

**A.Y. 2021/2022**

**Department of Management and Production Engineering**

**Master's Degree: Management Engineering**

**Master Thesis:**

Applying the PM approach to the academic context, building thesis projects management framework. The front-end developer (UI/UX designer) perspective



**Politecnico  
di Torino**

**PROFESSOR:**

**DEMAGISTRIS PAOLO EUGENIO**

**GRADUATE STUDENT:**

**SAIDBEK SAMIGJONOV**

# Table of contents

Abstract

1. Introduction
  - 1.1 Traditional thesis taking process
  - 1.2. Disadvantages of current thesis taking process
  - 1.3. Solution for current problem
2. Used technologies
  - 2.1 Web application structure
  - 2.2 How a web application works
  - 2.3 Front end Development
  - 2.4 Web Technologies Involved in Frontend Development
  - 2.5 List of popular frameworks to make web app
    - 2.5.1 Angular JS
    - 2.5.2 Vue JS
    - 2.5.3 React JS
  - 2.6 Why React?
  - 2.7 Npm packages for React project
  - 2.7 Back-end development
  - 2.8 Back-end Development technologies
    - 2.8.1 Database and Cache:
    - 2.8.2 Server
    - 2.8.3 API (REST & SOAP)
  - 2.9 List of popular languages to make web app
    - 2.9.1 Express JS
    - 2.9.2 Python & Django
    - 2.9.3 Php & Laravel
    - 2.9.4 Ruby on Rails
  - 2.10 Why Python/Django for web development
  - 2.11 Meet Django
  - 2.12 What's an API?
  - 2.13 Swagger
  - 2.14 Deployment – Netlify
  - 2.15 Deployment - PythonAnywhere
  - 2.16 GitHub
3. How platform works
  - 3.1 Login
  - 3.2 Tasks
  - 3.3 Backend
4. Conclusion

## **Abstract**

The thesis work aims is the development of a Thesis Application process applying Project Management approach to the academic context, building thesis projects management framework. Final examination for the master's degree (thesis defense) is a longer academic paper required to complete a degree program. These often require months (or even years) of research and may be defended in front of a university committee or online (pandemic situation). Thesis work process is one of the requirements of the university or institution to complete master's or bachelor's degree. A thesis is a document detailing the author's study and results and presented in support of candidacy for an academic degree or professional qualification. It takes a long time to research and find relevant information to complete it in a great way. Thesis Application process is initial phase of starting thesis work and after approving my professor thesis work starts. In my research I found that there should be platform which will be bridge between supervisors (most of the time university professors) and students (who is thesis defending) which will make their communication in smooth way and helps to track thesis work in remote. Tracking the process of the thesis work, giving comments on title of the tasks, discussing about thesis title should be in a single platform and it will be useful for not only students but also for supervisors. They can easily check the status of the task completion and getting latest information about thesis status.

# **1. Introduction**

## 1.1 Traditional thesis taking process

Thesis is the last learning activity of master's educational path. It is worth a certain number of credits (CFU) which is illustrated in the Teaching Regulations of every degree programs.

The thesis application is compulsory and must be submitted before start writing thesis. The road to completion master's starts by making thesis application. In most universities it is required to fill out thesis application form online from personal page of the Teaching Portal – Online services section – Degree and Final examination portlet. Student can choose topic from list of Politecnico di Torino thesis topics. In front of topic will be written name of the professor who provide thesis topic. Once application is sent, supervisor (professor) may accept or reject thesis application and student will be informed by e-mail.

After getting confirmation student start making research on thesis work. But there is an issue which plays an important role in thesis work – monitoring the thesis progress. Initially student and professor will make a schema for overall thesis research. Once student has completed some part of the thesis work, he/she will send a file to the professor email. Professor will check it and response. Every time professor should check student status and work which he completed. Usually, they make a meeting on videos conference or meeting on university.

## **1.2 Difficulties of current thesis taking process**

To work efficiently and get feedbacks from professor, student will send file to the professor email asking to check if there any corrections. On the other side professor also, every time check its email address checking if student send a file or not. This is too long process and sometimes sending files are not optional way. Also, there is file storage system provided by Politecnico di Torino which student can save his/her files, but it also takes a lot of time and tracking thesis status for professor is not good way. So current problem is tracking status of the thesis work

## **1.3 Solution for current issue**

The solution for current problem is platform which can track and follow thesis work. In this thesis work I'll demonstrate how PM-LAB tools platform works, usability and advantage of using it. In this web app student can easily enter thesis title, entering list of data which should be completed and editing/deleting/updating functions. The platform which is going to work will be bridge between professor and student, also other students which can see the status for the thesis work. In the platform all tasks which will be completed to finish thesis work will be written and visible to everybody (students/reviewers/professors).

The aim of the thesis was to apply the PM approach to the academic context, building thesis projects management framework. To implement this tool in a real-life case it is focused to make a web application which will integrate with [didattico.polito.it](http://didattico.polito.it) website. Using this web application students can easily follow their statuses and contact with professors. How does mechanism work and what benefits may bring this new web development application?

- Student who is going to take thesis for a final work will choose thesis title and wait until professor will confirm the request. Once professor confirms, status will be changed to approved. Students can easily move to the webapp and build task management.
- To get access to the platform, he is required to use Politecnico di Torino credentials which must be available in the website database. Student can easily write list of tasks which should be completed to finish thesis work and can contact a professor if there any questions or issues occurred
- There are 3 roles: supervisor, student, and reviewers. All of them have different access rights:
  - Student can complete several options like create task, update, delete and read task.
  - Supervisors also have these options which are mentioned above but they cannot delete task.
  - Reviewers can only read and in the future may left comments, feedbacks to title

This approach helps to track all processes of thesis defense and complete thesis work in a great, comfortable way.

# **2. Used technologies**

## 2.1 Web application structure

A web application (or web app) is application software that runs on a web server, unlike computer-based software programs that are run locally on the operating system (OS) of the device. Web applications are accessed by the user through a web browser with an active network connection. These applications are programmed using a client–server modeled structure—the user ("client") is provided services through an off-site server that is hosted by a third-party. Examples of commonly used web applications include webmail, online retail sales, online banking, and online auctions.

Millions of businesses use the Internet as a cost-effective communications channel. It lets them exchange information with their target market and make fast, secure transactions. However, effective engagement is only possible when the business can capture and store all the necessary data and have a means of processing this information and presenting the results to the user.

Web applications use a combination of server-side scripts (PHP and ASP) to handle the storage and retrieval of the information, and client-side scripts (JavaScript and HTML) to present information to users. This allows users to interact with the company using online forms, content management systems, shopping carts and more. In addition, the applications allow employees to create documents, share information, collaborate on projects, and work on common documents regardless of location or device.

## 2.2 How a web application works

Web applications are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the program executable. Some of the applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server.

The web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information. Application server technology ranges from ASP.NET, ASP and ColdFusion, to PHP and JSP.

Here's what a typical web application flow looks like:

1. User triggers a request to the web server over the Internet, either through a web browser or the application's user interface
2. Web server forwards this request to the appropriate web application server
3. Web application server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data
4. Web application server sends results to the web server with the requested information or processed data
5. Web server responds back to the client with the requested information that then appears on the user's display

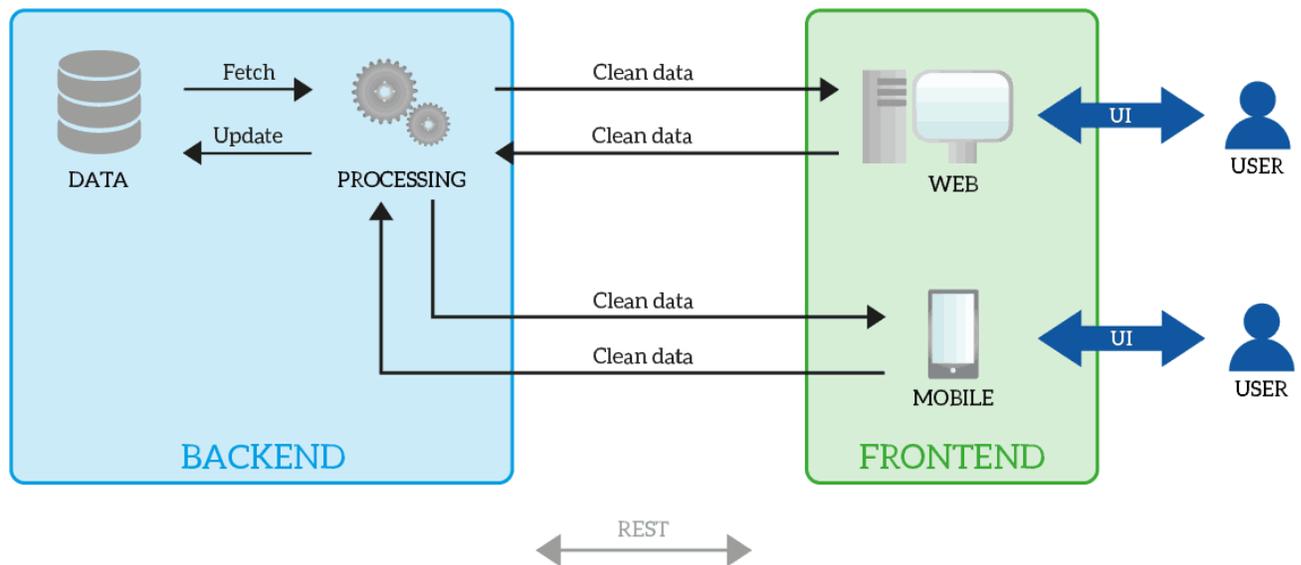


Figure 1: Integration of backend and frontend stacks

## 2.3 Front end Development

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

## 2.4 Web Technologies Involved in Frontend Development

Frontend developers use several web technologies to transform coded data into user-friendly interfaces. Among these are Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. Below are brief descriptions of the three technologies that frontend developers must be familiar with.

