The National Institute of Posts and Telecommunications - RABAT

Project Report: Modern Web Development

Carried out by :

TAOUCHIKHT Khouloud **LOUATI Oussama**

Supervised by :

Pr. HAQIQ Abdelhay

Academic year 2022/2023

Source code for Claimbook project:

https://github.com/OssamaLouati/Claimbook-Project

Link of the video demonstration:

https://drive.google.com/drive/folders/1x14T951dDpwHenZ9i8hKAy6NkfBpg-uT?usp=share_link

Url of the webiste:

We apologize, but unfortunately, we cannot host the Claimbook application for free. We are using AngularTS and Spring Boot for the backend, and there are no free hosting providers available for these technologies. Hosting this website would require paying the fees, which we are not able to do at this time. We hope you understand, and thank you for considering our project .

.

Table of contents

List of Figures

Résumé

Ce rapport résume les travaux accomplis dans le cadre de notre projet pour le cours du INGÉNIERIE DES APPLICATIONS WEB MODERNES à l'Institut National des Postes et Télécommunications. Notre objectif principal était de créer la plateforme "Claim Book" qui vise à numériser le service de gestion des réclamations au sein de l'INPT.

Notre travail s'est déroulé en plusieurs phases : la création d'un serveur Web Spring pour l'authentification des étudiants et des techniciens, ainsi que pour la gestion des réclamations des utilisateurs, et la mise en place d'un frontend pour la plateforme avec Angular. Nous avons également créé un serveur Flask et un modèle d'apprentissage automatique pour recommander le meilleur cochambre à un étudiant et pour fournir des solutions détaillées aux problèmes rencontrés par le technicien. Le projet implique plusieurs concepts, tels que l'apprentissage supervisé, le pattern MVC, Spring Security.

Keywords: Supervised Learning, MVC pattern , Spring, Angular , Flask

Abstract

This report summarizes the work carried out in the context of our project for the course on Development Web Applications at the National Institute of Posts and Telecommunications. Our main objective was to create a platform aimed at digitizing the claims management service within the INPT, with our "Claim Book" platform being the chosen support.

Our work was carried out in several phases: the creation of a Spring Web server for the authentication of students and technicians, as well as for the management of user claims, and the implementation of a frontend for the platform with Angular. We also created a Flask server and a machine learning model to recommend the best roommate to a student and provide a detailed solution to the problems encountered by the technician. The project involves several concepts, such as supervised learning, the MVC pattern, and Spring Security.

Keywords: Supervised Learning, MVC pattern , Spring, Angular , Flask

General Introduction

Most of the INPT students who are housed in the residence encounter several technical problems (power outage, hot water, broken handle). INPT offers a notebook solution that is outdated and inefficient, which demotivates students. In addition, first-year students may not be familiar with each other, which makes it difficult for them to decide which roommate to choose. To solve this problem, we thought about digitizing this service using a web application and a Machine Learning model to save human resources (technicians, students, etc.).

In this report we will go through all the steps to showcase the utility of the project in the general context, and we will tackle the technical aspect in the analysis and the design then move on to the
implementation by stating the tools that we used.

In the first chapter, we will talk about the general context of the project to put you in situation, then we will move on to the analysis and the design in the second chapter. The latter will contain a description of the use cases and the needs of each of them in terms of technologies and architecture. And finally in the 3rd chapter we will mention the tools used and insert screenshots of the application.

Chapter 1

General context of the project

This part allows us to present the circumstances of our project in a general way, you will discover the subject, the needs and the planning of the project.

1.1 Subject presentation

The housing facility provided by INPT is the only option for many students, but unfortunately, it is not without its issues. Many students face recurring technical problems like power outages, lack of hot water, and broken handles. The current solution provided by the institute, which is a notebook system, is outdated and ineffective. Moreover, the issue of students struggling to choose their roommates due to unfamiliarity with each other is not covered by the solution therefore some students end up with unsuitable roommate .

To tackle this pressing issue, we came up with a comprehensive solution that leverages the power of technology. By digitizing the service using a web application and a Machine Learning model, we can eliminate the need for human resources, such as technicians and students. This new platform will provide an easy-to-use interface that allows students to log their complaints and receive prompt solutions. Additionally, the Machine Learning model will recommend the best accommodation option to each student, eliminating the need for them to choose roommates themselves , Furthermore, we simplify the technician's work by providing a machine learning model that can advise them on potential solutions in case they encounter a technical problem that they are unsure how to repair. Overall, our solution aims to provide a better living experience for students at INPT and improve their academic performance by reducing unnecessary distractions.

1.2 Need existence

We conducted student and technician interviews to demonstrate the necessity of the project and gain insight into the problem or opportunity it addresses. This will also aid in directing the project's focus towards accomplishing specific objectives.
Here are some interview questions and responses:
Student questionnaire:

1. Have you ever had a problem in your room?
- Yes, several times
- Yes, almost every week
2. What do you do if this happens?
- I fill out the anomalies notebook
- I wait until the end of the week to go out and bring my needs (lamp, door handle...)
3. When filling out the anomalies notebook, how long does it take for technicians to intervene?
- It depends, sometimes they intervene immediately and sometimes they never intervene
- There is a big problem at this point. It's impossible to know if the technicians have already come or not. Once, the bathroom was leaking water in the middle of the room and filling it with water. I recorded the complaint several times. The cleaning lady told me that the technicians came to the room, yet the problem lasted for about half a semester, and unfortunately, with my busy studies, I didn't have time to meet the technicians and make them understand that they had not yet solved the problem, and I had no choice but to continue recording the complaint day after day.
4. When technicians intervene, is the problem definitively resolved?
- Unfortunately, it's not always the case. Sometimes they overlook the problem with anything, and after two days, it comes back.
5. What are your criticisms of the system that INPT relies on to manage anomalies? What are the shortcomings of this system?
- I was not aware of the existence of an anomaly management system within the campus. A sign should be placed to identify it within INPT.
- The anomalies notebook is available during the same study session, so I don't have time to fill it out.
- I'm too lazy to go get the notebook.
- There is a communication problem between technicians and students. We need a way for the technician to show that he has already been to the room, and the student as feedback will express their satisfaction.

Technician questionnaire:

1. As one of the technicians responsible for resolving issues that arise on the INPT campus, describe to us the most significant problems you face in your work.
    - Most students' writing is illegible
    - Sometimes I think I have solved the problem, but unfortunately the day after, I find the same complaint
    - Students are not always satisfied with our work
    - Description of the problem is not detailed enough. Sometimes I bring insufficient materials to solve a problem, and I have to go back
    - I don't have an exact schedule that shows me when I should be present. Sometimes I come, but I only find one complaint.

