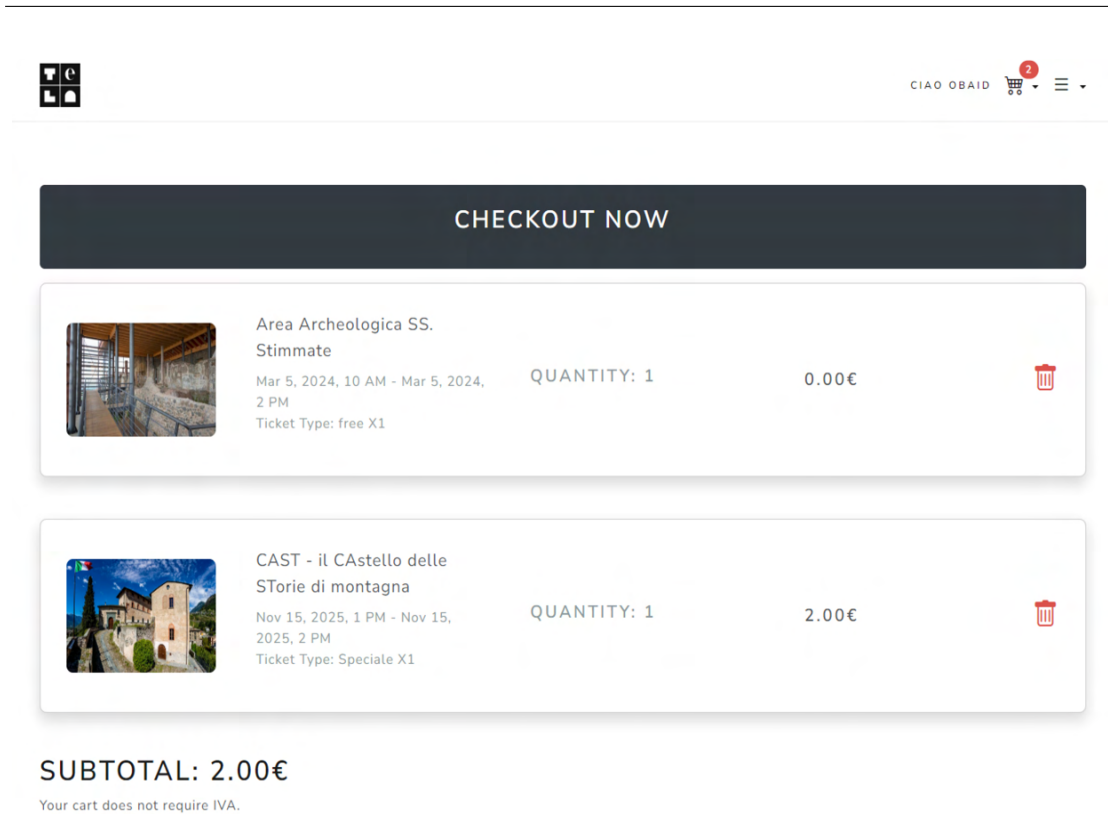


Figure 6.2: Cart Checkout

### 6.2.6 Deleting-Item

In Figure 6.3, deleting an item from the cart refers to the action of removing a selected product from the shopping cart during an online. Users typically have the option to delete items they no longer wish to purchase before checkout or empty the cart entirely. This feature provides flexibility and control to users, allowing them to refine their purchase choices before completing the transaction. The deleted items are then discarded from the cart.

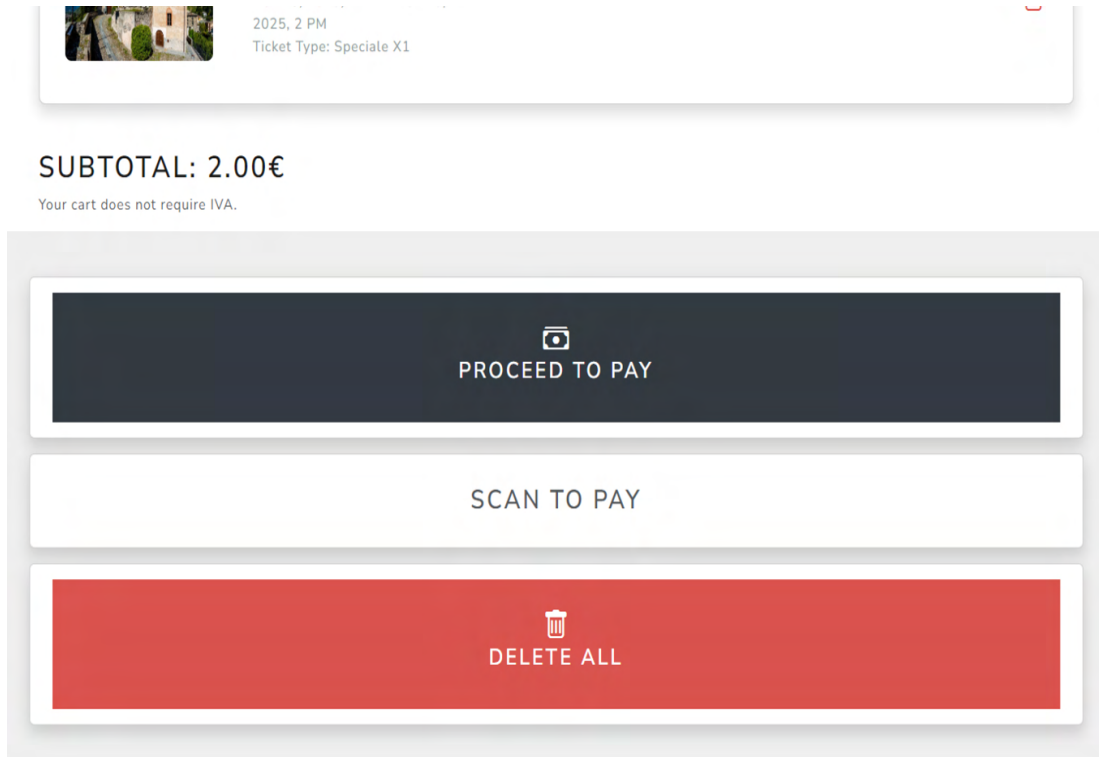
```

1  JSON Response
2      {
3      type: 'empty_cart',
4      userId : 'gGgCUC01UDM1GnBuSTWk9deeTCx2',
5      }
6
7  If the server response is successful the status will be done.
8  {
9  "status": "done",
10 "data": null,
11 "message": "Cart emptied correctly"

```



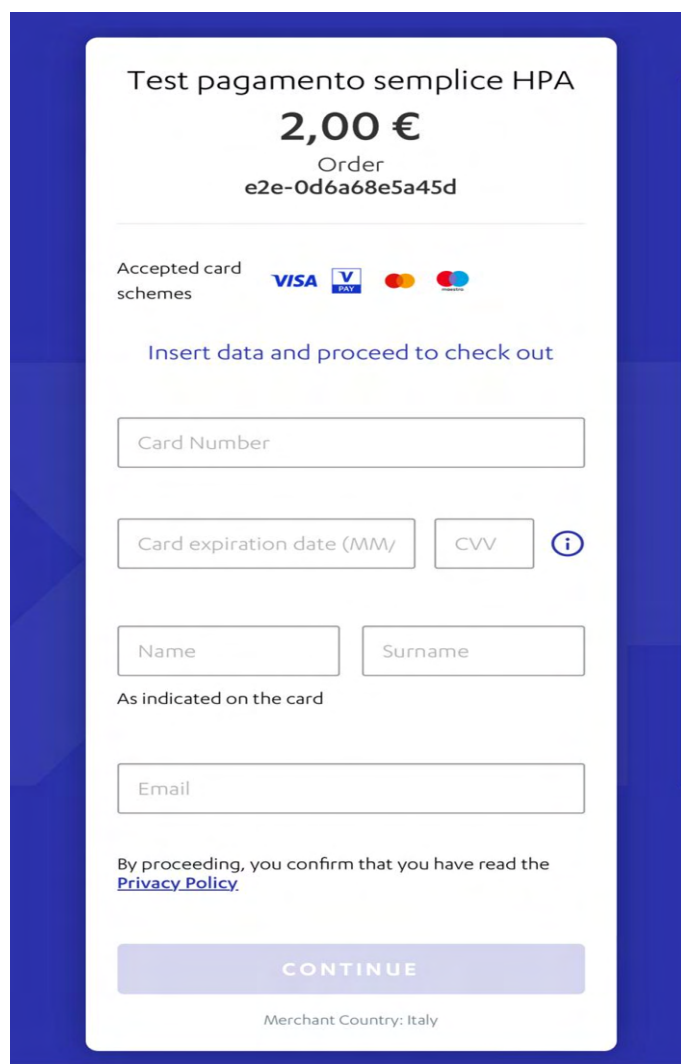
12 }  
13  
14



**Figure 6.3:** Proceed to Pay / Delete All

## 6.2.7 Payment

The payment process occurs directly on the provider's website, where it is the client's responsibility to redirect to the NEXI page that is received from the server. The issuance of tickets takes place when the backend receives a notification from NEXI, which indicates that the payment has been successfully processed. This flow ensures that ticket issuance is triggered only after the payment confirmation is received from the payment provider, and it's a secure transaction process. In Figure 6.5, users will be redirected to the NEXI payment page to pay for the ticket. After paying, users will receive the ticket.