### 1. HTML

HTML is the building block of websites. It is the programming language used to describe and mark content, so a browser displays it correctly. For example, an image on a blog post would appear as an `<img>` in HTML code, so browsers will know that they need to display an image.

### 2. CSS

CSS looks more like a set of instructions that control a webpage's style and structure than a programming language. It helps developers manage a website or web application's formatting, presentation, and layout. While HTML defines elements on a page, CSS dictates how users see the content. For instance, it controls the size, border, and alignment of an image in a blog post.

### 3. JavaScript

Frontend developers can already create websites using HTML and CSS. In fact, it wasn't until 1995 that JavaScript emerged. However, it is now difficult to imagine websites without JavaScript as it enables developers to make sites interactive. The programming language can change website content based on a user's action. Tec slangs, for example, was created using JavaScript. Selecting an answer and clicking "Vote" would display the total number of votes for each option.

## 2.5 List of popular frameworks to make web app

To reduce busywork, web developers often use frameworks. Frameworks are skeletons of prewritten code into which developers enter domain-specific coding. Front-end frameworks eliminate overhead in making a website's client-side and are often powered by JavaScript or Java. Some popular front-end frameworks include:

- AngularJS
- React.js
- Vue.JS



### 2.5.1 Angular JS

Angular, developed by Google, was first released in 2010, making it the oldest of the lot. It is a TypeScript-based JavaScript framework. A substantial shift occurred in 2016 on the release of Angular 2 (and the dropping of the “JS” from the original name – AngularJS). Angular 2+ is known as just *Angular*. Although AngularJS (version 1) still gets updates, we will focus the discussion on Angular. The latest stable version is Angular 11, which was released in November 2020.

**Angular** is the most mature of the frameworks, has good backing in terms of contributors and is a complete package.

However, the learning curve is steep and concepts of development in Angular may put off new developers.

Angular is a good choice for companies with large teams and developers who already use TypeScript.

## 2.5.2 VUE.js



Vue, also known as Vue.js, is the youngest member of the group. It was developed by ex-Google employee Evan You in 2014. Over the last three years, Vue has seen a substantial shift in popularity, even though it doesn't have the backing of a large company. The current stable version is 3.0, released in September 2020 (with some small incremental releases since then). Contributors for Vue are supported by Patreon. It should be noted that Vue 3 is currently in its own GitHub repo, and has moved to TypeScript.

However, it has done really well in the last few years to come out as a strong competitor for Angular and React, and especially so with the release of Vue 3.0. This is perhaps playing a role with a lot of Chinese giants like Alibaba and Baidu picking Vue as their primary front-end JavaScript framework. Vue should be your choice if you prefer simplicity, but also like flexibility.

## 2.5.3 React JS



React, developed by Facebook, was initially released in 2013. Facebook uses React extensively in their products (Facebook, Instagram, and WhatsApp). The current stable version is 17.X, released in October 2020 (with smaller incremental updates since then).

React is just old enough to be mature and has a huge number of contributions from the community. It has gained widespread acceptance. The job market for React is really good, and the future for this framework looks bright.

React looks like a good choice for someone getting started with front-end JavaScript frameworks, startups and developers who like some flexibility. The ability to integrate with other frameworks seamlessly gives it a great advantage for those who would like some flexibility in their code.

## 2.6 Why React?



There are several reasons why framework React was chosen among other frameworks

- **Intuitive**

ReactJS is extremely intuitive to work with and provides interactivity to the layout of any UI. Plus, it enables fast and quality assured application development that in turn saves time for both - clients and developers.

- **Declarative**

ReactJS enables significant data changes that result in automatic alteration in the selected parts of user interfaces. Owing to this progressive functionality, there is no additional function that you need to perform to update your user interface.

- **Provides Reusable Components**

ReactJS provides reusable components that developers have the authority to reuse and create a new application . Reusability is exactly like a remedy for developers.

This platform gives the developers the authority to reuse the components build for some other application having the same functionality. Thereby, reducing the development effort and ensuring a flawless performance.

- **JavaScript library**

A strong blend of JavaScript and HTML syntax is always used, which automatically simplifies the entire process of writing code for the planned project. The JS library consists several functions including one that converts the HTML components into required functions and transforms the entire project so that it is easy to understand.

- **Components Support**

ReactJS is a perfect combination of JavaScript and HTML tags. The usage of the HTML tags and JS codes, make it easy to deal with a vast set of data containing the document object model. During this time, ReactJS works as a mediator which represents the DOM and assists to decide which component needs changes to get the exact results.

- **SEO-friendly**

React JS was introduced after immense research and improvements by Facebook. Naturally, it stands out from the crowd and allows developers to build amazing, SEO-friendly user interfaces across browsers and engines.

- **Proficient Data Binding**

ReactJS trails one-way data binding. This means that absolutely anyone can track all the changes made to any particular segment of the data. This is a symbol of its simplicity.

## **AntDesign**

Every react project requires the UI elements such as Input, Buttons and so on. You don't really have to waste time inc creating those custom UI components instead you can take help from these third-party libraries.

Personally, I am a big fan of Ant Design UI but I would highly recommend checking other libraries also because every npm package has its own use case.

## **Redux**

There is a big debate going on to the story about Redux is in the position to be replaced. But I accept that to painting the states within pages and components we are heavily dependent on Redux for sure.

## **React Icons**

Every website does not and then has fewer or more icons. Icons just make the website so live and appealing. But then back in vanilla javascript, we have to download SVG or icons and add the image manually but in react we have an entire npm module for it.

## **AXIOS**

In ReactJS, **Axios** is a library that serves to create HTTP requests that are present externally. It is evident from the fact that we may sometimes in React applications need to get data from the external source. It is quite difficult to fetch such data so that they can be normally shown on the website. Thus, it helps in retrieving the data thereby adding it to the state to facilitate the application whenever the requirement arises.

Additionally, react Axios is very easy to modify and is quite lightweight. It also works great with many other frameworks present today. The main purpose of using Axios is to get support for request and response interception, conversion of data

into JSON format, and transform it. It also helps you in protecting XSRF forgery by default while you request cross-site access.

Axios is promise-based, which gives you the ability to take advantage of JavaScript's `async` and `await` for more readable asynchronous code.

## 2.7 Back-end development

**Backend Development** is also known as **server-side development**. It is everything that the users don't see and contains behind-the-scenes activities that occur when performing any action on a website. It focuses primarily on databases, backend logic, APIs, and Servers.

The backend of a website is a combination of servers, applications, and databases. Code written by backend developers helps browsers in communicating with the databases and store data into the database, read data from the database, update the data and delete the data or information from the database.

When you click the submit button after filling in your details, they are sent to the server that then processes your data and stores it in the database. Now we have the details shared by you in our database. But did you see any of this happening? No, because it happened behind the scenes. Thus, it is known as the backend. It is just an overview of what the backend is, but this is not it, there is a lot to it.

Most modern websites are dynamic, meaning webpage content is generated on-the-fly. A dynamic page contains one or more scripts that run on the web server each time the page is accessed. These scripts generate the content of the page, which is sent to the user's web browser. Everything that happens before the page is displayed in a web browser is part of the backend.