1.2 Functional requirement:

Student:

- Authentication
- Claim a problem in the dorm
- Track the status of the claim
- Edit a claim
- Cancel a claim
- Getting suitable roommate

Technician:

- Authentication
- view all claim
- confirm the resolution of a claim
- get detailed solution

1.3 Besoins non fonctionnels

Once we have dealt with the functional side, we must also consider the non-functional needs, for example:

• Confidentiality

• Easy-to-use

• Progressive Change

• Adaptability

Chapter 2

Analysis and Conception

This chapter will present an analysis of our project.

2.1 Actors:

The actors in our application are mainly as follows:
The developer: He can manages user accounts, he can also add
services to the application (Add more data)

The student: view/add/edit/cancel claim , Obtain the 3 recommended roommates .

The technician: view/cancel claim , get response of question

2.2 Use case diagram

A use case diagram captures the behavior of a system, subsystem,
class, or component as an outside user sees it. It splits the functionality of the system into coherent units, use cases, that make sense to the actors. The use cases make it possible to express the need of the users of a system, they are therefore a user-oriented vision of this need, unlike a computer vision.

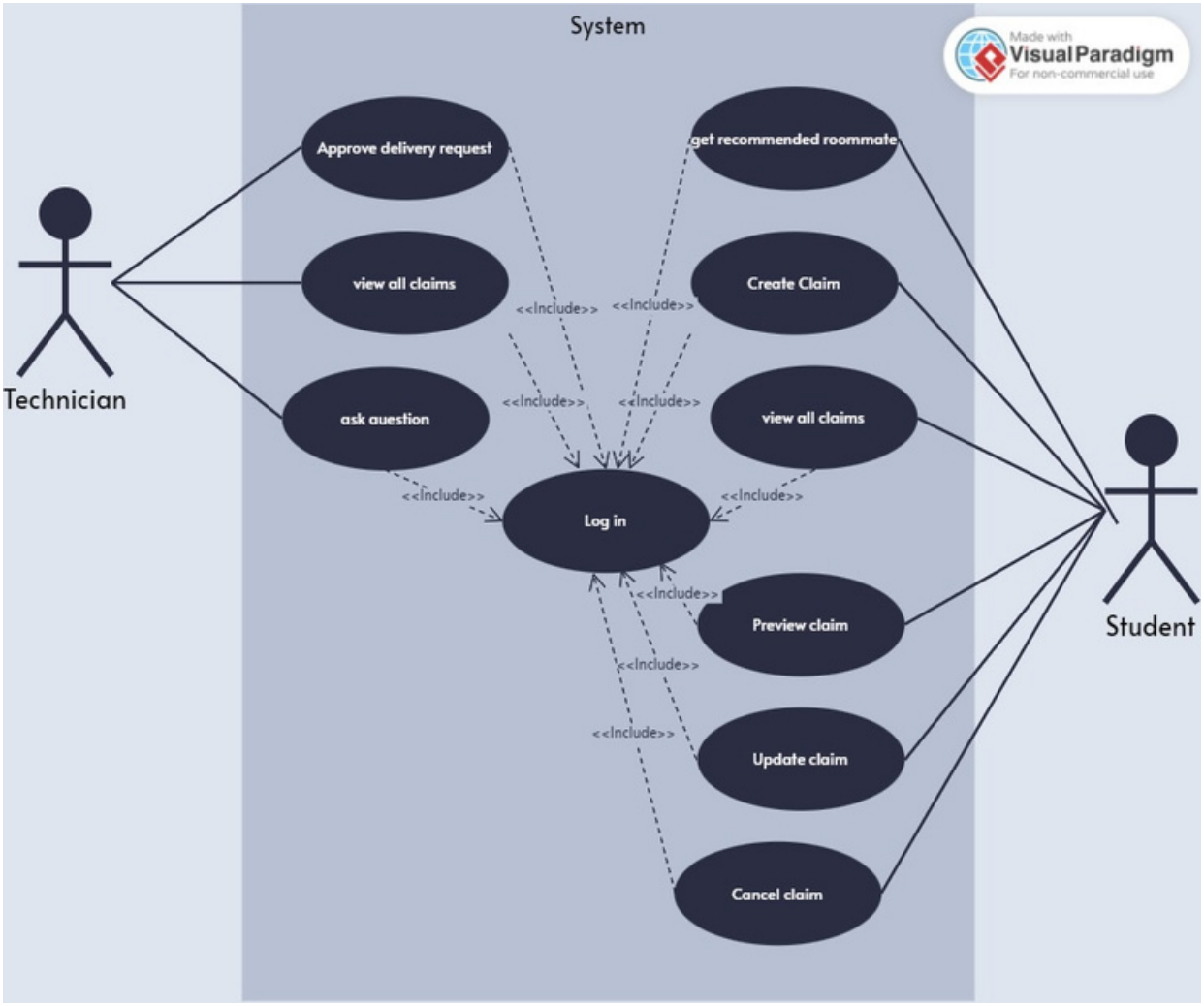


Figure 2.1: Use Case Diagram

2.2.1 Class diagram

A class diagram is a type of UML (Unified Modeling Language) diagram that describes the structure and behavior of a system by representing its classes, attributes, methods, and relationships. It provides a graphical representation of the system's architecture, showing how the different classes are related to one another and how they collaborate to achieve the system's functionality.