The image shows a payment form titled "Test pagamento semplice HPA" with a transaction amount of "2,00 €". The order ID is "e2e-0d6a68e5a45d". Below the title, it lists "Accepted card schemes" with logos for VISA, V PAY, and Mastercard. A blue link says "Insert data and proceed to check out". The form contains several input fields: "Card Number", "Card expiration date (MM/)" and "CVV" (with an information icon), "Name" and "Surname" (with the note "As indicated on the card"), and "Email". At the bottom, there is a "CONTINUE" button and the text "Merchant Country: Italy".

**Figure 6.4:** Payment through NEXI

### 6.2.8 Pay through QR-Code

For security purposes, I generated a dynamic payment link in the form of a QR code, which I showed in Figure 6.5. With this approach, users can make payments in public spaces or on large screens. By scanning the QR code, users can proceed with the payment process without exposing their sensitive bank details on public displays. This method adds an extra layer of security, ensuring a more protected transaction experience, especially in shared or public environments.

This approach offers several benefits.

- **Enhanced Security:** By using dynamic QR codes, I ensure that each transaction has a unique identifier, reducing the risk of unauthorized access to or interception of sensitive information.
- **Convenience:** Users can initiate payments quickly and easily by scanning the QR code on their smartphones. This eliminates the need for manual input of payment details and streamlines the transaction process.
- **Privacy Protection:** With the QR code method, users can complete transactions without revealing their bank details on public displays. This protects their privacy and reduces the risk of potential identity theft or fraud.

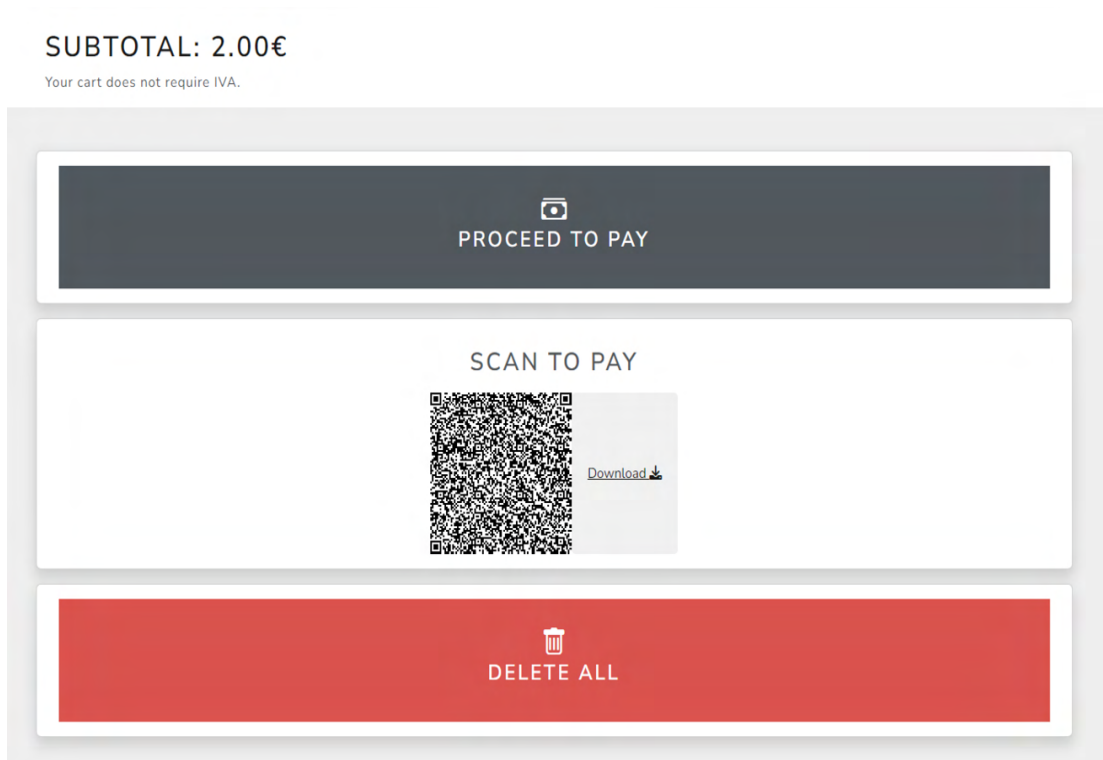


Figure 6.5: Pay with QR-Code

### 6.2.9 ChatAI

I added an AI chat feature to help users with their questions anytime, day or night, without any interruption. I used an advanced model called GPT-3.5 Turbo, which I fine-tuned to understand and respond to users better. To train this AI, I relied on a machine learning library called scikit-learn (Sklearn). By using Sklearn, I could test and train the AI models effectively.

With this approach, the AI chat can provide quick and accurate answers to users questions. It can handle various types of queries and improve the overall user experience. Whether users need help with ticket information, events, or anything else, the AI chat is there to assist them efficiently.

#### Install an OpenAI library

```
1 pip install -U openai
```

## Prepare the Dataset:

```

1 import pandas as pd
2 df = pd.read_csv("dataset.csv")
3 df.head()

```

## Format the Data

I organized the data for GPT-3.5 in a specific format. Each message is stored as an object within an array, and each object is assigned a role, which could be 'system', 'user', or 'assistant'. This structured format helps GPT-3.5 better understand the context of the conversation, enabling it to generate more accurate and relevant responses.

```

1 def convert_to_gpt35_format(dataset):
2     fine_tuning_data = []
3     for _, row in dataset.iterrows():
4         json_response = '{"Top Category": "' + row['Top
5             Category'] + '", "Sub Category": "' + row['Sub Category'] + '"}'
6         fine_tuning_data.append({
7             "messages": [
8                 {"role": "user", "content": row['Support Query']},
9                 {"role": "system", "content": json_response}
10            ]
11        })
12     return fine_tuning_data
13

```

## Creating Training and Validation Sets

After formatting the dataset, my next step is to train and validate the model. This step is crucial for ensuring the effectiveness of the training process. Initially, I train the model on a subset of the data, allowing it to learn from examples. Following this, I validate the model's performance using various subsets of the data. This validation process helps assess how well the model has learned and how accurately it can generate responses in different contexts.

```

1 from sklearn.model_selection import train_test_split
2
3 #Stratified splitting. Assuming 'Top Category' can be used for stratification
4 train_data, val_data = train_test_split(
5     converted_data,
6     test_size=0.2,
7     stratify=dataset['Top Category'],

```

```
8     random_state=42 #for reproducibility
9 )
10
```

## Creating JSONL Files

To fine-tune GPT-3.5 with OpenAI, the data needs to be in JSONL format. Therefore, I converted both the training and validation sets into this format and saved them as separate files. This conversion ensures compatibility with the fine-tuning process and allows for efficient training and validation of the model.

```
1 def write_to_jsonl(data, file_path):
2     with open(file_path, 'w') as file:
3         for entry in data:
4             json.dump(entry, file)
5             file.write('\n')
6
7 training_file_name = "train.jsonl"
8 validation_file_name = "val.jsonl"
9
10 write_to_jsonl(train_data, training_file_name)
11 write_to_jsonl(val_data, validation_file_name)
```

## Uploading Data and Starting the Fine-Tuning Job

Now that the datasets are formatted into JSONL format, the next step is to upload them to OpenAI and begin the fine-tuning process. This involves providing the formatted datasets to the OpenAI platform, which will use them to fine-tune the GPT-3.5 model. During fine-tuning, the model will learn from the provided datasets to improve its performance and generate more accurate responses. Once the fine-tuning process is complete, the model will be better equipped to understand and respond to queries in the specified context.

```
1 from openai import OpenAI
2 client = OpenAI(api_key="your_open_ai_key")
3
4 # Upload Training and Validation Files
5 training_file = client.files.create(
6     file=open(training_file_name, "rb"), purpose="fine-tune"
7 )
8 validation_file = client.files.create(
9     file=open(validation_file_name, "rb"), purpose="fine-tune"
10 )
11
```

```
12 # Create Fine-Tuning Job
13 suffix_name = "yt_tutorial"
14 response = client.fine_tuning.jobs.create(
15     training_file=training_file.id,
16     validation_file=validation_file.id,
17     model="gpt-3.5-turbo",
18     suffix=suffix_name,
19 )
20
```

## Testing the Fine-Tuned Model

Now that the fine-tuning process is complete, the next step is to test the performance of the model. To do this, I used the prediction function with the fine-tuned model. This function allows us to input queries or messages and receive responses generated by the model.

After testing the model's performance, I stored this method for future use. This may involve saving the model's parameters and configurations in a file format that can be easily loaded and utilized whenever needed. Additionally, I consider deploying the model to a cloud platform to make it accessible to users or integrate it into applications.

```
1 from sklearn.metrics import accuracy_score, precision_score, recall_score,
2 f1_score
3
4 def format_test(row):
5     formatted_message = [{"role": "user", "content": row['Support Query']}]
6     return formatted_message
7
8 def predict(test_messages, fine_tuned_model_id):
9     response = client.chat.completions.create(
10         model=fine_tuned_model_id, messages=test_messages, temperature=0,
11         max_tokens=50
12     )
13     return response.choices[0].message.content
14
15 def store_predictions(test_df, fine_tuned_model_id):
16     test_df['Prediction'] = None
17     for index, row in test_df.iterrows():
18         test_message = format_test(row)
19         prediction_result = predict(test_message, fine_tuned_model_id)
20         test_df.at[index, 'Prediction'] = prediction_result
21
```

```
22     test_df.to_csv("predictions.csv")
23
24
```

Now that I have trained the GPT-3.5 model, my next big task is to put it into our system. This means making sure it fits well with what I already have. I need to check how well it works by testing it out in different situations. Once I am sure it is good to go, I set it up in our main system for everyone to use. After that, I keep an eye on it to make sure it keeps working smoothly. My goal is to make things easier for users.