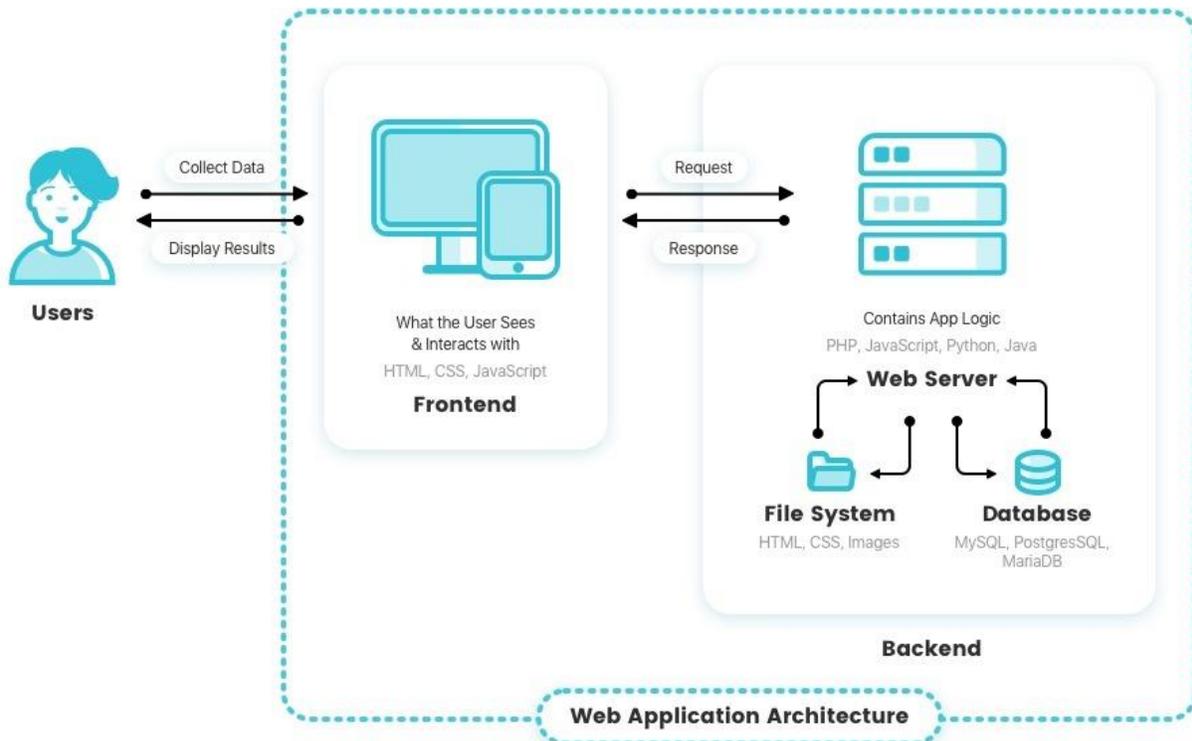


Figure 2: Web application architecture

## 2.8 Back-end Development technologies

There are several mechanism which is important in building back end development process.

### 2.8.1 Database and Cache:

Knowledge of various DBMS technology is one of the important Backend developer skills. MySQL, MongoDB, Oracle, SQL Server, Redis are widely used for this purpose. Knowledge of caching mechanisms like varnish, Memcached, Redis is a plus.

### 2.8.1 Server:

Exposure to handling Apache, Nginx, IIS servers, Microsoft IIS is desirable. A good background in Linux helps tremendously in administering servers.

### 2.8.3 API (REST & SOAP):

Knowledge of web services or API is also important for full stack developers. Knowledge of creations and consumption of REST and SOAP services is desirable.

## 2.9 List of popular languages to make web app

### 2.9.1 Express JS



Node.js is appreciated by the community of the developers. It can be run outside the browser and well-known as one of the most popular technologies. Due to the high level of performance supplemented by consistency, Node.js has become an efficient instrument for building different kinds of apps, including web and video streaming. Average full-stack developers can use Node.js for the front-end and JavaScript for the back-end.

Express is the highly demanded backend web development framework for Node.js. It helps to build apps easily and quickly. Developers use it to create Restful API's that accept requests and send suitable responses.

Let's find out why Express is highly valued among beginners and experienced developers. There are various reasons for this, namely:

Despite its minimalism, Express supports plugins that help to make web apps more functional. Processes of configuring and customizing are quite simple.

Any route of the application can be defined through the use of URLs and HTTP methods.

The great support of the framework. The process of the back-end development is realized with the use of NPM packages.

The minimum amount of time spent on debugging. This is connected to the convenient debugging mechanism.

The framework is compatible with others. For example, Kraken, LoopBack, or Sails. It's also possible to connect it with databases like MySQL, Redis, and MongoDB.

The flexibility. Due to this peculiarity, Express has been appreciated by famous companies like Stack, Twitter, and Accenture. The backend part of other products is also wholly or partly built on Express. We are referring to PayPal, IBM, GoDaddy, Flickr, Storylens, and Walmart.

## 2.9.2 Python & Django

# django

The vast set of frameworks and libraries, simplicity, and versatility turned Python into one of the most popular programming languages. Django is well-known as a high-level framework for Python that brings together various libs. Developers can use it to create an efficient app because the framework is suitable for almost any project.

The advantages of Django are:

**Fast development.** Due to Django, developers can move from concept to 100% project completion very quickly.

**High level of security.** Django helps developers to avoid standard mistakes related to security. Also, this framework automatically protects the website. Let's try to understand how it does it. First of all, Django manages an account and password without putting sensible information in cookies where it can be stolen. As usual, in such cases cookies save keys, and the important data is stored in the database.

## 2.9.3 Php & Laravel



PHP is a backend scripting programming language that can be used both for creating websites and apps.

Web PHP framework Laravel, as usual, serves for the development of apps. Due to its flexibility, features, and convenient tools, Laravel will become one of the most efficient and popular frameworks in 2021. It's both fast and simple to use Laravel.

The creator of Laravel is Taylor Otwell. The syntax of the framework is considered to be elegant and expressive.

Among the main possibilities and features of Laravel you will find various items:

Authentication and authorization can be implemented with the help of a framework.

It will be done simply helping to prevent access for the unauthorized user.

There is a strong possibility to integrate a web app with mail services. Laravel is directly connected to SwiftMailer library and has drivers for various services – from Amazon SES to Mailgun. Such drivers allow an app to send mails. Also, with the help of Laravel

it's possible to provide sending notifications through other channels. For example, Slack.

Increasing the performance of web applications. This was made possible through integration with a caching backend that is supported by Laravel.

The decrease in the technical vulnerabilities. Laravel helps to build a web app that is protected against cross+site scripting or SQL injections.

Clear error handling of the app is already configured for Laravel-based projects.

The availability of the automation testing for the app.

## 2.9.4 Ruby on Rails

Programming language Ruby is still in high demand among developers who create web applications and realize another set of various tasks. It's slower than Python but also well-known as an interpreted language. Ruby is one of three of the most famous languages for the backend. Due to advantages, lots of companies, including Airbnb, Fiverr, Github, use Ruby for their projects.



Ruby on Rails is a server-side web app framework that was written with the help of Ruby. It was presented in 2003 and written by David Hansson. The main aim of the framework is to simplify the process of website and application creation. Developers are fond of it because Ruby on Rails reduces time spent on standard tasks like a building of form, menu, or table. They just choose the out-of-the-box solution that is reusable here. Therefore, developers do not waste time on building a website or app from scratch.

Speaking about the reasons why Ruby on Rails is in high demand among companies from different industries, we should mention that this framework has obvious advantages, namely:

It's an open-source backend framework. It is allowed to reuse code in different projects.

A great community of talented developers who use Ruby on Rails and make it better. The code that is written with the usage of the framework is clear and consistent. This peculiarity is appreciated by companies that are going to improve the project that was written by the external team in the future.

Ruby on Rails is well-known as one of the cost and time-efficient frameworks.

Model-View-Controller-Format. It means that business logic is separated from other parts of an app. Each subsystem is responsible for concrete actions that help to speed up the process of development.

Scalability. You can scale an app that is based on Ruby on Rails to process more requests per one second. This is valuable for apps that are growing and should process thousands of requests.

DRY. With the help of RoR, developers do not repeat the same code writing it.

Self-describing. The code that is written can be readable without any additional documentation. Newcomers are ready to participate in any existing project.

Of course, like any other framework or programming language, Ruby on Rails has disadvantages. For example, not all web hosting providers support RoR apps. Anyway, it's still possible to find a reliable hosting provider. It means that this peculiarity won't be an obstacle to prospering of the framework in 2021-2022.

## **2.10 Why Python/Django for web development**

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm RedMonk found that it was the most popular programming language among developers in 2020

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it's relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

Python, including:

- **Its ease of use.** For those who are new to coding and programming, Python can be an excellent first step. It's relatively easy to learn, making it a great way to start building your programming knowledge.
- **Its simple syntax.** Python is relatively easy to read and understand, as its syntax is more like English. Its straightforward layout means that you can work out what each line of code is doing.
- **Its thriving community.** As it's an open-source language, anyone can use Python to code. What's more, there is a community that supports and develops the ecosystem, adding their own contributions and libraries.
- **Its versatility.** As we'll explore in more detail, there are many uses for Python. Whether you're interested in data visualisation, artificial intelligence or web development, you can find a use for the language.