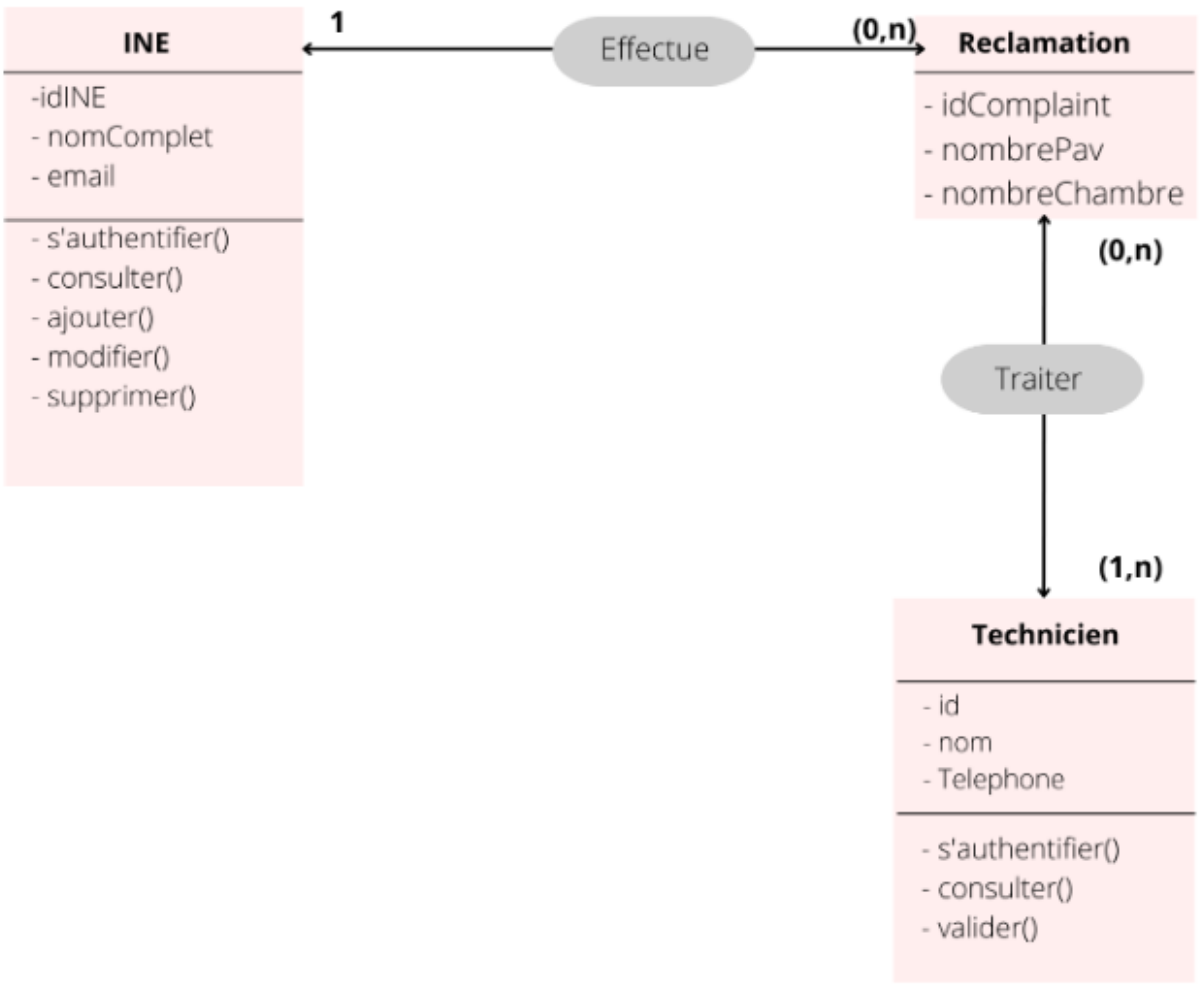


Figure 2.1.1: Class Diagram

2.3 Design and analysis of general architecture of the Platform

After presenting what the application is all about. In this part of the chapter, we will provide the general architecture of our application, including the workflow for user interaction with the application, and how the server will handle the authentication and process of user queries. And the problems of securing the information from client side to the backend

2.3.1 Different Elements of Architecture

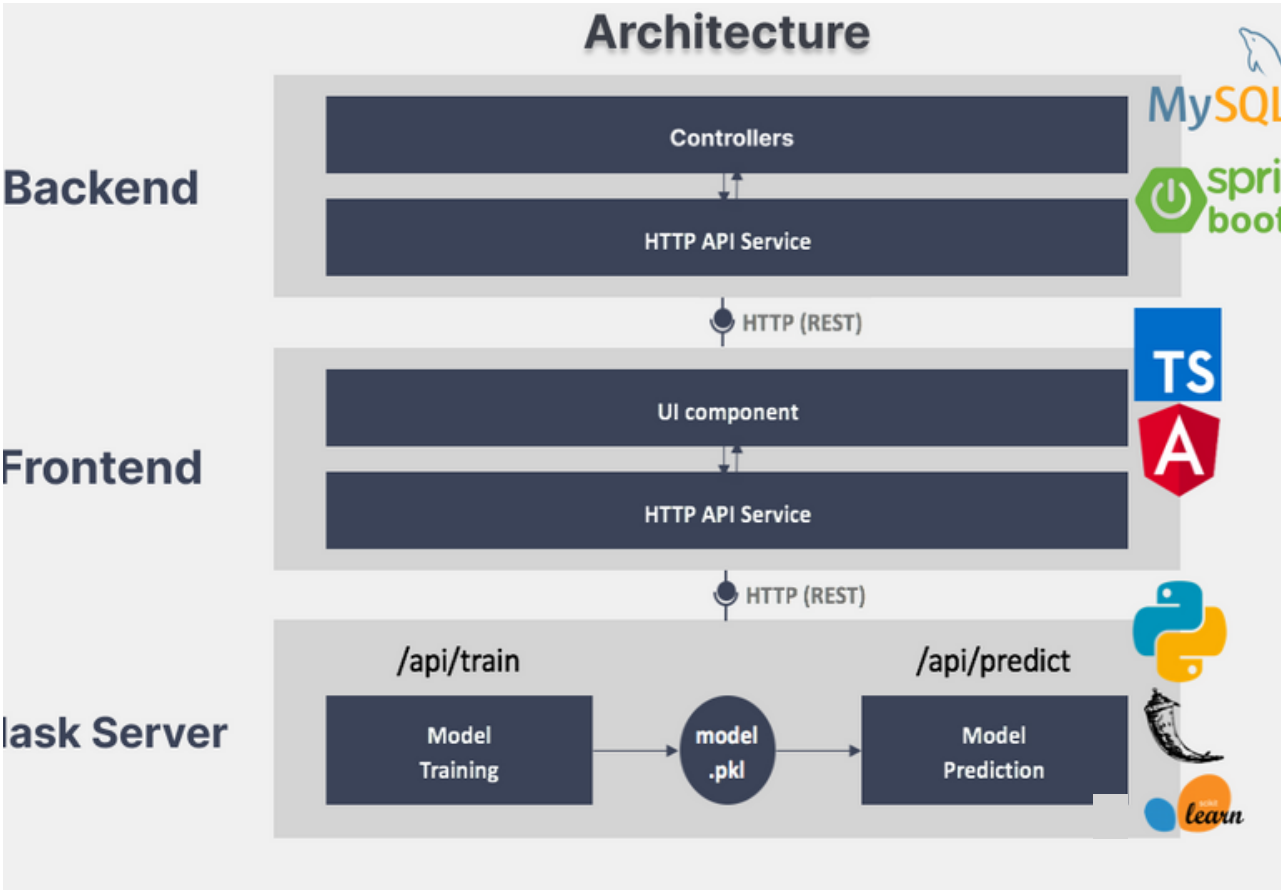


Figure 2.2: Architecture

Client Side : In web development, 'client side' refers to everything in a web application that is displayed or takes place on the client (end user device). This includes what the user sees, such as text, images, and the rest of the UI, along with any actions that an application performs within the user's browser.