### 6.2.10 Analytics

In Figure 6.6 on the analytics page, I offer a detailed overview of our ticketing system. I display important metrics like the total number of tickets sold, the revenue earned from ticket sales, and the available tickets remaining for purchase. To make it easier to understand, I have included visual graphs that show how tickets are distributed across different time slots. These graphs help users see the number of tickets left for each slot. Additionally, I categorize tickets as free, full-price, and half-price and provide a separate graph showing the quantity of each ticket type sold. For even more useful insights, our analytics tool lets users check data for specific timeframes and months, giving them dynamic and up-to-date information to make informed decisions.

```
1 //JSON query to get the slots.
2
3 { 'type': 'get_daily_slots_with_sold_tickets',
4   'activityId': activity_id,
5   'date': {year, month, day},
6 };
```

### Collections and Accesses

In the collection and access section, I provide detailed reports in table format, organizing all tickets by type. This table includes data such as the total revenue from online sales (transactions made without cash) and the overall sales from the box office. Users can select a specific slot ID from the calendar to access a list of participants who made online purchases. To make it easier for users, dates without online sales are initially deactivated in the calendar. Then, a function is activated to show only the slots that have been sold online, helping users obtain specific and relevant information for effective reporting and analysis.



```
1 //JSON query to get the slots of the day.
2
3 {
4   type: 'get_slots_days',
5   activity_id,
6 };
7
```

With this query, I can retrieve the names and contact details of all users who have made online purchases from our system's database. This information includes the names and associated contact details of users who have completed transactions online.

```
1 //JSON query to get the report.
2
3 {
4   type: 'get_month_report_users_by_slot',
5   slot_id: <SLOT_ID>
6 };
7
```

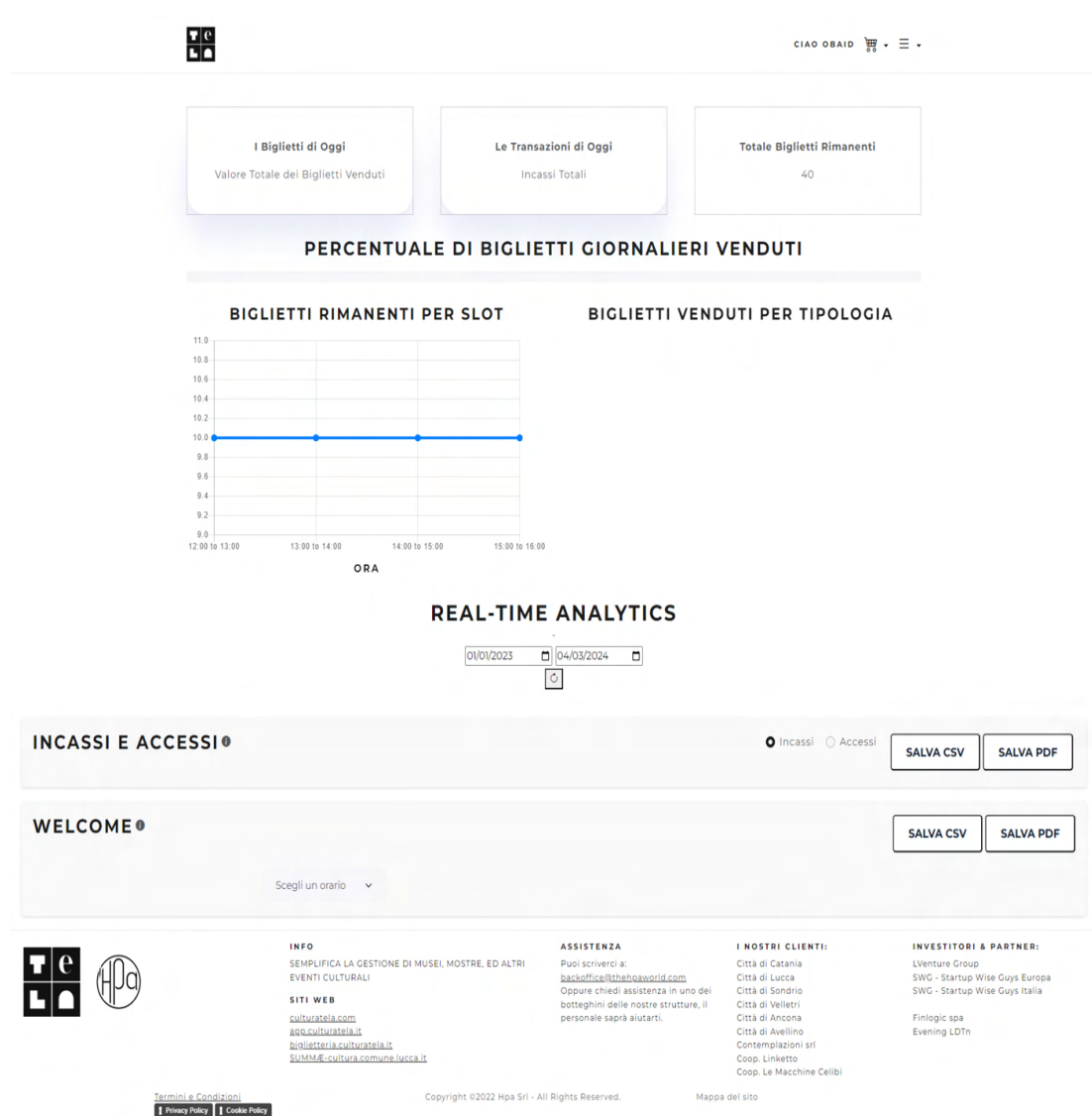


Figure 6.6: Analytics / Collections and Accesses

### 6.2.11 Scanner

I created the scanner for quick validation purposes. The main purpose of developing a scanner is to read the QR codes on tickets. The scanner checks critical information such as activity ID, ticket user ID, and ticket type. If these details match, the ticket is validated; otherwise, it remains unvalidated. This page is designed for rapid ticket validation. The initial query is employed to verify that the necessary management permissions are satisfied, ensuring a secure and controlled validation

process.

```
1 //JSON query to get the gestionale details.
2
3 {
4   'type': 'gestionale',
5   'userId': userId,
6   'activityId': activityId,
7 };
8
```

After confirming the tickets through the QR code scanner, I am inserting the data into a table displaying important details about each ticket. This table will show everything you need to know about the validated tickets, as shown in Figure 6.7. Additionally, upon receiving the results, I can create a popup containing information about all the tickets that were sold successfully. This pop-up makes it easy to view and manage data about ticket sales, providing a convenient way to track sales information.

```
1 //JSON query to get the gestionale_today_orders.
2
3 {
4   'type': 'gestionale_today_orders',
5   'userId': userId,
6   'activityId': activityId,
7 };
8
```

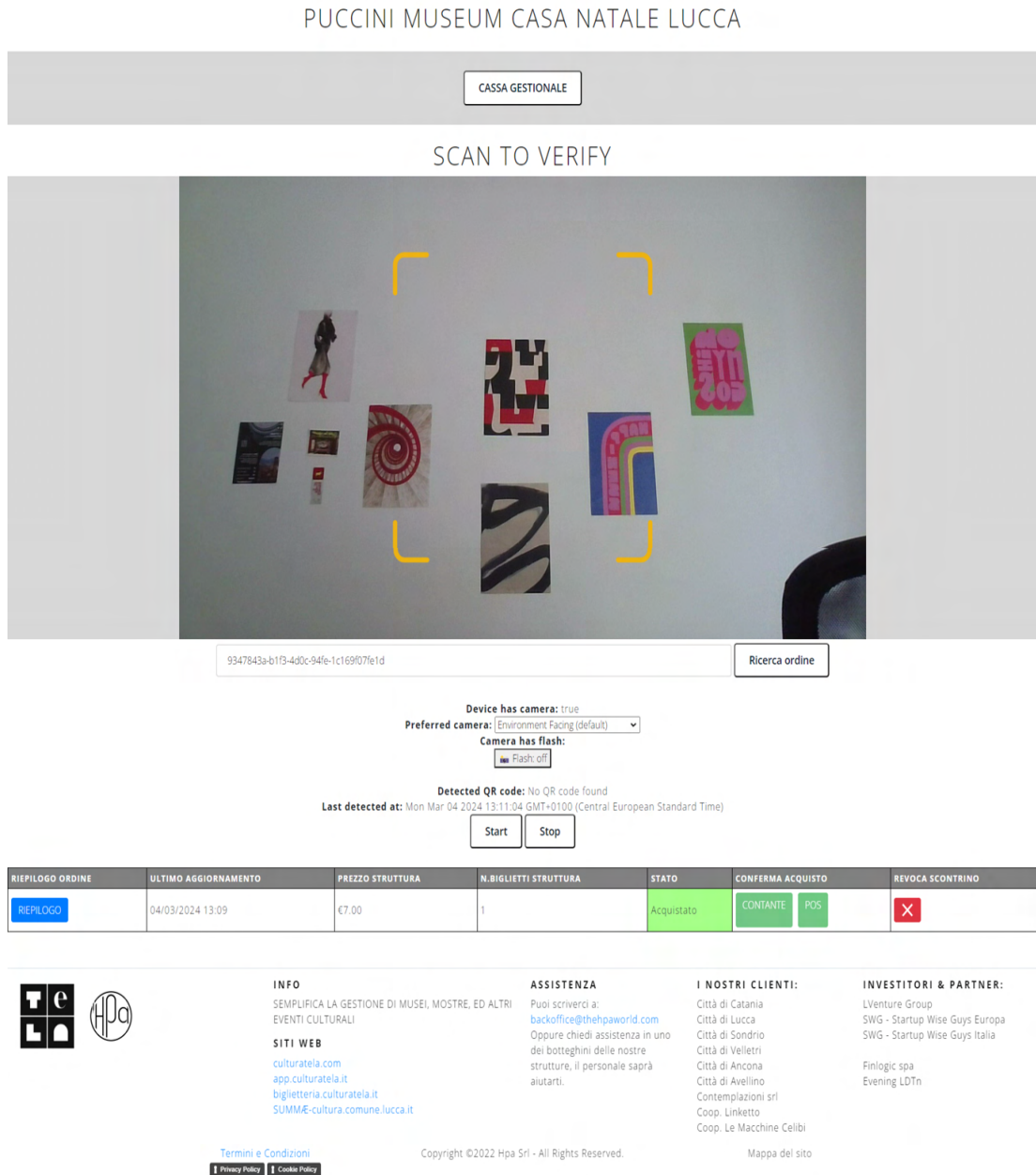


Figure 6.7: Scanner Page

## Fetching ID for scanning & showing modal

After retrieving the ID and details, I developed a model to showcase the ticket's status. This model includes information indicating whether the ticket has been