## 2.11 Meet Django



**Why Is Django Right for You?**

- KISS and DRF compliant
- Batteries-included
- Intelligent Templating System
- Powerful Object Relational Mapper (ORM)
- Automatic Admin Interface
- The REST Framework
- Safety and Security
- Testing and Debugging Support

Monocubed

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

### 1. Django is simple

Django's documentation is exemplary. It was initially launched with high-quality docs, and they are still maintained at the same level, which makes it easy to use.

More than that, one of Django's main purposes is to simplify the development process: it covers the basics, so you can focus on the more unique and/or complex features of your project.

### 2. Django is time-effective

Is Django development good for MVPs and prototypes? Yes, thanks to multiple features that make it time- and cost-effective.

Let's sum them up:

- There's a flexible, well-structured admin panel, better than Laravel or Yii's, for example.
- It allows you to reuse code from current or other projects (there is also a library of reusable apps, tools, and features).
- It has great templates and forms; they were even copied by other projects.
- It has many out-of-the-box libraries and tools that allow you to assemble a good prototype in record time.

### **3. Django suits any kind of project**

Django is not an enterprise solution like C# or Java, yet it suits most types of projects, no matter their size. For example, if you're building a social media type web application, Django can handle the growth at any scale and capacity, be it heavy traffic or volumes of information. But if you want to make something simple, using Django for web development of a blog or a book database, for instance, is an excellent choice as well since it has everything you need to quickly assemble a working application.

In addition to that, Django is:

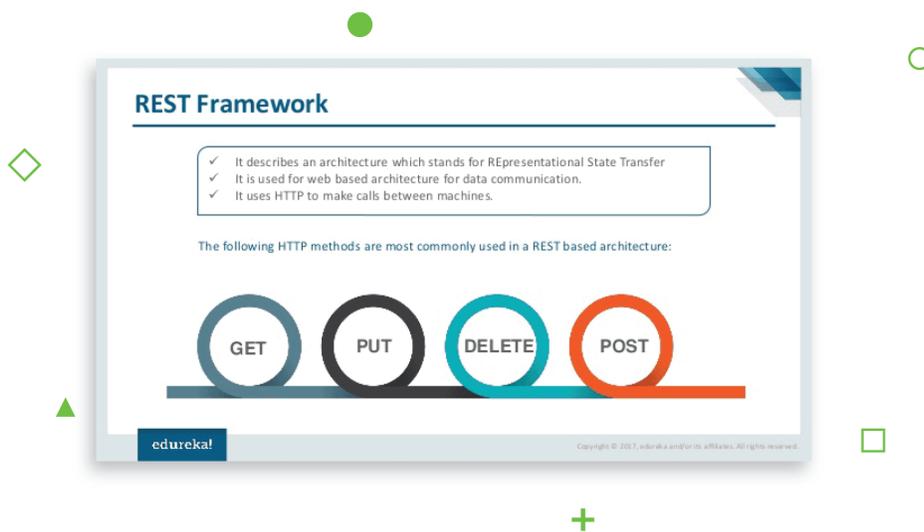
- Cross-platform. You can create applications that will run on Windows, as well as on Mac or Linux.

- Compatible with most major databases. You can use one or several different databases in one project thanks to Django's ORM, and switch between the databases with only one line of code.

#### 4. Django has a REST framework for building APIs

The benefits of using Django for web development also include its Representational State Transfer (REST) framework — a popular toolkit for building web APIs. Django's REST is powerful enough to build a ready-to-use API in just three lines of code.

One of its key advantages is that it's extremely flexible: data is not tied to any methods or resources, so REST can return different data formats and handle multiple types of calls. As a result, it can meet the requirements of different customers.



#### 5. Django is time-tested

The Django framework has been around for more than a decade, and during that time, it has become the choice of many companies for creating their web applications.

A few of the famous examples are:

- Instagram;
- Spotify;
- NASA;
- Disqus.

## 6. Django has a big, supportive, and professional community

Advantages of Django also include its big, professional community. It's quite easy to find good developers who know Django inside out and have experience coding with it.

That's a good testament to the framework's popularity – but it also means that:

- You can find help or, at least, the right direction in solving harder programming cases;
- The Django community is quick to respond to bugs and fix them;
- As an open source framework, Django is constantly improving – by means of new libraries, for example.

**Ridiculously fast** - Django was designed to help developers take applications from concept to completion as quickly as possible.

**Fully loaded** - Django includes dozens of extras you can use to handle common web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box.

**Reassuringly secure** - Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request

forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.

**Exceedingly scalable** - Some of the busiest sites on the planet use Django's ability to scale quickly and flexibly to meet the heaviest traffic demands.

## 2.12 What's an API?

An API is a set of definitions and protocols for building and integrating application software. It's sometimes referred to as a contract between an information provider and an information user—establishing the content required from the consumer (the call) and the content required by the producer (the response). For example, the API design for a weather service could specify that the user supply a zip code and that the producer reply with a 2-part answer, the first being the high temperature, and the second being the low.

In other words, if you want to interact with a computer or system to retrieve information or perform a function, an API helps you communicate what you want to that system so it can understand and fulfill the request.

You can think of an API as a mediator between the users or clients and the resources or web services they want to get. It's also a way for an organization to share resources and information while maintaining security, control, and authentication—determining who gets access to what.

Another advantage of an API is that you don't have to know the specifics of caching—how your resource is retrieved or where it comes from.

REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways.

When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint. This information, or representation, is delivered in one of several formats via HTTP: JSON (Javascript Object Notation), HTML, XML, Python, PHP, or plain text. JSON is the most generally popular file format to use because, despite its name, it's language-agnostic, as well as readable by both humans and machines.

Something else to keep in mind: Headers and parameters are also important in the HTTP methods of a RESTful API HTTP request, as they contain important identifier information as to the request's metadata, authorization, uniform resource identifier (URI), caching, cookies, and more. There are request headers and response headers, each with their own HTTP connection information and status codes. In order for an API to be considered RESTful, it has to conform to these criteria:

- A client-server architecture made up of clients, servers, and resources, with requests managed through HTTP.
- Stateless client-server communication, meaning no client information is stored between get requests and each request is separate and unconnected.
- Cacheable data that streamlines client-server interactions.
- A uniform interface between components so that information is transferred in a standard form. This requires that:
  - resources requested are identifiable and separate from the representations sent to the client.
  - resources can be manipulated by the client via the representation they receive because the representation contains enough information to do so.
  - self-descriptive messages returned to the client have enough information to describe how the client should process it.

- hypertext/hypermedia is available, meaning that after accessing a resource the client should be able to use hyperlinks to find all other currently available actions they can take.
- A layered system that organizes each type of server (those responsible for security, load-balancing, etc.) involved the retrieval of requested information into hierarchies, invisible to the client.
- Code-on-demand (optional): the ability to send executable code from the server to the client when requested, extending client functionality.

Though the REST API has these criteria to conform to, it is still considered easier to use than a prescribed protocol like SOAP (Simple Object Access Protocol), which has specific requirements like XML messaging, and built-in security and transaction compliance that make it slower and heavier.

Before we get to the code, it's worth considering why you would want to build an API. If someone had explained these basic concepts to me before I started, I would have been so much better off.

A REST API is a standardized way to provide data to other applications. Those applications can then use the data however they want. Sometimes, APIs also offer a way for other applications to make changes to the data.