Angular is the client-side in our project, It is an open-source, TypeScript-based framework designed by Google. Angular offers many advantages for which we have chosen to develop with, among these advantages:

=> Automatic synchronization with Two-way data binding.

=> Consistency and reusability of the code, thanks to the Angular CLI (line interface of command) and the documentation style guide

=> Default Ivy renderer.

=> Declarative and intuitive user interface that eliminates the need to invest a lot of time in program streams and schedule what loads first.

Server Side: Much like with client side, 'server side' means everything that happens on the server, instead of on the client. In the past, nearly all business logic ran on the server side, and this included rendering dynamic webpages, interacting with databases, identity authentication, and push notifications.
For the server side we used Spring Boot which is Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications.
Our application is based on the Restful model, so all actions are triggered by exchanging data through between controllers and the view and this entire process is satisfied through the backend which is responsible for managing:

Dataset: the database is used for storage purposes. Here we use MySql databases for production, using hibernate to do the communication (ORM).
The dataset is stored in a database managed by the backend. Communication between the frontend and the backend is done through http requests by communicating encrypted data in JSON (Restful API) format. Thus, the concerned controller processes the requests and allows to extract the required data from the database, format them, and then generate the view to the user.

Users: The backend is the part responsible for making a set of routes available to the different users of the system. He will be the intermediary between the devices and the different client applications

(direct the user during his navigation within the application), this is organized by granting controllers for each category of users (student and technician).

Data security: The backend part is responsible for managing user access rights by giving rights to each authenticated user according to the authorizations he has. For example, a student cannot access the information of other students .

2.4 General architecture and workflow

Now that all the elements are known to the reader, we can dive into the general architecture and see how each component of our application interacts with other components. thus we will establish the knowledge necessary to see why the MVC approch is great. and can now discuss what kind of problems it solves for our use case, and also what kind of problem it generates and how we will handle these problems.

Ideal workflow:

By using requests to the Spring server, The authentication process validates the credentials of the client, who sends an HTTP request that includes the page number, email, and password. Once authenticated, the student user can view all their claims, create new ones, edit existing ones if they have not been processed, or cancel specific claims. the student can also view his profile, including their name, bio, skills, gender, grade, and major. If the student does not have a roommate, there is a section that recommends three students, and this request is delegated to the Flask server to extract the required information from the student's profile. The server then delegates the information to the /predict endpoint, where the machine learning model loads and makes predictions on the provided data. The response is then wrapped and delegated to the Angular frontend. If the student accepts a recommendation, a notification is sent to the other student, and if they both accept, they become roommates.
The technician, when authenticated, can view all claims . Once a technician solves a claim, he can mark it as done , or he can reject it if the request is illogic. If a technician encounters a problem they cannot solve, they can write a question and get a detailed response from the Flask server that uses GPT3 service to generate the response.

2.5 Design patterns:

in this section, we will talk about the design patterns we used to develop our development.

2.5.1 Inversion of control and dependencies injection:

The spring framework is based on these two concepts which offer a production-ready design pattern.

2.5.2 MVC model:

We used the famous MVC model (with small customizations because we don’t have a "VIEW" here)