validated, the timestamp of validation, and details such as the ticket type and activity ID. By presenting this information in a structured manner, the model offers a clear and organized representation of the ticket's current status and relevant details, facilitating efficient management and tracking of ticket validation processes.

```
1 //JSON query to get the recap_order_activity.
2
3 {
4   'type': 'recap_order_activity',
5   'userId': userId,
6   'orderId': orderId,
7   'activityId': activityId,
8 };
9
```

### Validated or not Validated

The result is displayed in a pop-up window, where the only available action is to validate the ticket and proceed further, as shown in Figure 6.8. This validation option is only enabled if the order has not been confirmed previously and remains valid. By limiting the action to validation only, I ensure that the ticket is processed accurately and efficiently and prevent any unauthorized access or misuse.

```
1 //JSON query to get the validate_ticket_from_desky.
2
3 {
4   'type': 'validate_ticket_from_desk',
5   'userId': userId,
6   'activityId': activityId,
7   'orderId': orderId,
8 };
9
```

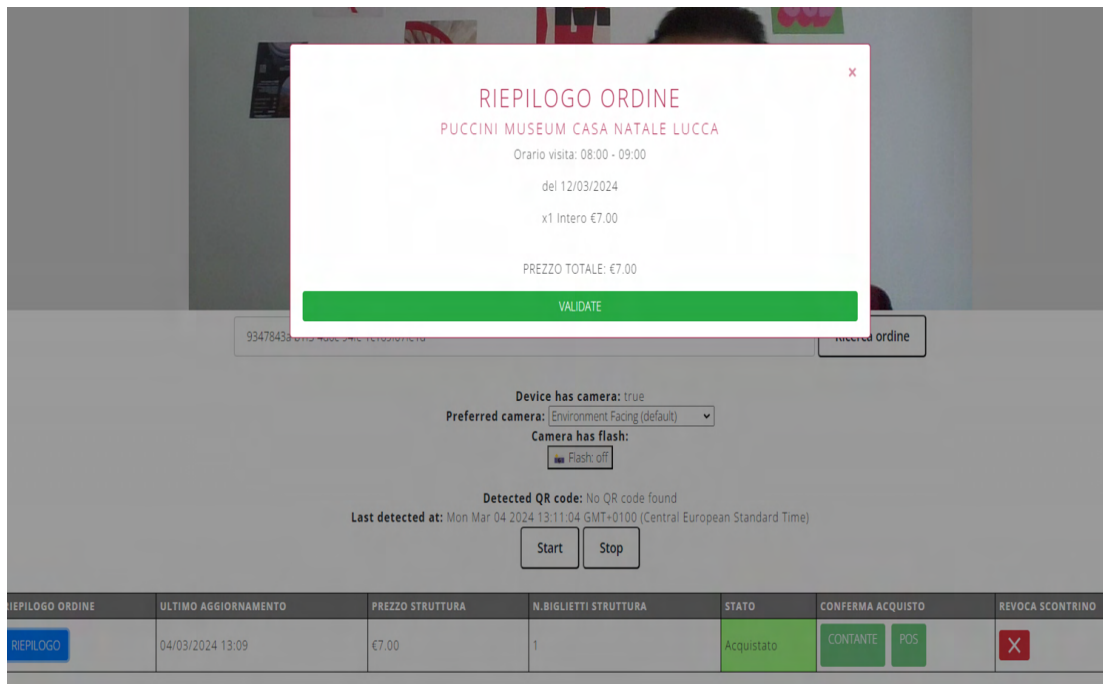


Figure 6.8: Validation

### 6.2.12 Detail

On the detail page, users can view all the details about a specific activity. This process involves specifying the activity ID in the URL to access the relevant information. The management of slots and tickets is closely linked to this activity ID, ensuring that the displayed details correspond accurately to the specified activity.

1 <https://culturatela.com/it/detail/648c206b5020bf4f83b199bd>

By including the activity ID in the URL, it becomes possible to make a call to access and retrieve the specific details of that particular activity. This ID serves as a unique identifier, enabling the system to fetch and display accurate information associated with the specified activity. This approach ensures that users can easily access and view the relevant details they need, contributing to a smoother and more efficient user experience.

```

1 // the server-side query is:
2
3 {
4   'type': 'get_activity',

```

```
5     'activityId': activityId
6 }
```


If the server response is successful, it will get the get activity details.



```
1 {
2   status: 'done',
3   data: {
4     title: 'Titolo',
5     description: 'Descrizione',
6     info: {
7       nameLocation: 'Torre Guinigi',
8       address: 'address',
9       streetNumber: 41,
10      howToReachUs: 'Come Raggiungerci'
11    },
12    type: 'museum',
13    hashtags: [ 'museum', 'torreguinigi' ],
14    userId: 'N67InOk3qgUONGjKKSveWPltTB92',
15    tickets: [
16      { type: 'normale', price: 300 },
17      { type: 'ridotto_bambini', price: 150 },
18      { type: 'ridotto_Studenti', price: 200},
19    ],
20    mainPhotoObj: { id: '5f75d37b4379130805fa6f9b', type: 'image/jpeg' },
21    galleryObjs: [
22      { id: '5f75d37b4379130805fa6f9d', type: 'image/jpeg' },
23      { id: '5f75d37b4379130805fa6f9f', type: 'image/jpeg' }
24    ],
25    activityId: '5f75d37b4379130805fa6fa1',
26    sellingDays: [
27      { day: '2020-02-11T00:00:00.000Z' },
28      { day: '2020-01-27T00:00:00.000Z' },
29      { day: '2020-03-06T00:00:00.000Z' },
30      { day: '2020-03-04T00:00:00.000Z' },
31      { day: '2020-01-18T00:00:00.000Z' },
32      { day: '2020-03-25T00:00:00.000Z' },
33      { day: '2020-03-26T00:00:00.000Z' }
34    ],
35    creator: "...",
36  },
37  message: null
38 }
39
```

The frontend triggers the execution of the following rendering method, which is implemented on the backend. This method is responsible for processing and presenting the details associated with the specified activity. Using the provided activity ID, it fetches the required data from the backend and initiates the rendering process. This ensures that the frontend receives accurate and up-to-date information about the specified activity, facilitating seamless interaction and the user experience.

In Figure 6.9, on the detail page, users have the option to select a specific activity, and there are methods available on the page for adding that particular activity to the cart. These methods streamline the process of adding the chosen activity to the shopping cart, providing a user-friendly way for users to include items in their cart for subsequent purchases. This functionality enhances the overall shopping experience by simplifying the selection and purchase process for users.






CIAG ORAID  

---

## PUCCINI MUSEUM CASA NATALE LUCCA





**Dove:**  
PUCCINI MUSEUM - CASA NATALE LUCCA, piazza cittadella 5


**Quando:**  
La programmazione dipende dai giorni, scoprirli sotto!


Il museo è situato nell'appartamento, al secondo piano di un antico edificio nel cuore di Lucca. In cui è nato Giacomo Puccini il 22 dicembre 1858.

**DA SAPERE**  
PUCCINI MUSEUM - CASA NATALE LUCCA

piazza cittadella 5 

3461385322 

Contatta 

Vai al Sito 

---

### PRENOTA LA TUA VISITA

1. Data

2. Orario

3. Tipo

4. Riepilogo

#### 1. SELEZIONA LA DATA DI VISITA

Ottobre 2024						
LU	MA	ME	GI	VE	SA	DO
30	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	01	02	03

**Avanti >**

**BIGLIETTO INTERO**  
Biglietto acquistabile da tutti coloro che hanno un'età compresa fra i 18 e i 65 anni. Vedi tra le **CONVENZIONI** se hai diritto ad una agevolazione.

**BIGLIETTO RIDOTTO**  
Biglietto per minori tra 11 e 17 anni e maggiori di 65 anni; studenti fino a 26 anni; residenti nel Comune di Lucca; disabili vedi **ACCESSIBILITÀ**. Vedi tra le **CONVENZIONI** se hai diritto ad una agevolazione.

**BIGLIETTO FAMIGLIA TEST**  
Il biglietto famiglia comprende due adulti e un minore tra 11 e 17 anni. Per ogni altro minore il biglietto è di € 3,00. Per bambini fino a 10 anni prenotare comunque il biglietto gratuito.

**ANCHE SE GRATUITO IL BIGLIETTO DEVE ESSERE PRENOTATO. LA GRATUITÀ VIENE CONCESSA A:**  
Bambini fino a 10 anni;  
Accompagnatori dei visitatori disabili;  
Giornalisti (all'ingresso verrà richiesto la tessera professionale);  
Guide turistiche (all'ingresso verrà richiesta la tessera professionale).