There are a few key options for a REST API request:

**GET** — The most common option, returns some data from the API based on the endpoint you visit and any parameters you provide

**POST** — Creates a new record that gets appended to the database

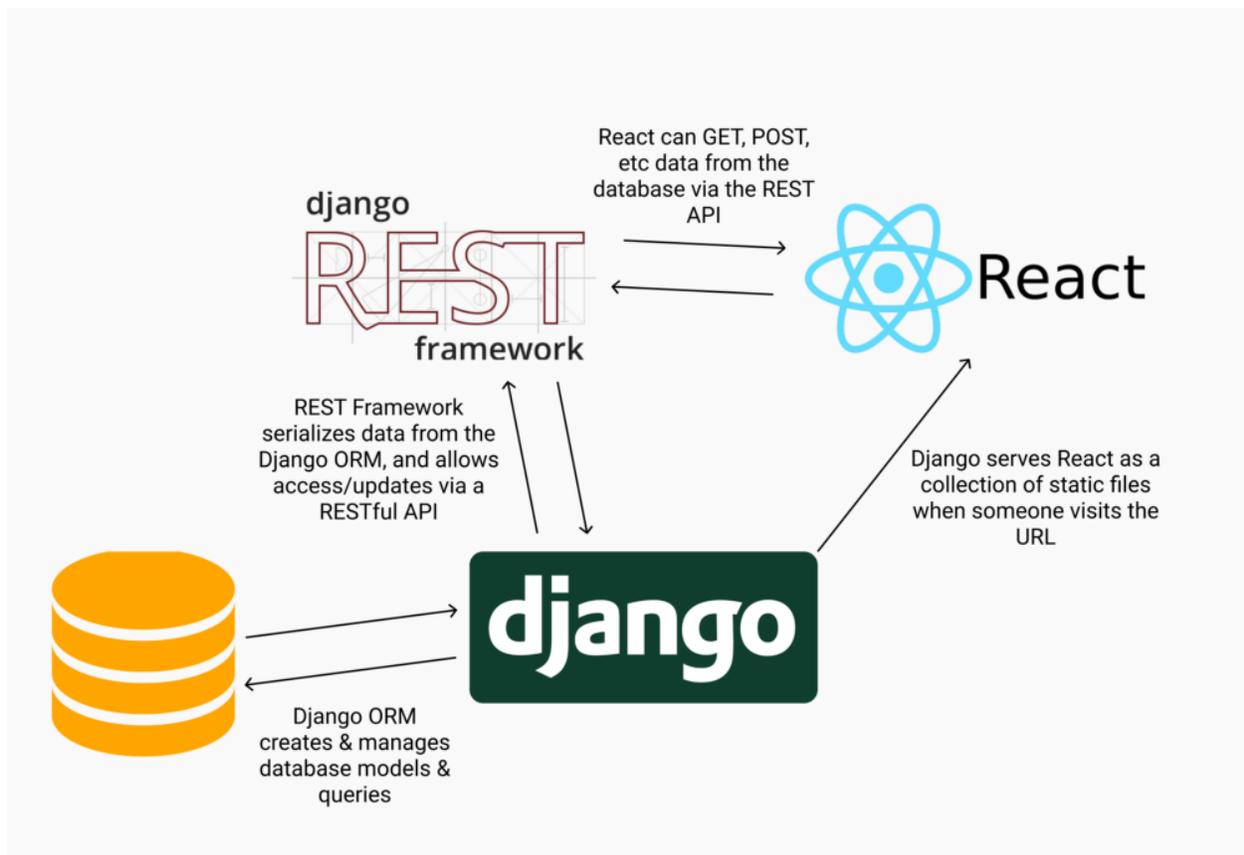
**PUT** — Looks for a record at the given URI you provide. If it exists, update the existing record. If not, create a new record

**DELETE** — Deletes the record at the given URI

## PATCH — Update individual fields of a record

Typically, an API is a window into a database. The API backend handles querying the database and formatting the response. What you receive is a static response, usually in JSON format, of whatever resource you requested.

REST APIs are so commonplace in software development, it's an essential skill for a developer to know how they work. APIs are how applications communicate with one another or even within themselves.



For example, in web development, many applications rely on REST APIs to allow the front end to talk to the back end. If you're deploying a React application atop Django, for instance, you'll need an API to allow React to consume information from the database.

The process of querying and converting tabular database values into JSON or another format is called **serialization**. When you're creating an API, correct serialization of data is the major challenge.

## 2.13 Swagger



# Swagger™

What Is OpenAPI?

**OpenAPI Specification** (formerly Swagger Specification) is an API description format for REST APIs. An OpenAPI file allows you to describe your entire API, including:

- Available endpoints (`/users`) and operations on each endpoint (`GET /users`, `POST /users`)
- Operation parameters Input and output for each operation
- Authentication methods
- Contact information, license, terms of use and other information.

API specifications can be written in YAML or JSON. The format is easy to learn and readable to both humans and machines. The complete OpenAPI Specification can be found on GitHub: [OpenAPI 3.0 Specification](#)

**What Is Swagger?**

**Swagger** is a set of open-source tools built around the OpenAPI Specification that can help you design, build, document and consume REST APIs. The major Swagger tools include:

- Swagger Editor – browser-based editor where you can write OpenAPI specs.
- Swagger UI – renders OpenAPI specs as interactive API documentation.
- Swagger Codegen – generates server stubs and client libraries from an OpenAPI spec.

### Why Use OpenAPI?

The ability of APIs to describe their own structure is the root of all awesomeness in OpenAPI. Once written, an OpenAPI specification and Swagger tools can drive your API development further in various ways:

- Design-first users: use Swagger Codegen to **generate a server stub** for your API. The only thing left is to implement the server logic – and your API is ready to go live!
- Use Swagger Codegen to **generate client libraries** for your API in over 40 languages.
- Use Swagger UI to generate **interactive API documentation** that lets your users try out the API calls directly in the browser.
- Use the spec to connect API-related tools to your API. For example, import the spec to SoapUI to create automated tests for your API.
- And more! Check out the open-source and commercial tools that integrate with Swagger.

## 2.14 Deployment – Netlify



Netlify is one of the most amazing web development platform which is meant to multiply your productivity in the best possible way. The platform helps developers to build, test, and deploy websites. By unifying the modern decoupled web elements from local development processes to advanced logics, Netlify is offering an amazingly faster way to ensure much more performant, scalable, and secure websites and applications. The website industry is changing away on a continuous as well as rapid basis from monolithic to decoupled approaches. While, developers are storming ahead with much more power than ever before. However, Netlify has been developed to cater this moment. It is offering incredible web automation technology as well as web hosting infrastructure. Netlify is offering next-generation solutions in both aspects and most amazingly, at very affordable range.

It will work by connecting your GitHub repository to pull your website's source code. After that it will run a build process to pre-render your website's pages into static HTML. The resulting pages will then be distributed and deployed across a wider range

of content delivery networks. However, whenever, a user will request to access your website, it will choose a nearest data center automatically to serve users effectively.

## 2.15 Deployment



PythonAnywhere is an online integrated development environment (IDE) and web hosting service (Platform as a service) based on the Python programming language.[1] Founded by Giles Thomas and Robert Smithson in 2012, it provides in-browser access to server-based Python and Bash command-line interfaces, along with a code editor with syntax highlighting. Program files can be transferred to and from the service using the user's browser. Web applications hosted by the service can be written using any WSGI-based application framework.

PythonAnywhere was created by Resolver Systems,[2] who also produced Resolver One, a Python-based Spreadsheet program. On 16 October 2012 the product was acquired by a new company, PythonAnywhere LLP, who will develop and maintain the product in the future, and have taken on the existing development team.

What you can do on PythonAnywhere?

1. Code, run and host python scripts.

2. Build & host Django web app.
3. Build & host Flask web app.
4. Build & host web2py web app
5. Build & host Bottle web app



### Start hosting quickly

Just write your application. No need to configure or maintain a web server — everything is set up and ready to go.



Develop anywhere

Take your development environment with you! If you have a browser and an Internet connection, you've got everything you need.



Teach and learn

PythonAnywhere is a fully-fledged Python environment, ready to go, for students and teachers — concentrate on teaching, not on installation hassles.



Amazing support

Need help with PythonAnywhere? If you get in touch, you can talk directly with the development team. Help for developers, from developers.



At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, you need to know two connected principles:

## **Version control**

### **Git**

#### **What Is Version Control?**

Version control helps developers track and manage changes to a software project's code. As a software project grows, version control becomes essential. Take WordPress...

At this point, WordPress is a pretty big project. If a core developer wanted to work on one specific part of the WordPress codebase, it wouldn't be safe or efficient to have them directly edit the "official" source code.

Instead, version control lets developers safely work through branching and merging. With branching, a developer duplicates part of the source code (called the repository). The developer can then safely make changes to that part of the code without affecting the rest of the project.

Then, once the developer gets his or her part of the code working properly, he or she can merge that code back into the main source code to make it official.

All of these changes are then tracked and can be reverted if need be.

What Is Git?

Git is a specific open-source version control system created by Linus Torvalds in 2005. Specifically, Git is a distributed version control system, which means that the entire codebase and history is available on every developer's computer, which allows for easy branching and merging.

According to a Stack Overflow developer survey, over 87% of developers use Git.

### **Advantages of using Git?**

#### **1. It makes it easy to contribute to your open source projects**

To be honest, nearly every open-source project uses GitHub to manage their project. Using GitHub is free if your project is open source and includes a wiki and issue tracker that makes it easy to include more in-depth documentation and get feedback about your project. If you want to contribute, you just fork a project, make your changes and then send them a pull request using GitHub web interface.

#### **2. Documentation**

By using GitHub, you make it easier to get excellent documentation. Their help section and guides have articles for nearly any topic related to git that you can think of.

Bottom of Form

#### **3. Showcase your work**

Are you a developer and wishes to attract recruiters? GitHub is the best tool you can rely on for this. Today, when searching for new recruits for their project, most companies look into the GitHub profiles. If your profile is available, you will have a

higher chance of being recruited even if you are not from a great university or college.