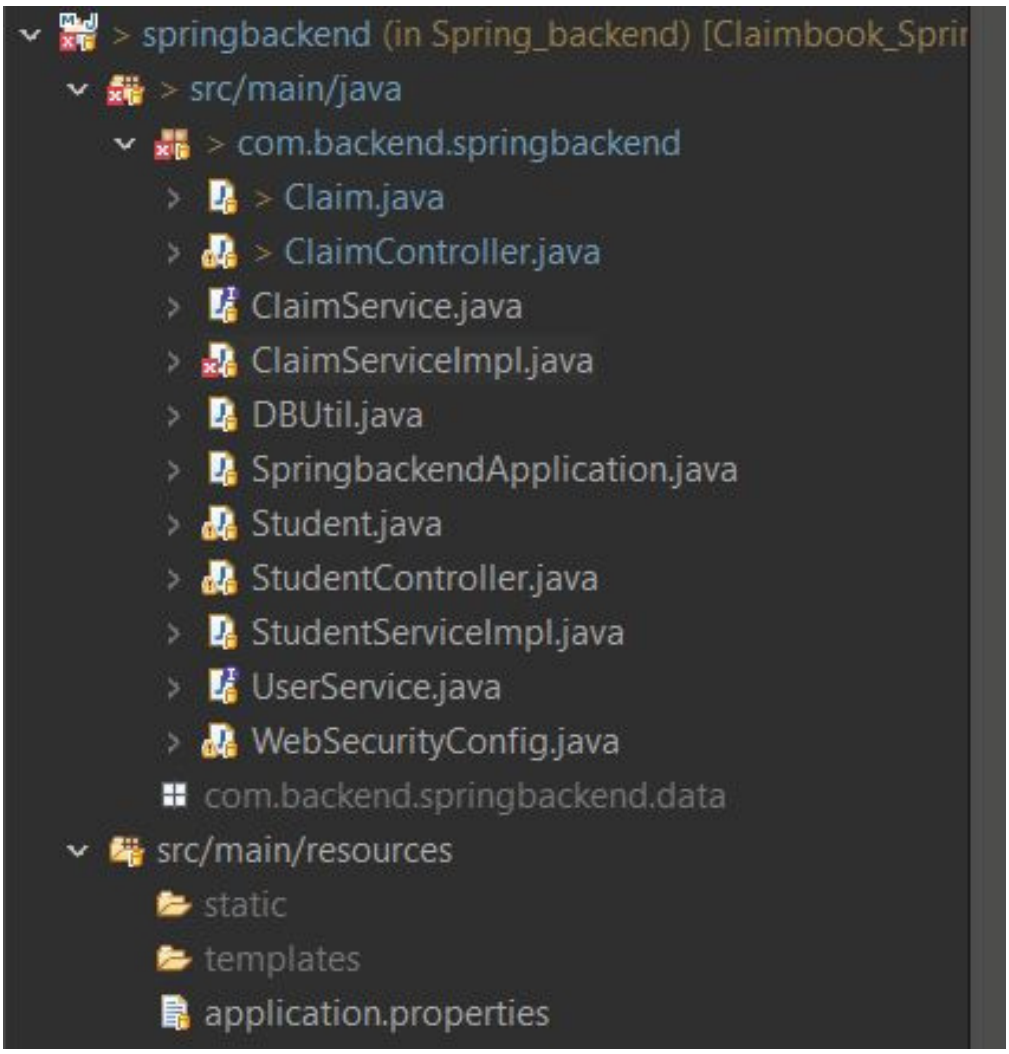


Figure 2.3: MVC Model

2.5.3 ORM:

An object-relational mapping (ORM) is a type of computer program that interfaces between an application program and a relational database to simulate an object-oriented database. This program defines correspondences between the schemas of the database and the classes of the application program. We could designate it by this, “as a layer of abstraction between the object world and the relational world”. Because of its function, we find this type of program in a large number of frameworks in the form of an ORM component that has been either developed or integrated from an external solution.

```java
@Data
@NoArgsConstructor
@Entity
public class Student {

    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    private String email;
    private String password;
    public String name;
    private String bio;
    private String niveau;
    private int room;
    private int pavillon;
    private boolean roommate;
    private String filiere;
    private String skills;
    private String avatar;
    private String gender;
```

Figure 2.4: ORM Entity of Company

2.6 Conclusion

In this chapter we have presented in a global way the main stages of the analysis and design of our application according to the different diagrams, in order to compelte the implementation phase. The next chapter will be devoted to design of the application.

Chapter 3

Design

3.1 Introduction

After the conception stage, We made the app logo design. Then, We made a model of the application to have a clear image on the structure and the design of the application.

3.2 LOGO

To give an identity to our platform, we chose a logo, which contains the name of the platform "Claimbook" . The word "claim" implies that the site is related to managing requests for compensation or addressing issues that require attention, and "book" suggests that the site is a place where these claims are recorded and organized in a systematic way.

Figure 3.1: LOGO

3.3 PROTOTYPE

In order to give shape to our solution before starting the development, We made the prototype, here are some interface of the prototype

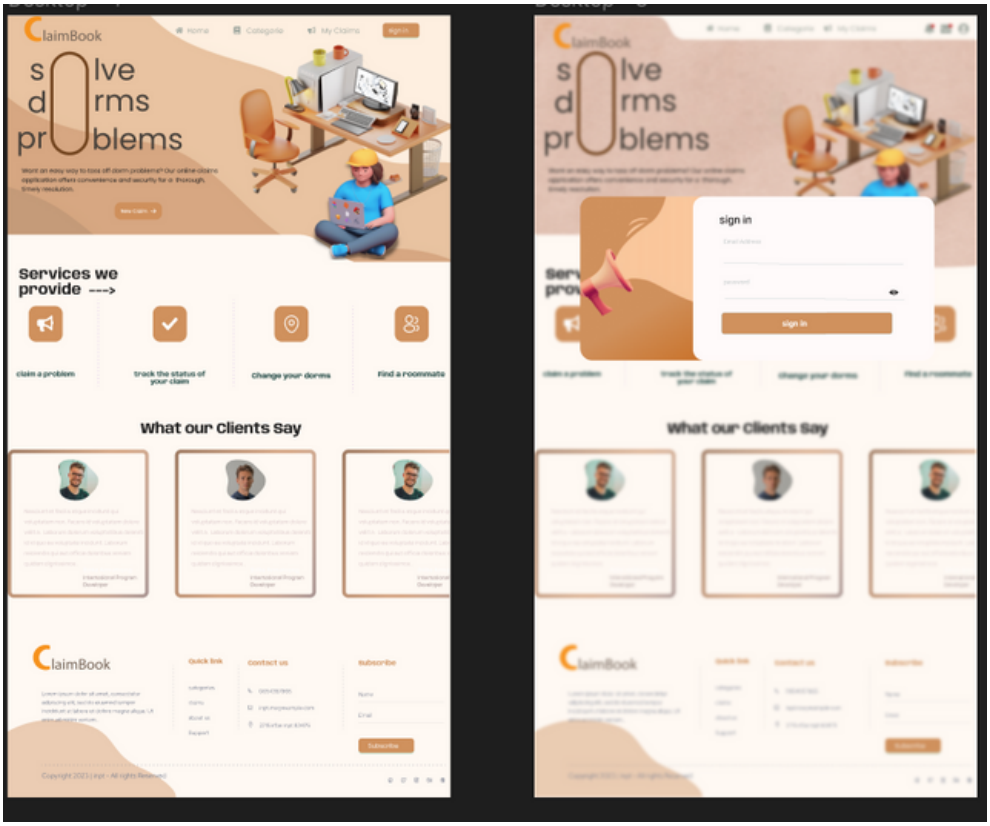


Figure 3.1: PROTOTYPE

to consult the whole prototype you can accessed the link below:

https://www.figma.com/file/IiYtbsgnNTUUqAzcru2EHM/student's-dorms-claims?node-id=0%3A1&t=ai2ksqGU3lbt5kMZ-1

3.4 Conclusion

In conclusion, the design phase is a critical step in the development of a web project. During this phase, various aspects of the project are considered and defined, such as the project scope, user requirements, technical requirements, and project goals. This phase is also when the user interface and user experience are designed, and the project architecture and database schema are planned.

Chapter 4

Machine Learning

4.1 Introduction

Our platform must be able to predict the recommended roommate of student from his profile. After getting the data, we will build a machine learning model to do this.

Machine Learning can be defined as an artificial intelligence technology that allows machines to learn without having been previously programmed specifically for this purpose. Machine Learning is explicitly tied to Big Data, as to learn and grow, computers need streams of data to analyze, to train on.