---



**INFO**  
SEMPLIFICA LA GESTIONE DI MUSEI, MOSTRE, ED ALTRI EVENTI CULTURALI

**SITI WEB**  
culturatela.com  
app.culturatela.it  
biglietteria.culturatela.it  
SUMME-cultura.comuna.lucca.it

**ASSISTENZA**  
Puoi scriverci a: [backoffice@thepaworld.com](mailto:backoffice@thepaworld.com)  
Oppure chiedi assistenza in uno dei botteghini delle nostre strutture, il personale saprà aiutarti.

**I NOSTRI CLIENTI:**  
Città di Catania  
Città di Lucca  
Città di Sondrio  
Città di Velletri  
Città di Ancona  
Città di Avellino  
Contemplazioni srl  
Coop. Linketto  
Coop. Le Macchine Cellibi

**INVESTITORI & PARTNER:**  
Lventure Group  
SWG - Startup Wise Guys Europa  
SWG - Startup Wise Guys Italia  
  
Finlogica spa  
Evening LDTn

Termini e Condizioni  

Copyright ©2022 Hpa Srl - All Rights Reserved.

Mapa del sito

Figure 6.9: Detail Page

### 6.2.13 DESK

On the desk page, I oversee all aspects of ticket management, ensuring a smooth experience for users. This includes organizing ticket activities by dates and available time slots, making it easy for users to find and book tickets for their desired events, as shown in Figure 6.10. I manage ticket bookings, keeping track of which tickets have been paid for and which are only reserved. This distinction helps in managing inventory and ensuring accurate availability for users. Additionally, I handle ticket types and quantities, ensuring users can easily see how many tickets they are purchasing and at what prices. Furthermore, I facilitate the delivery of purchased tickets to users via email, providing a convenient way for them to access their tickets. Lastly, I utilize a robust reporting system to generate detailed reports, offering insights into ticketing performance. These reports help in making informed decisions regarding ticket sales strategies and event planning. I make buying tickets easy and smooth for users, while also making sure I handle tickets efficiently and effectively.

```
1 //the server side query is:
2
3 {
4   'type': 'get_desk_activity',
5   'activityId': activityId
6 }
```

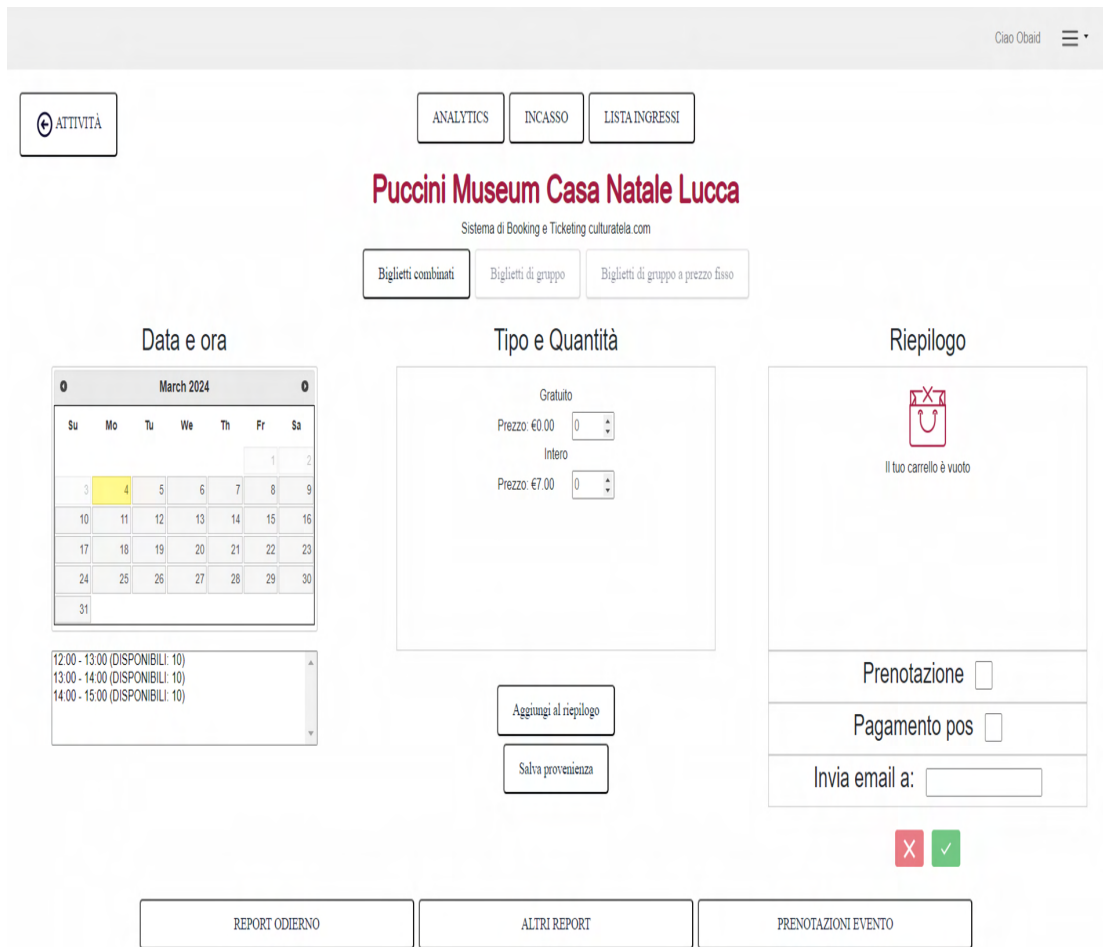


Figure 6.10: Desk Page

### 6.2.14 Flutter

I have developed a Flutter application with the purpose of providing detailed information about various festivals and events. This mobile app serves as a comprehensive guide, offering users a rich repository of festival details from different corners of Italy. Through the intuitive interface and interactive features, users can search deep into the details of each festival or event. From vibrant imagery capturing the essence of the festivities to descriptive narratives clearly indicating cultural significance, the app encapsulates the essence of diverse celebrations. Users can explore a different kind of festival, each presented with careful attention to detail, including dates, locations, traditions, and highlights. The application not only serves as a platform for discovery but also fosters cultural appreciation and

understanding. With its cross-platform functionality, the app ensures accessibility for users across iOS and Android devices, facilitating seamless exploration and enjoyment of festivals regardless of the device used.

# Chapter 7

## Testing

During the testing phase of the software development life cycle, the developed software is thoroughly examined to ensure it meets the specified requirements and functions correctly. This involves various types of testing, including unit testing to test individual components, integration testing to test how different parts work together, system testing to test the entire system as a whole, and user acceptance testing to ensure it meets user expectations. Any issues or bugs found during testing are identified, documented, and addressed. The goal of this phase is to identify and fix any defects before the software is deployed to users, ensuring a high-quality and reliable final product.

### 7.1 Unit Testing

Unit testing is a critical phase in software development I tested individual components and units of code in isolation to verify their behavior and functionality. During this process, each unit is subjected to various inputs to ensure it performs as expected under different scenarios, with a focus on identifying and fixing defects early in the development cycle. By isolating units and testing them independently, I can correct or fix bugs, receive immediate feedback on code changes, and prevent further issues. Unit testing promotes code correctness, enhances software quality, and facilitates rapid iteration and improvement throughout the development process.

After completing the unit testing I perform integration of the modules together, where software development is aimed at evaluating how various modules or components of the software interact and function together as a unified system. This process verifies that integrated units behave as expected when combined, ensuring seamless operation and compatibility across different parts of the software like front end modules with the backend. It helps to detect any inconsistencies or issues that

may arise when integrating individual components, enabling me to address them proactively and ensure the overall functionality and performance of the software system.

## 7.2 Acceptance Testing

In acceptance testing, I evaluate software to ensure it aligns with the requirements and expectations of end-users or stakeholders. This process aims to validate that the software meets specified criteria, such as functionality, performance, security, and usability, ensuring it is ready for deployment. Validation is typically done through a series of predefined tests and scenarios that simulate real-world usage, involving both automated tests and manual inspections. It includes real-world scenarios and user interactions to verify that the software functions as intended and delivers the desired outcomes. It serves as a final check to confirm that the software meets the needs of its intended users and satisfies business objectives. By conducting acceptance testing, end-users can gain confidence in the software's readiness for deployment and ensure its successful adoption.

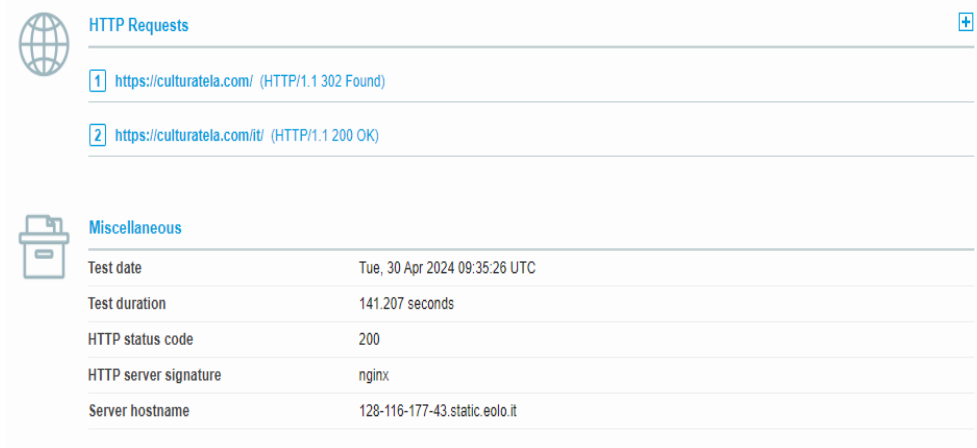
## 7.3 Regression Testing

In regression testing, I retested the previously developed components to ensure that recent changes or enhancements have not adversely affected its existing functionality. It is very crucial whenever modifications are made to the software, such as adding new features, fixing bugs, or implementing updates. The primary goal of regression testing is to identify and address any unintended side effects or regressions introduced by the changes, thereby ensuring that the software continues to perform as expected across various scenarios. By systematically rerunning test cases and verifying the behavior of the software, regression testing helps maintain the overall quality and stability of the software over time.