#### **4. Markdown**

Markdown allows you to use a simple text editor to write formatted documents. GitHub has revolutionized writing by channeling everything through Markdown: from the issue tracker, user comments, everything. With so many other programming languages to learn for setting up projects, it's really a big benefit to have your content inputted in a format without having to learn yet another system.

#### **5. GitHub is a repository**

This was already mentioned before, but it's important to note, GitHub is a repository. What this means that it allows your work to get out there in front of the public. Moreover, GitHub is one of the largest coding communities around right now, so it's wide exposure for your project.

#### **6. Track changes in your code across versions**

When multiple people collaborate on a project, it's hard to keep track revisions—who changed what, when, and where those files are stored. GitHub takes care of this problem by keeping track of all the changes that have been pushed to the repository. Much like using Microsoft Word or Google Drive, you can have a version history of your code so that previous versions are not lost with every iteration.

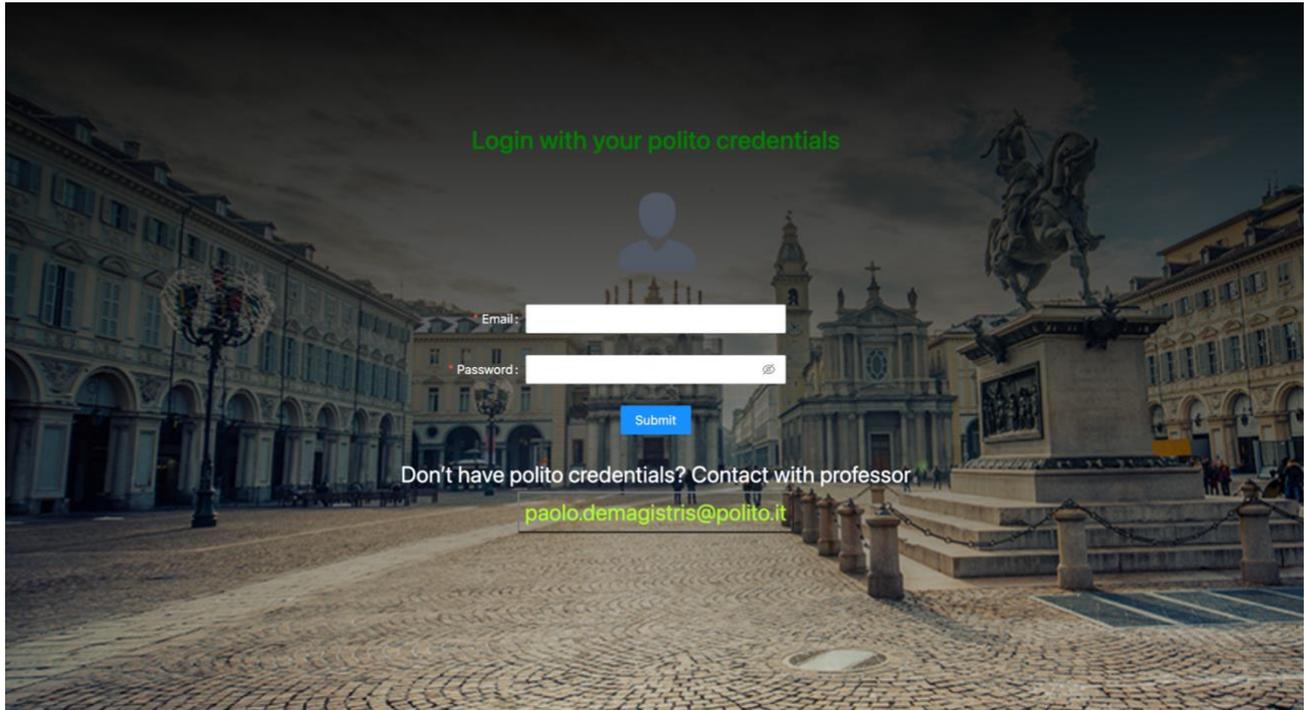
#### **7. Integration options**

GitHub can integrate with common platforms such as Amazon and Google Cloud, services such as Code Climate to track your feedback, and can highlight syntax in over 200 different programming languages.

# **3. How platform works**

## 3.1 Login

When user visit web application login page will be shown. It is first step to access home page. To access web application POLITO credentials (login and password) must be entered to the input fields. Users only enters writing their username(email) and password provided by the professor.



*Figure 3: Login page*

There made a modal which just clicking it, new users can easily send a message to professor asking access to the portal. On request portal asked student email, message and should choose to whom sending message. There are 2 options: professor and admin. New users get confirmation mail on their email address.

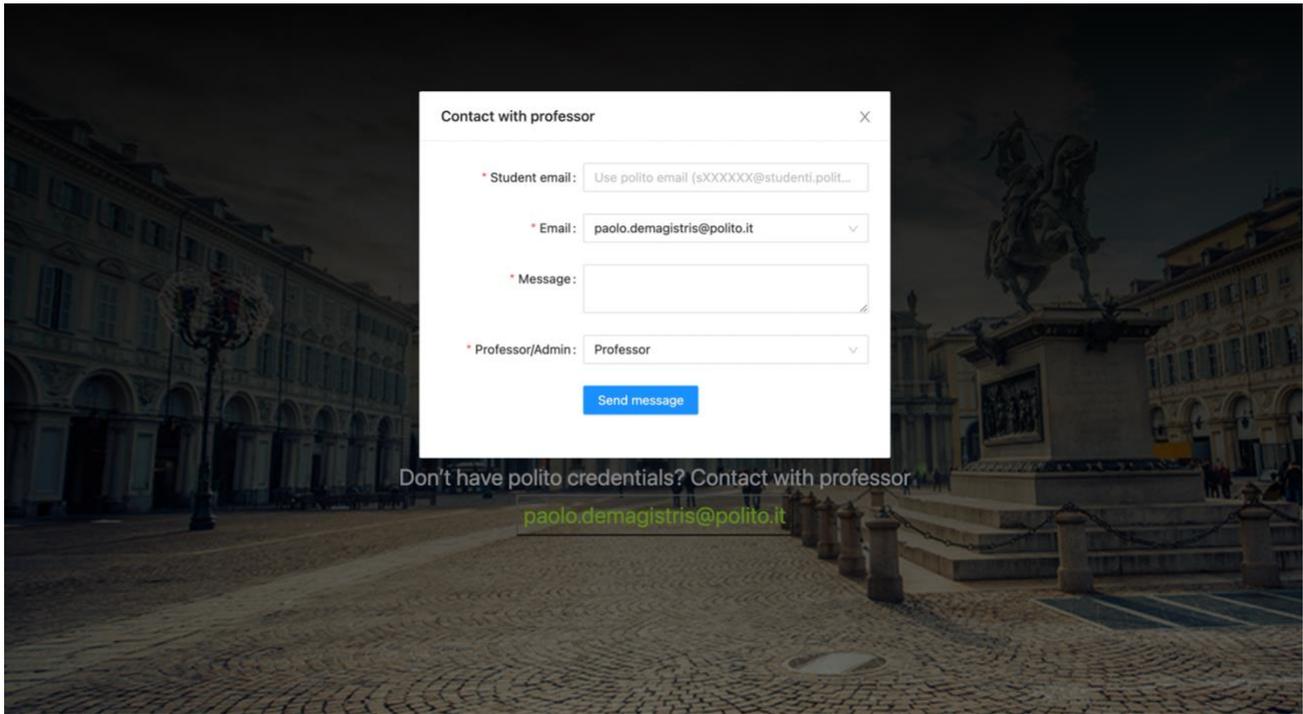


Figure 4: Contact Modal with admin/professors

When user logs in he/she will be redirected to home page where user can see tasks section.