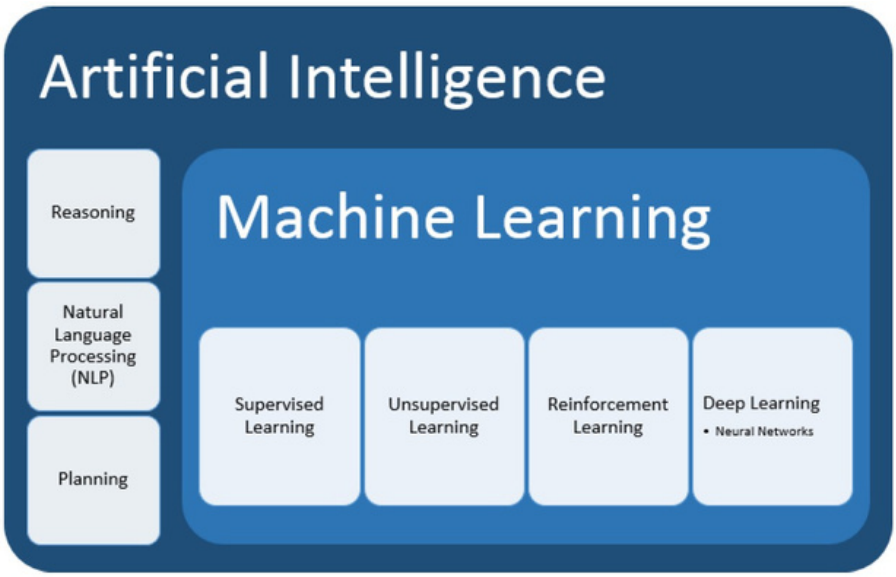


Figure 4.1: AI Machine Learning

there are different type of Machine Learning, in our case we will build a recommendation system that is Supervised Learning service. However we must go through several steps to achieve a model capable of performing this task.

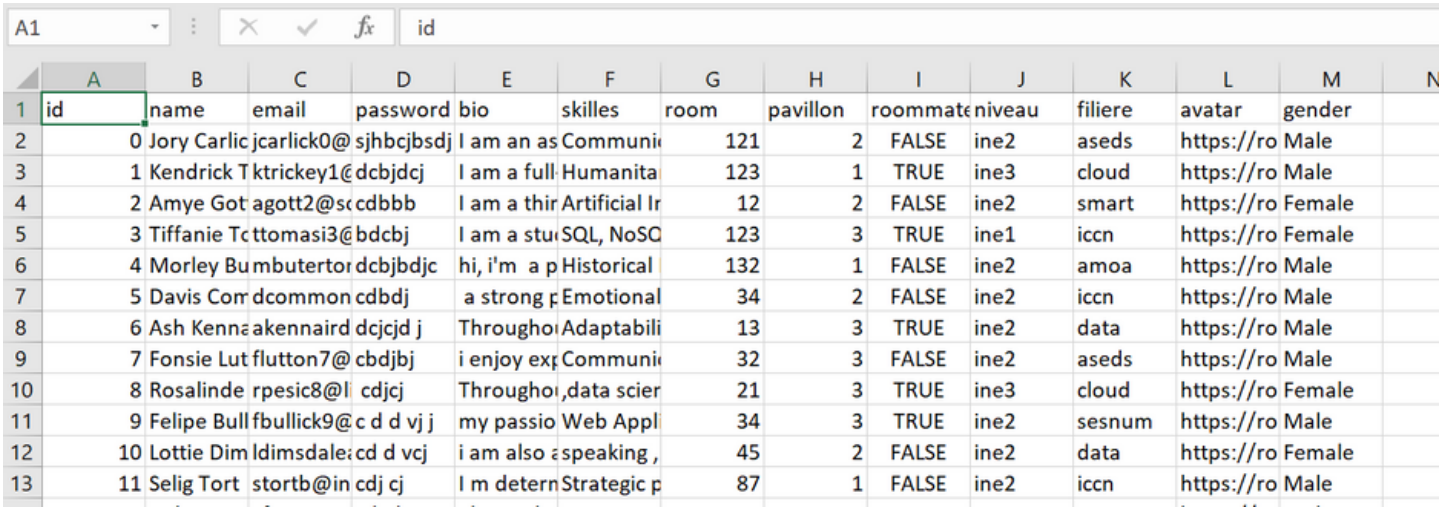
4.2 Define the problem

The first step is to clearly define the problem and what kind of recommendation system is needed. In this case, the goal is to recommend potential roommates to a student based on their profile.

Figure 4.2: Define the problem

4.3 Data collection

The second step is to collect data on students' profiles, including attributes such as name, gender, major, skills, interests, and room preferences. This data can be obtained through a student's profile form our school's website.



A1 | id

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id	name	email	password	bio	skilles	room	pavillon	roommate	niveau	filiere	avatar	gender
2	0	Jory Carlic	jcarlick0@	sjhbcjbsdj	I am an as	Communi	121	2	FALSE	ine2	aseds	https://ro	Male
3	1	Kendrick T	ktrickey1@	dcbjdcj	I am a full	Humanita	123	1	TRUE	ine3	cloud	https://ro	Male
4	2	Amye Got	agott2@sc	cdbbb	I am a thir	Artificial Ir	12	2	FALSE	ine2	smart	https://ro	Female
5	3	Tiffanie Tc	ttomasi3@	bdcbj	I am a stu	SQL, NoSC	123	3	TRUE	ine1	iccn	https://ro	Female
6	4	Morley Bu	mbutertor	dcbjbdjc	hi, i'm a p	Historical	132	1	FALSE	ine2	amoa	https://ro	Male
7	5	Davis Com	dcommon	cdbdj	a strong p	Emotional	34	2	FALSE	ine2	iccn	https://ro	Male
8	6	Ash Kenna	akennaird	dcjcjd j	Throughoı	Adaptabili	13	3	TRUE	ine2	data	https://ro	Male
9	7	Fonsie Lut	flutton7@	cbdjbj	i enjoy exp	Communi	32	3	FALSE	ine2	aseds	https://ro	Male
10	8	Rosalinde	rpesic8@l	cdjcj	Throughoı	,data scier	21	3	TRUE	ine3	cloud	https://ro	Female
11	9	Felipe Bull	fbullick9@	c d d vj j	my passio	Web Appl	34	3	TRUE	ine2	sesnum	https://ro	Male
12	10	Lottie Dim	ldimsdale	cd d vcj	i am also a	speaking ,	45	2	FALSE	ine2	data	https://ro	Female
13	11	Selig Tort	stortb@in	cdj cj	I m detern	Strategic p	87	1	FALSE	ine2	iccn	https://ro	Male