## 7.4 System

System testing, where the entire software system is evaluated as a unified entity. This comprehensive testing process assesses the behavior and performance of the software system as a whole, ensuring that all individual components work together seamlessly and fulfill the specified requirements. System testing validates various aspects of the software, including its functionality, reliability, performance, and usability, by subjecting it to real-world scenarios and user interactions. By thoroughly testing the integrated system, system testing helps identify any potential

defects or discrepancies, enabling me to address them before the software is deployed to end-users. Overall, system testing plays a crucial role in ensuring the quality, reliability, and functionality of the software system prior to its release.



**Figure 7.1:** System Testing

In Figure 7.1, the system test for the entire software was conducted using a comprehensive suite of automated and manual tests. The test suite included functional tests to verify that all features perform as intended, performance tests to ensure the software can handle expected loads, and security tests to identify vulnerabilities. Real-world scenarios and user interactions were simulated to assess usability and reliability. The result of this extensive testing process yielded a 200 status code. This status code indicates that the software has passed the test successfully, confirming that all components and functionalities are operating as expected without any errors or issues. The successful test result demonstrates that the software meets the specified requirements and is ready for deployment.

## 7.4.1 Performance Testing

Performance testing is aimed at evaluating how well a software application performs under different conditions. It assesses the software's responsiveness, speed, scalability, and stability when subjected to varying levels of load, stress, and concurrency. The primary objective of performance testing is to ensure that the software can handle the expected workload and user interactions without experiencing performance degradation or system failures. By simulating real-world usage scenarios and measuring key performance metrics, such as response times, throughput, and resource utilization, performance testing helps identify performance bottlenecks, scalability limitations, and areas for optimization. Ultimately, performance testing enables developers to fine-tune the software's performance, optimize its resource utilization, and deliver a reliable and responsive user experience.

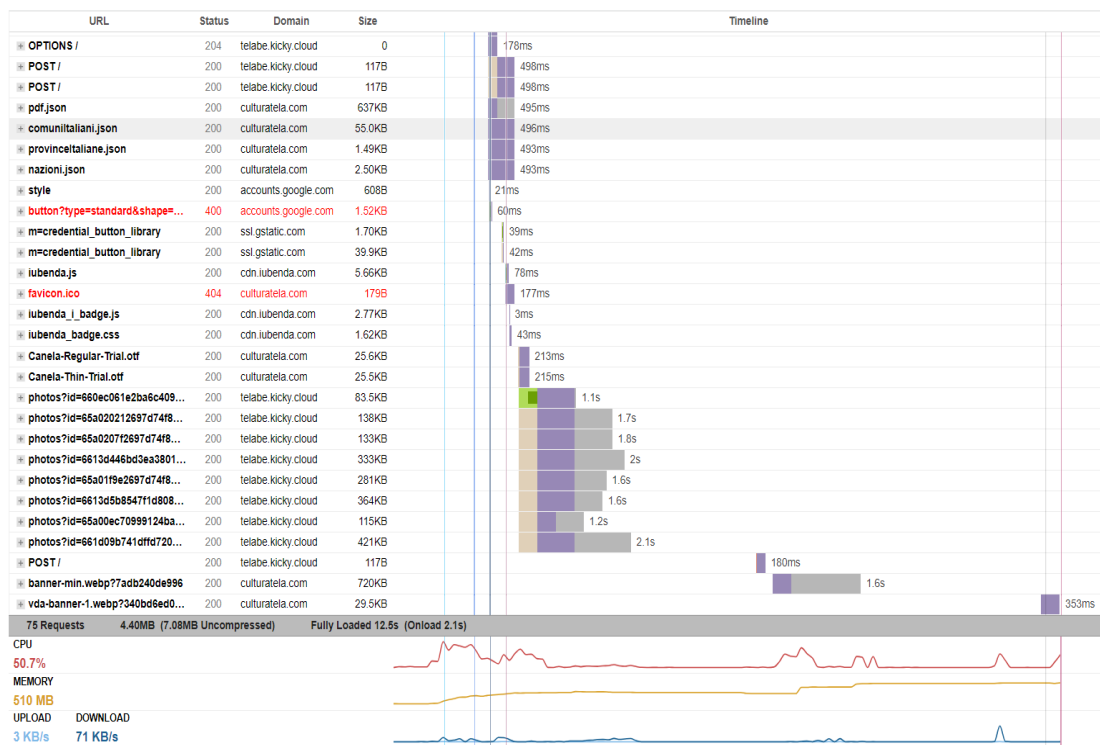


Figure 7.2: Waterfall Chart



Figure 7.2 offers a comprehensive view of the loading process of culturatela.com, presenting a detailed breakdown of every script, media file, and third-party resource requested by the webpage. Waterfall charts serve as a visual representation of loading behavior, they allow me to see the order in which resources are loaded. Each request is accompanied by overview details, including duration and execution times, depicted by varying bar lengths indicating the time taken for each request to initiate, download, and execute. This data proves invaluable for diagnosing performance issues and addressing general page concerns. By analyzing the loading behavior section by section, users can gain insights into the intricacies of the webpage's loading process and pinpoint areas for optimization. Despite the complexity of the chart, it categorizes the information into manageable sections and facilitates a structured analysis, making it easier to identify potential optimization opportunities. From the file names and status codes to the origin of the files and the breakdown of duration times, each component contributes to a comprehensive understanding of the webpage's loading performance.



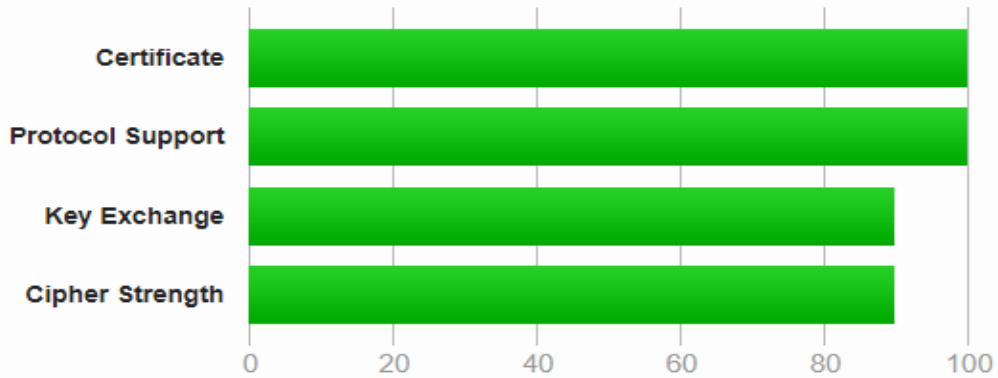
**Figure 7.3:** Page Visualisation

In Figure 7.3, the speed visualization graph depicts the loading process of the main page, highlighting various performance metrics obtained during the analysis. The total page load time is reported as 12.4 seconds, which includes the time it takes for the page to become interactive (1.8 seconds), the on-load time (2.1 seconds), the first content paint (1.0 seconds), the largest contentful paint (1.5 seconds), which is used to measure the loading performance of webpages, and the time to first byte (TTFB) of 716 milliseconds. Speed visualization captures the progression of page load at intervals, presenting key performance metrics overlaid on the timeline. This visualization offers a comprehensive understanding of the visitor's experience by providing contextual information about performance metrics relative to the page load. It helps identify areas for improvement and optimize the webpage's performance to enhance the user experience and engagement.

### 7.4.2 Security Testing

Security testing identifies vulnerabilities and weaknesses in the software's security mechanisms. It helps ensure that the software is resistant to unauthorized access, data breaches, and other security threats.

Security testing is aimed at identifying potential vulnerabilities and weaknesses in a software application's security defenses. This type of testing involves systematically evaluating the software's security mechanisms to ensure that it is robust and resilient against various security threats, including unauthorized access, data breaches, and malicious attacks. Security testing encompasses a range of techniques and methodologies, including penetration testing, vulnerability scanning, and code review, to uncover potential security flaws and gaps in the software's defenses. By identifying and addressing security vulnerabilities early in the development process, security testing helps mitigate the risk of security breaches and protects sensitive data from unauthorized access or exploitation. Ultimately, security testing plays a vital role in ensuring the overall integrity, confidentiality, and availability of the software, thereby safeguarding both the application and its users against potential security threats.



**Figure 7.4:** Security Testing

In Figure 7.4, I conducted security testing on the website using SSL-lab to predict these scores, utilizing certificates to assess various security parameters. The results indicate a robust security posture, with a 100 percent score for certificate and protocol support, signifying strong encryption protocols and secure communication channels. Although key exchange and cipher strength scored slightly lower at 90 percent, the overall website security remains high. These results affirm the implementation of effective security measures, ensuring data integrity, confidentiality, and protection against potential threats or vulnerabilities. With comprehensive security measures in place, users can trust that their sensitive information is safeguarded while interacting with the website.