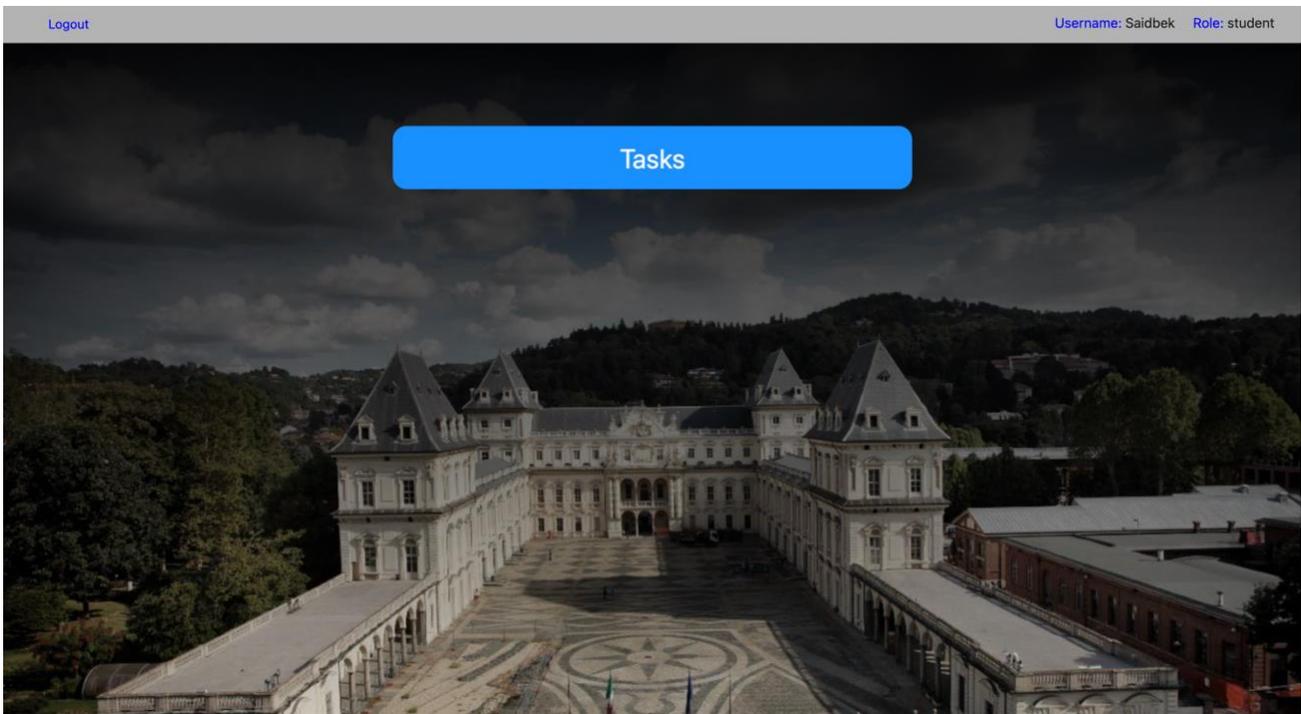


Figure 5: Home page

On the header has button Logout which user can easily logout from portal. Also, on the right-side username and role of the user will be displayed. When user press tasks user will be on tasks page. On tasks page there are list of tasks which should be completed for thesis work

### 3.2 Tasks

Logout		Username: Saidbek	Role: student
<b>Thesis title:</b> Applying the Project Management approach to the academic context and building thesis projects management framework. The front-end developer (UI/UX designer) perspective			<a href="#">Create task</a>
Title	Description	Action	
Identifying structure of web application	Discussion with partners and draw structure of the web app. Making research on the field of web application. Explore structure of to do list and its implementations. Testing other platforms and making conclusion. Determination main functions of the web applications	<a href="#">Delete</a>	<a href="#">Edit</a>
Choosing best framework and usage on web app	Check list of most popular technologies for front end development. Read brief documentation of popular frameworks and its applications. Explore Vue, React and Angular frameworks. Decide to use best framework and preparing project for coding.	<a href="#">Delete</a>	<a href="#">Edit</a>
Creating react app and connect with gitHub	Installing Webstorm and taking licence for official usage using polito credentials. Connecting it to the github profile opening new repositories in order to track all changes. Making repository private and giving access to the professors and supervisors which they can see changes and leaving comments on it	<a href="#">Delete</a>	<a href="#">Edit</a>
Determination of backend development	Discussion with backend developer how to implement API for this web app. Choosing framework among popular ones and start writing code. Testing it using Postman or Insomnia tools.	<a href="#">Delete</a>	<a href="#">Edit</a>
Identity and Access Management (IAM) for web app	Determination of the roles for web app. There will be 4 roles: Student, Professor, reviewer and admin. Giving access writes for 4 roles. Admin will have superpower who can control over all users. Also discussion actions for tasks. Editing, deleting and creating tasks	<a href="#">Delete</a>	<a href="#">Edit</a>
Layout of Login page. Determination components	Discussion UI/UX and write list of pages which will be visible for user. Examples pages are: Login, Home and Tasks pages. Also explore latest css preprocessors to use it inside web application	<a href="#">Delete</a>	<a href="#">Edit</a>
Swagger documentation for REST API	Check swagger documentation which is provided by backend developer. Test endpoints using Insomnia or Postman tool before integrating it with front end. Leave comments or feedbacks. Discussions with backend developer how to optimize code	<a href="#">Delete</a>	<a href="#">Edit</a>
Integrate the endpoints for authentication with th	Discuss how to connect API with backend developer. Installing node js package. Using axios fetching data and displaying it to the user interface	<a href="#">Delete</a>	<a href="#">Edit</a>

Figure 6: Tasks page

On the page there is button create task on clicking it user can create new task for thesis work. Below given table which illustrates list of tasks crated by the user. In front of each task has several action buttons (delete and edit).

By pressing it, user can easily delete task or edit all tasks. For example, they can enter title of the task and description.

**Edit task** ✕

Title:

Description:

\* User:  ▼

- student
- supervisor

*Figure 7: Edit task*

**Create task** ✕

Title:

Description:

\* User:  ▼

*Figure 8: Create task*

### 3.3 Backend Part

On the backend part which admin can control and give access to user used Django framework. And workable code made documentation using Swagger tool. There are several API which are used to make website dynamic and sending request to the server. On admin panel, users may be deleted or edited by moderator/admin who is responsible for this action.

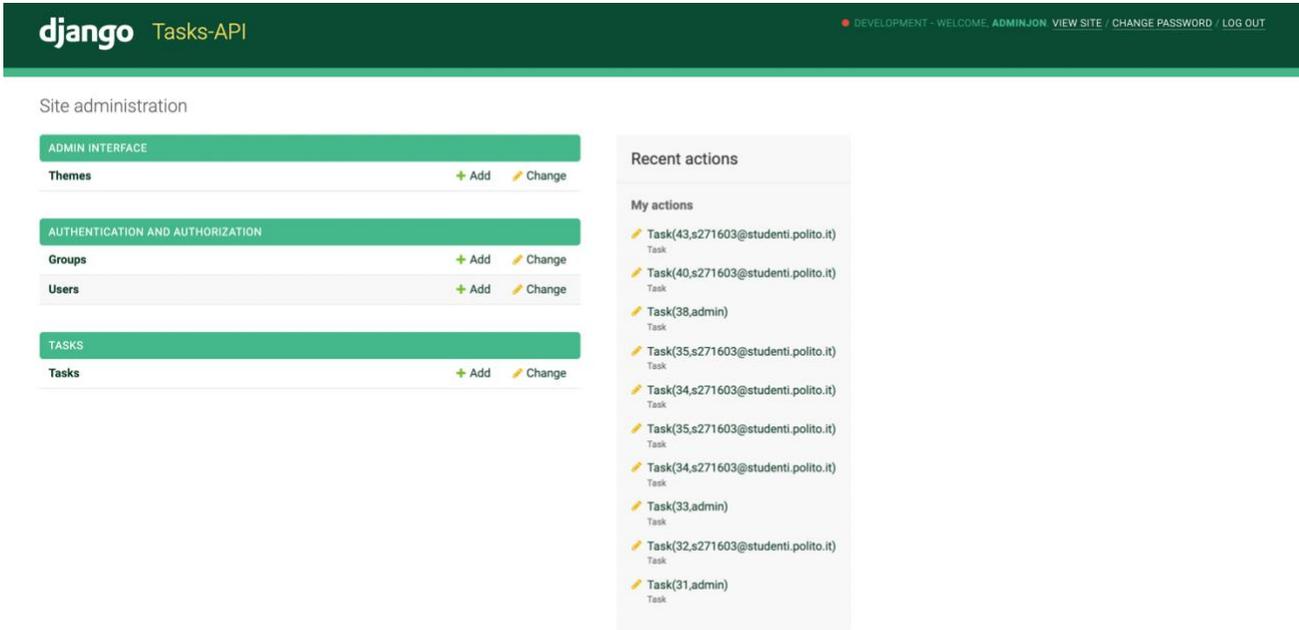


Figure 9: API platform for web app

There created 3 group members with difference accesses. Student, professor, reviewer. Student can do all actions which are explained on front end part. Professor cannot delete but edit the task. Reviewer can only see task and maybe later he can leave a comment related to the task.