Figure 4.3: Collect data

4.4 Preprocess the data

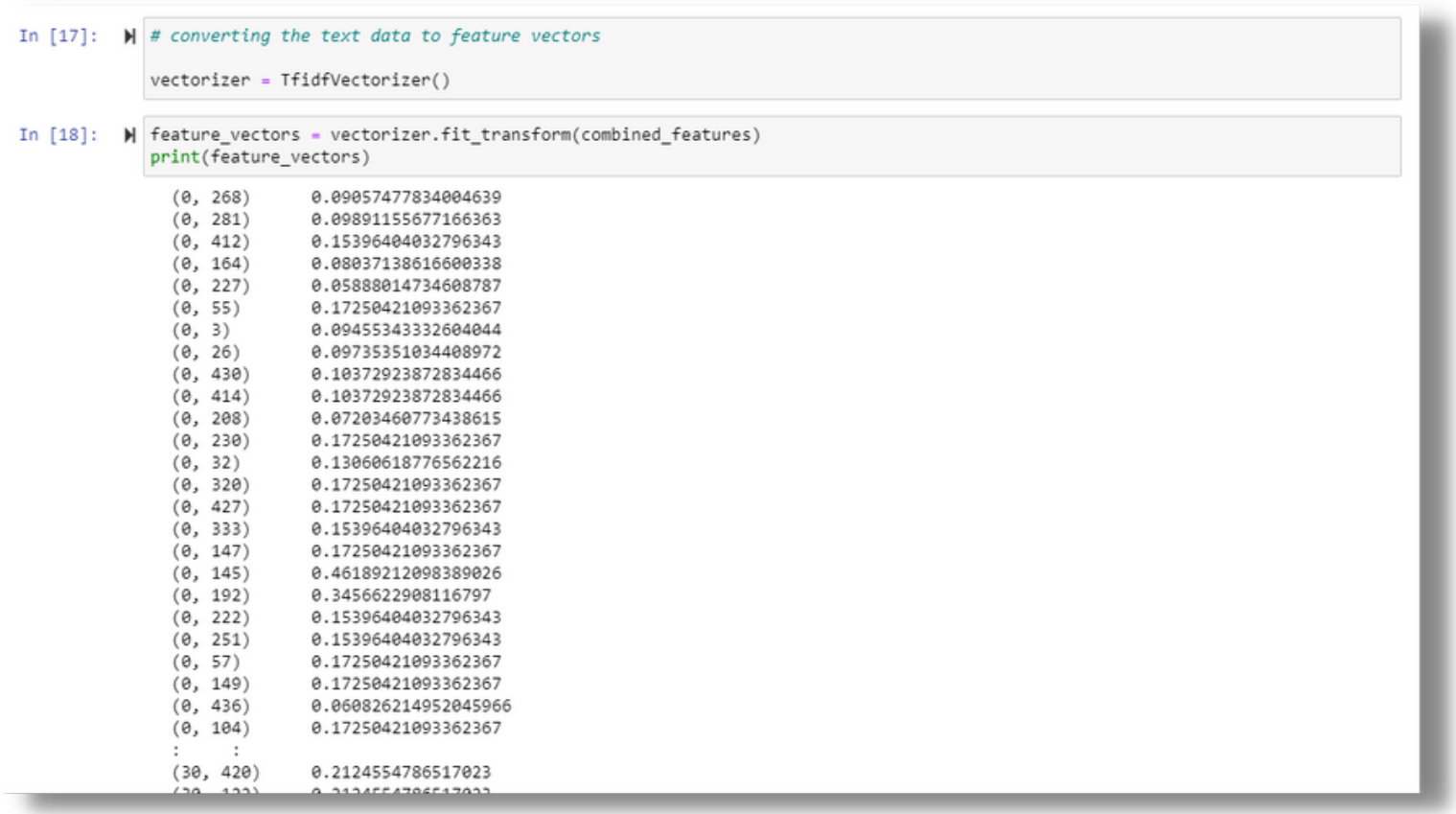
After the data understanding step, the next step in the lifecycle stages is data preparation. This step is also known as Data Cleaning or Data Wrangling. It includes steps such as selecting relevant data, integrating data by merging datasets, cleaning, managing missing values by deleting or imputing them with relevant data, dealing with erroneous data by removing them, also checking for outliers and dealing with them.

```
In [15]:
# selecting the relevant features for recommendation

selected_features = ['skilles','bio']
print(selected_features)

['skilles', 'bio']

In [16]:
# combining all the 5 selected features

combined_features = data['skilles'] + ' ' + data['bio']
print(combined_features)

0     Communication skills , Python , Java , C++ I a...
1     Humanitarian Assistance , visualization , Data...
2     Artificial Intelligence , Machine Learning , D...
3     SQL, NoSQL, HTML, CSS, JavaScript I am a stude...
4     Historical Research , java , Emotional intelli...
5     Emotional intelligence , java , c++ ,Windows, ...
6     Adaptability , speaking , html , css , Data an...
7     Communication , python , c# , java , react , M...
8     ,data science, management machine learning , d...
9     Web Applications , mobile developpement , mach...
10    speaking , html , css , javascript , machine l...
11    Strategic planning, Time management , java , h...
12    routing and switching, firewalls, java , devop...
13    photoshop , figma , mac , css I am a student p...
14    Mentorship , coaching , react , java , c++ I h...
15    DevOps , docker , listening, clear , concise s...
16    Artificial Intelligence , Machine Learning , W...
```

Figure 4.4: Preprocess the data

4.5 Feature Engineering

Feature engineering is the process of transforming raw data into features that can be used to build models. It involves selecting and creating relevant features from the raw data, as well as preprocessing and to ensure that it is in a suitable format for modeling. Feature engineering is a critical step in our project because most of our features are in text format and can't be understood by our model.