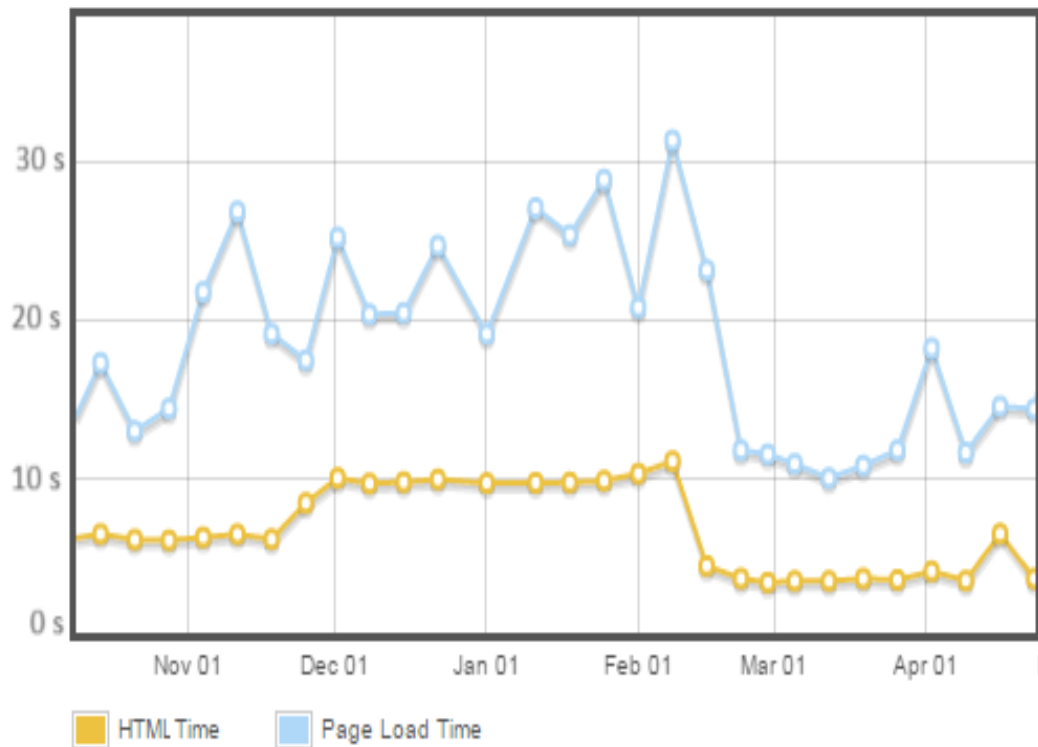
## 7.5 Usability Testing

Usability testing is an essential process in software development that focuses on assessing the user-friendliness and ease of use of the software interface. This type of testing evaluates how easily users can interact with the software and accomplish their intended tasks. By observing real users as they navigate through the software, usability testing helps identify any usability issues, such as confusing navigation, unclear instructions, or cumbersome workflows. The goal of usability testing is to ensure that the software provides a smooth and intuitive user experience, ultimately enhancing user satisfaction and productivity. Through iterative testing and feedback, I can refine the software interface to address usability issues and improve overall usability. Ultimately, usability testing plays a crucial role in creating software that meets the needs and expectations of its users, leading to higher adoption rates and user satisfaction.

I have achieved an impressive 85% performance score, which assesses our website's performance under various browser and hardware conditions, along with specified analysis options like AdBlock and connection speeds. I used gtmatrix for measuring the performance score. One crucial metric evaluated is the Largest Contentful Paint (LCP), introduced in 2020 to measure the perceived loading experience for users. Simply put, LCP measures how long it takes for the largest "content element" on the page, such as a hero image or heading text, to become visible within the visitor's viewport. I have achieved an LCP of 1.5 seconds and 3.8 seconds for the whole software, indicating that the largest content on culturatela.com loads swiftly, contributing to a positive user experience. It signifies the moment during page loading when the primary content of the page is likely to have loaded. A fast LCP is particularly valuable as it reassures users that the page is useful and relevant to their needs. By prioritizing a swift LCP, I aim to enhance the overall user experience and instill confidence in our website's utility and effectiveness.

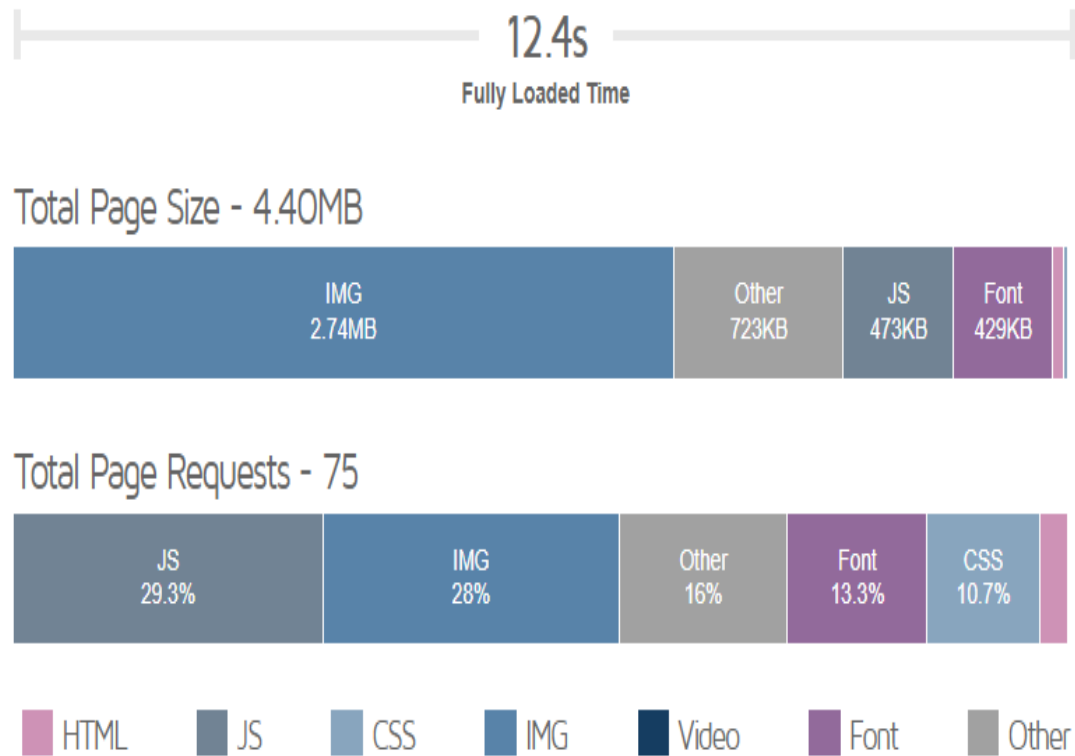
Total Blocking Time (TBT) is like counting how long it takes for your webpage to respond when someone tries to do something on it, like clicking a button or scrolling. When certain tasks on the webpage take too long to finish, it can make the webpage slow to respond to people's actions. These tasks might include things like loading images or running scripts. For example, if someone is trying to click a button on a webpage, it feels like it is taking forever to respond. That is because the tasks that the webpage is doing in the background are taking too long, blocking it from responding to you quickly. Each time this happens, it adds up to the total blocking time. Having a low total blocking time means my webpage responds quickly to people's actions, making it easier and more enjoyable to use. It is like having a smooth and fast experience every time you interact with a webpage, without any frustrating delays. TBT should be less than 50 ms and I got 2 ms which is quite good and 495ms for the whole software.

Cumulative Layout Shift (CLS) is a measure of how stable a webpage looks while it is loading. It checks if elements on the webpage move around unexpectedly as the page loads. If they do, it adds up all these movements to give a score. This score shows how often and how much the page shifts while loading. A low CLS score means the page stays visually stable as it loads. This is good for users because it makes the webpage easier to use and more enjoyable. So, when a webpage loads, some parts might move unexpectedly, which can be annoying for users. But if we have a low CLS score, like the 0.02 I achieved, it means our webpage does not shift around too much, making it nicer for people to use.



**Figure 7.5:** Load Time

In Figure 7.5, the graph illustrates the overall page load time and specifically highlights the HTML load time over the past six months. During this period, significant improvements have been made to reduce both the total page loading time and the time it takes for the browser to load the HTML content. Previously, the entire page took 32 seconds to load, whereas now it only takes 12.4 seconds. Similarly, the HTML load time in the browser has been reduced from 11 seconds to 3.5 seconds. These improvements signify a substantial enhancement in the performance and efficiency of the webpage. By decreasing the HTML load time, users experience faster access to the webpage's content, resulting in a smoother and more responsive browsing experience. Additionally, the overall reduction in page loading time contributes to improved user satisfaction and engagement with the website.



**Figure 7.6:** Page Size

Figure 7.6, illustrates the overall size of the webpage, which amounts to 4.40 megabytes (MB), and the number of requests made to load the page, totaling 75. This breakdown includes various types of resources, such as JavaScript (JS), images (IMG), fonts, CSS files, and others. For instance, JavaScript files comprise a total size of 473 kilobytes (KB), accounting for approximately 29.3% of the total number of requests. Similarly, images contribute significantly to the page size, with a total size of 2.74 MB, representing around 28% of the total requests. Additionally, other resources, such as fonts and CSS files, are also included in the analysis, providing a comprehensive overview of the composition of the webpage in terms of size and the number of requests made to fetch these resources. This breakdown helps in identifying potential areas for optimization, such as minimizing file sizes, reducing the number of requests, and optimizing resource loading strategies, to improve the overall performance and loading speed of the webpage.