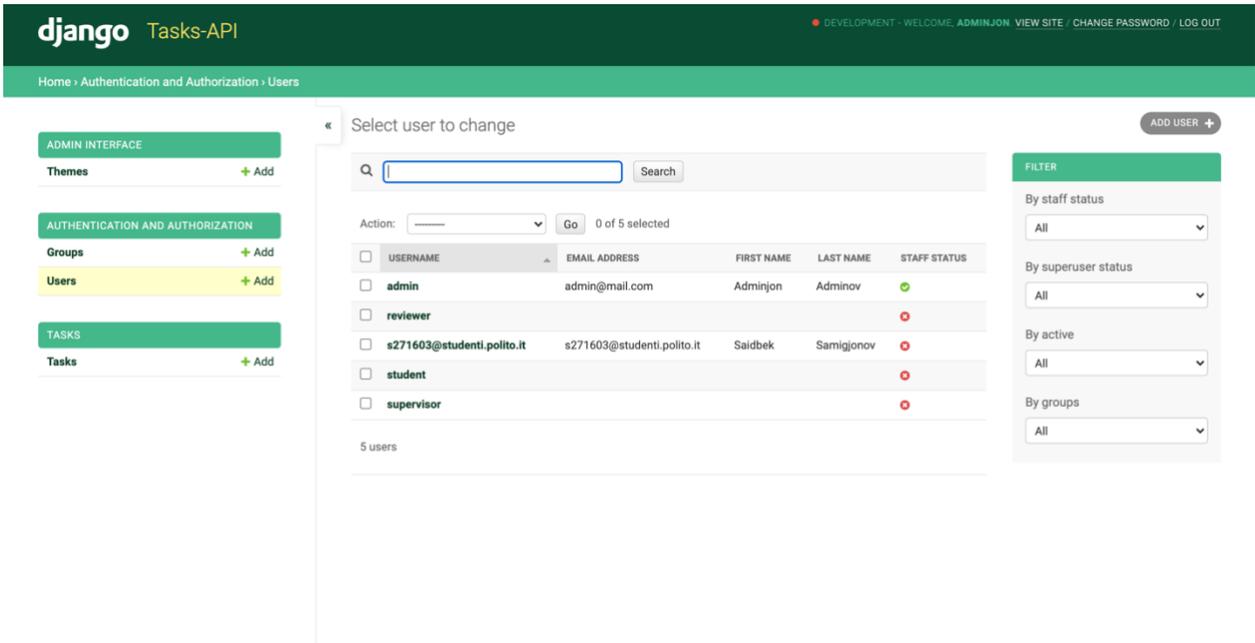


Figure 10: Access rights control panel

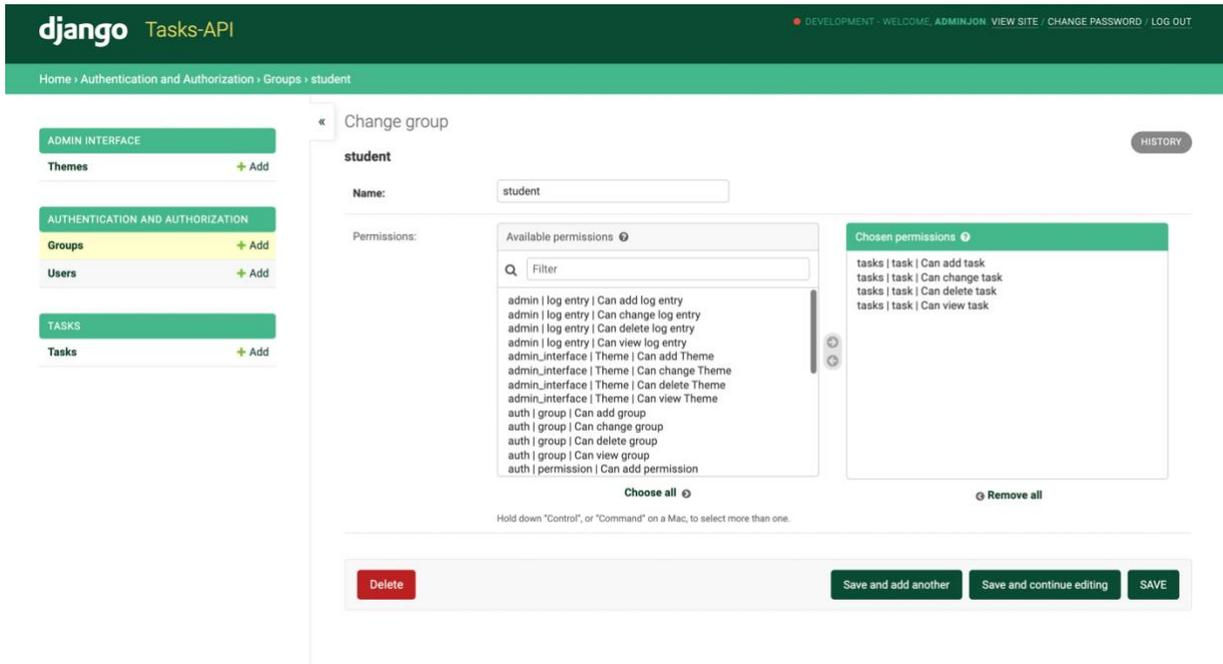


Figure 11: Access rights for student

# 4. Conclusion

The main goal of this thesis work was to lift thesis taking process into the next level and build a bridge between professors and thesis defenders (students) using modern way and creating a new web application. In this fast-changing world, time is being valuable than ever. Applying this tool will help both side (students and professors) to make their collaboration closer than ever. However, this is not a final version of the web application.

## Findings

Core concept of the project is Processing implementation of communication and information exchange in a digital environment for project management framework. Using latest technologies in Information system field we can create a app which will be tool for Digitalization in Project management. Our experiment demonstrated that a basic artifact of the project management information such as a task list can be easily ported to a complex framework of servers and Information and Communication technologies (ICT). It is important that this result was reached by detailed exploration of core technologies available and ready for use. The idea is not to re-invent bicycle, but to lift make app which help on transporting information and make process smarter.

There are three possible ways of improving of the application in future perspective:

- Reviewers will be able to comment student's thesis title. This will help for student who is making a thesis to get feedbacks and comments and improve task lists adding or removing them.
- Providing discussion and chatting possibility between professors and students. Thesis work is important process and communication with professors plays an important role in this thesis development process
- Adding support to a file management. Sometimes professors may want to see status of the thesis work looking to the file which is filling by student. Student

can easily share it is file to the system and discuss with professor how to improve thesis work.

The process which could be easily understood by the students and make easy transition from tradition way of thesis development to the proposed way of thesis development. This dissertation is limited to the development of thesis statements in which students work with internal supervisors of the Politecnico di Torino. This research paved the way for the future expansion of the heuristic method developed for the paper to include external monitors (participants) such as research centers and companies. In this fast-changing world, this Project Management approach methodology represents a solid foundation to build on and will be one of the useful features of thesis preparation process.

# Webography

1. <https://medium.com/swlh/build-your-first-rest-api-with-django-rest-framework-e394e39a482c> - check this one
2. <https://www.codecademy.com/articles/back-end-architecture> - CHECK THIS ONE
3. <https://www.thirdrocktechkno.com/blog/business-benefits-of-using-react-for-frontend-development/> - check this source
4. <https://blog.stackpath.com/web-application/>
5. [https://www.ictshore.com/software-design/web-application-structure/attachment/sfw0001-01-web application structure/](https://www.ictshore.com/software-design/web-application-structure/attachment/sfw0001-01-web%20application%20structure/)
6. <https://www.techslang.com/definition/what-is-frontend-development/>
7. <https://www.zdnet.com/article/front-end-vs-back-end-development/>
8. <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>
9. <https://clockwise.software/blog/angular-vs-react-vs-vue/>
10. <https://www.clariontech.com/blog/7-advantages-of-reactjs-for-building-interactive-user-interfaces>
11. <https://www.thirdrocktechkno.com/blog/business-benefits-of-using-react-for-frontend-development/>
12. <https://www.codecademy.com/articles/back-end-architecture>
13. <https://www.temok.com/blog/backend-web-development/>
14. <https://reinvently.com/blog/fundamentals-web-application-architecture/>
15. <https://www.interviewbit.com/blog/backend-developer-skills/>
16. <https://cybercraftinc.com/backend-web-development-technologies-in-2021-2022-2/>
17. <https://djangostars.com/blog/top-14-pros-using-django-web-development/>
18. <https://www.monocubed.com/django-web-framework/>
19. <https://www.ideamotive.co/blog/why-and-when-should-you-consider-django-for-web-development>
20. <https://medium.com/swlh/build-your-first-rest-api-with-django-rest-framework-e394e39a482c>
21. <https://swagger.io/blog/api-documentation/what-is-api-documentation-and-why-it-matters/>
22. <https://agilitycms.com/resources/posts/what-is-netlify-and-why-should-you-care-as-an-editor>
23. <https://www.pythonanywhere.com/>
24. <https://angular.io/>
25. <https://reactjs.org/>
26. <https://vuejs.org/>
27. <https://github.com/>
28. <https://www.netlify.com/>