After completing these different types of testing, I can confidently say that the software meets high-quality standards, functions reliably, and provides users with a positive experience. It operates smoothly and efficiently, fulfilling its intended purpose without encountering any issues. This reliability means users can depend on it to perform consistently and without interruptions. Moreover, it guarantees that users have a satisfying experience while using the software, which aligns perfectly with my goals.

# Chapter 8

## Conclusion

These days, lots of people use new technology and applications. Many people around the world buy things using their mobile phones instead of going to physical locations. It is becoming more common to purchase on a mobile or on a computer instead of going there because it's easier and faster.

The purpose of this thesis is to build a project that aims to create a frontend module for museums and events that works well with an existing platform. It will involve studying what users need, designing it to be easy to use, and making sure it works smoothly with the platform. I also add a smart chat feature to make it even better. The goal is to make managing museums and events easier and more enjoyable for users. They will keep updating and supporting it to ensure it stays compatible and improves over time.

### 8.1 Technical Development

- Compatibility with the existing SaaS platform infrastructure.
- Utilization of modern frontend development technologies such as HTML5, CSS3, and JavaScript frameworks, Django, MangoDB etc.
- Integration with backend systems for data retrieval and storage.
- Implementation of secure authentication mechanisms to ensure user data privacy.
- Optimization for performance and scalability to handle large volumes of user interactions.
- Compliance with industry standards and best practices for software development and security.



- Compatibility with different web browsers and devices to ensure a consistent user experience.
- Implementation of robust error handling and logging mechanisms for troubleshooting and maintenance purposes.

## 8.2 Functional Development

- User authentication and authorization system to control access to different features and data.
- User-friendly interface for managing museum and event-related tasks such as scheduling, ticketing, and resource allocation.
- Interactive dashboard providing real-time insights and analytics on user engagement and performance metrics.
- Integration with third-party services for additional functionalities such as payment processing and communication tools.
- Seamless navigation and an intuitive layout to facilitate easy interaction with the frontend module.
- Customizable settings and preferences allow users to tailor their experience according to their needs.
- Integration with social media platforms for sharing and promoting museum exhibitions and events.

## 8.3 User Development

- Intuitive and user-friendly interface that does not require extensive training to use.
- Access to comprehensive documentation and support resources for assistance with any issues.
- Fast and responsive performance minimizes wait times and frustration during usage.
- Customizable preferences to personalize the experience according to individual preferences.

- Secure handling of personal and financial information to ensure user privacy and data protection.
- Compatibility with different devices to accommodate users' preferred methods of access.
- Regular updates and improvements based on user feedback are needed to enhance functionality and address any usability issues.
- Reliable customer support channels for resolving queries and providing assistance when needed.

## **8.4 Future Development**

In future development, Culturatela.com plans to enhance the system by adding new functionalities. The system is designed to be scalable, allowing for the integration of online metro tickets and tickets for other platforms. They also aim to implement a new approach and technologies to further improve Culturatela.com. This approach will enable users to easily browse and access information about available museums, concerts, and events, and also allow users to visit virtual tours of museums. Users will be able to select their desired destination and reserve tickets in advance after experiencing the virtual tours. By adopting this approach, they aim to meet all user requirements effectively and provide a robust booking experience for their users.

## Chapter 9

# Bibliography

- [1] The Business Research Company. «*Museums Market Report*». Available: <https://www.thebusinessresearchcompany.com/report/museums-historical-sites-zoos-and-parks-global-market-report>. Last accessed: 2023-06-18. 2019.
- [2] Alexandre Augusto da Costa and Leonardo Rosa Maricato Santos. «*Polycam and the Power of Heritage Registration in the Palm of Your Hand: UNESCO Strategy to Safeguard Memory in Ukraine War*». In: *Soft Power and Heritage*. Springer, 2023, pp. 67–81.
- [3] Mateusz Naramski, Adam R Szromek, Krzysztof Herman, and Grzegorz Polok. «*Assessment of the activities of European cultural heritage tourism sites during the COVID-19 Pandemic*». In: *Journal of Open Innovation: Technology, Market, and Complexity* 8.1 (2022), pp. 55–59.
- [4] Maurizio Calderamo, Simona Ibba, Filippo Eros Pani, Francesco Piras, and Simone Porru. «*Advanced Museum Services*». In: *The Success of European Projects Using New Information and Communication Technologies* (2015), pp. 47–53.
- [5] Kyriaki Garyfala. «*Digital museums, galleries and online platforms in the current marketplace*». In: *Cultural Heritage Management and Sustainable Development* (2022).
- [6] Eirini Alexandrou. «*Digital Strategy in Museums: A case study of The Rijksmuseum, Amsterdam.*» In: *Museum Management and Curatorship* (2020).
- [7] Theodora Topali. «*Digitality and Museums; The Benefits, the Challenges and a Case of a Non-digital Museum*». <http://hdl.handle.net/11544/30355>. Accessed: June 22, 2024. 2024.

- [8] L. Richter. «*Museums and Digital Media. Potentials of Mobile Applications for Online Audience Engagement.: A Uses and Gratification Perspective on 'MuseumStars'*». In: *International Journal of Cultural Studies* (2022), pp. 19–27.
- [9] Lakshmanan Konar, Stuti Ahuja, RS Surya, and Pranav Konar. «*Streamlining Heritage Tourism: A Mobile App for Booking and QR Ticketing of Heritage Sites*». In: *International Conference on Soft Computing for Security Applications*. Springer. 2023, pp. 877–891.
- [10] Omnia Mohamed Heniedak Abo Zeid. «*Museum Collection Management in the Past, Nowadays and in the Future (Grand Egyptian Museum "Case Study")*». In: *International Journal of Multidisciplinary Studies in Heritage Research* 5.2 (2022), pp. 69–96.
- [11] D Beulah Pretty. «*Online Ticket Booking Using Secure QR Code*». In: *International Journal of Research in Engineering, Science and Management* 1.12 (2018).
- [12] Vincenzo Mighali, Giuseppe Del Fiore, Luigi Patrono, Luca Mainetti, Stefano Alletto, Giuseppe Serra, and Rita Cucchiara. «*Innovative IoT-aware services for a smart museum*». In: *Proceedings of the 24th international conference on world wide web*. 2015, pp. 547–550.
- [13] Shao Xiwen. «*RFID technology-based museum ticketing and visitor management systems*». In: *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering*. Vol. 3. IEEE. 2013, pp. 304–306.
- [14] Muhammad Hanif Irsyad and Abdul Barir Hakim. «*Developing Web Based Ticket Booking System for The Millenials Event*». In: *I-STATEMENT* 5.2 (2020), pp. 93–102.
- [15] Alisha Litman. «*Museums' Websites and Usage*». Senior Honors Thesis. Indiana University, School of Public and Environmental Affairs, 2012.
- [16] Aarohi Rathore, Aayush Gupta, Abhay Gour, and Ankur Nagar. «*E-Ticketing System for Indian Museums and Heritage Sites*». In: *International Research Journal of Modernization in Engineering Technology and Science* (2022), pp. 1–4.
- [17] Ainin Sulaiman, Josephine Ng, and Suhana Mohezar. «*E-ticketing as a new way of buying tickets: Malaysian perceptions*». In: *Journal of Social Sciences* 17.2 (2008), pp. 149–157.
- [18] Jacopo Riva. «*The assessment of the level of digital innovation in Italian museums*». In: *Journal of Cultural Heritage Management and Sustainable Development* (2019).

- [19] Matylda SIWEK. «*Impact of the COVID-19 Pandemic on The Development of Digital Services in The Activities of The Museums*». In: *Innovation Management and Sustainable Economic Development in the Era of Global Pandemic* (2020).
- [20] Adie Wahyudi Oktavia Gama, John Junieargo Timotius, I Gusti Ngurah Darma Paramartha, and I Wayan Aditya Suranata. «*Application Model As Online Ticketing And Funding Platform For Mobile-Based Cultural Events*». In: *Jurnal Mantik* 6.2 (2022), pp. 1340–1348.
- [21] Chih-Tung Hsiao and Ming-Hung Yao. «*System dynamic approach to visitors' long-term satisfaction with the museum: A case study of the national museum of national science*». In: *International Journal of Electronic Business Management* 10.2 (2012), pp. 113–121.
- [22] Shirley Radack. *The system development life cycle (SDLC)*. Tech. rep. National Institute of Standards and Technology, 2009.
- [23] Alok Ranjan, Abhilasha Sinha, and Ranjit Battewad. «*JavaScript for modern web development: building a web application using HTML, CSS, and JavaScript*». BPB Publications, 2020.
- [24] Jeff Forcier, Paul Bissex, and Wesley J Chun. «*Python web development with Django*». Addison-Wesley Professional, 2008.
- [25] Mithun Satheesh, Bruno Joseph D'mello, and Jason Krol. «*Web Development with MongoDB and NodeJs*». Packt Publishing Ltd, 2015.
- [26] Yasset Perez-Riverol et al. «*Ten simple rules for taking advantage of Git and GitHub*». In: *PLoS computational biology* 12 (2016).
- [27] Divya Sharma and Hiren Dand. «*Firebase as BaaS for College Android Application*». In: *International Journal of Computer Applications* 178.20 (2019), pp. 1–6.
- [28] Mithun Satheesh, Bruno Joseph D'mello, and Jason Krol. «*Web Development with MongoDB and NodeJs*». Packt Publishing Ltd, 2015.
- [29] Sven Amann, Sebastian Proksch, Sarah Nadi, and Mira Mezini. «*A study of visual studio usage in practice*». In: *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. Vol. 1. IEEE. 2016, pp. 124–134.
- [30] Dimitri Aivaliotis. «*Mastering NGINX*». Packt Publishing Ltd, 2016.