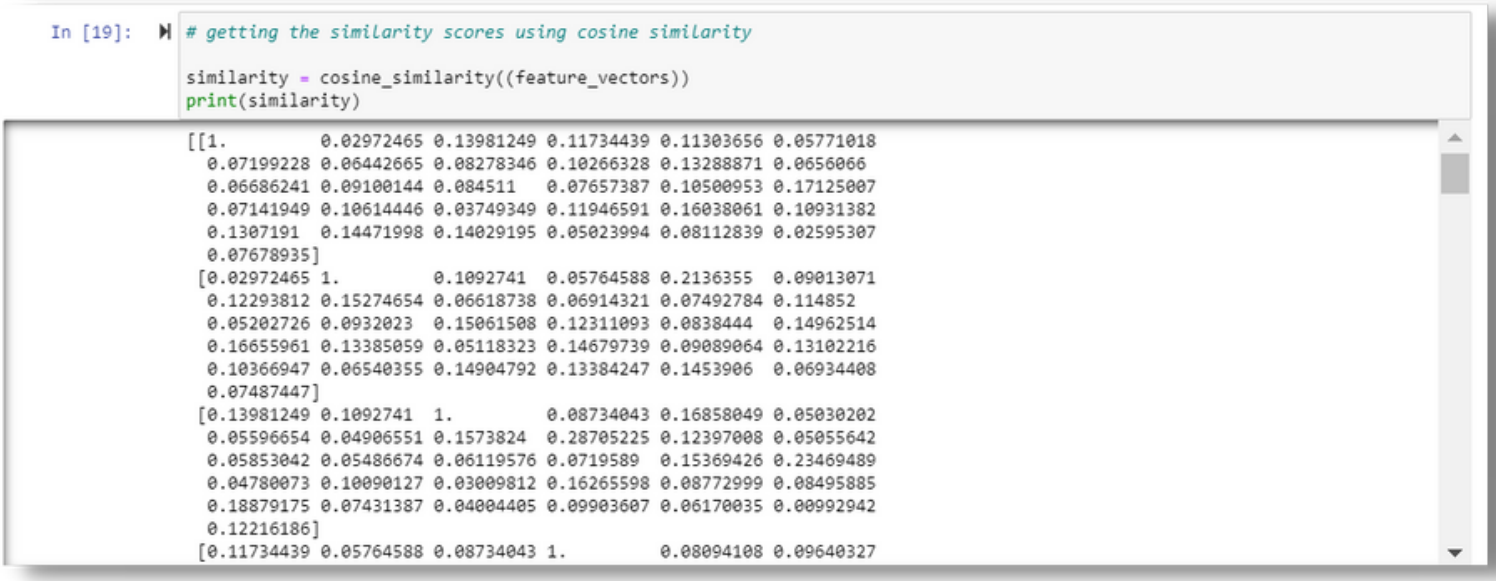


```
In [17]: # converting the text data to feature vectors

         vectorizer = TfidfVectorizer()

In [18]: feature_vectors = vectorizer.fit_transform(combined_features)
         print(feature_vectors)

  (0, 268)      0.09057477834004639
  (0, 281)      0.09891155677166363
  (0, 412)      0.15396404032796343
  (0, 164)      0.08037138616600338
  (0, 227)      0.05888014734608787
  (0, 55)       0.17250421093362367
  (0, 3)        0.09455343332604044
  (0, 26)       0.09735351034408972
  (0, 430)      0.10372923872834466
  (0, 414)      0.10372923872834466
  (0, 208)      0.07203460773438615
  (0, 230)      0.17250421093362367
  (0, 32)       0.13060618776562216
  (0, 320)      0.17250421093362367
  (0, 427)      0.17250421093362367
  (0, 333)      0.15396404032796343
  (0, 147)      0.17250421093362367
  (0, 145)      0.46189212098389026
  (0, 192)      0.3456622908116797
  (0, 222)      0.15396404032796343
  (0, 251)      0.15396404032796343
  (0, 57)       0.17250421093362367
  (0, 149)      0.17250421093362367
  (0, 436)      0.060826214952045966
  (0, 104)      0.17250421093362367
  :     :
  (30, 420)     0.2124554786517023
```

Figure 4.5: Feature Engineering

4.6 Choose a recommendation algorithm

The fifth step is to choose a recommendation algorithm that suits the problem at hand. There are several types of recommendation algorithms, including content-based filtering, collaborative filtering, and hybrid recommendation algorithms. In this case, a collaborative filtering algorithm may be suitable as it relies on the behavior and preferences of other students to make recommendations.



```
In [19]: # getting the similarity scores using cosine similarity

similarity = cosine_similarity((feature_vectors))
print(similarity)
```

```
[[1.         0.02972465 0.13981249 0.11734439 0.11303656 0.05771018
  0.07199228 0.06442665 0.08278346 0.10266328 0.13288871 0.0656066
  0.06686241 0.09100144 0.084511   0.07657387 0.10500953 0.17125007
  0.07141949 0.10614446 0.03749349 0.11946591 0.16038061 0.10931382
  0.1307191  0.14471998 0.14029195 0.05023994 0.08112839 0.02595307
  0.07678935]
 [0.02972465 1.         0.1092741  0.05764588 0.2136355  0.09013071
  0.12293812 0.15274654 0.06618738 0.06914321 0.07492784 0.114852
  0.05202726 0.0932023  0.15061508 0.12311093 0.0838444  0.14962514
  0.16655961 0.13385059 0.05118323 0.14679739 0.09089064 0.13102216
  0.10366947 0.06540355 0.14904792 0.13384247 0.1453906  0.06934408
  0.07487447]
 [0.13981249 0.1092741  1.         0.08734043 0.16858049 0.05030202
  0.05596654 0.04906551 0.1573824  0.28705225 0.12397008 0.05055642
  0.05853042 0.05486674 0.06119576 0.0719589  0.15369426 0.23469489
  0.04780073 0.10090127 0.03009812 0.16265598 0.08772999 0.08495885
  0.18879175 0.07431387 0.04004405 0.09903607 0.06170035 0.00992942
  0.12216186]
 [0.11734439 0.05764588 0.08734043 1.         0.08094108 0.09640327
```

Figure 4.6:Choose a recommendation algorithm

4.7 Model to production

The final step is to deploy the model in a production environment where it can make recommendations to students. This involves integrating the model with the application

```
In [20]:
def recommend(nom):
    index_of_the_name = data[data.name == nom]['id'].values[0]
     #getting a list of similar persons
    ids = list(data["id"])
    if (data[data.id==index_of_the_name]['roommate'].values[0]==True):
        print("you aleardy have a roommate :)")
    else:
        similarity_score = list(similarity[index_of_the_name])
        #print(similarity_score)
        similarity_score_resultat=[]
        for i in range(len(ids)):
              similarity_score_resultat.append((ids[i], similarity_score[i]))

        #print(similarity_score_resultat)
        # sorting the persons based on their similarity score

        sorted_similar_names = sorted(similarity_score_resultat, key = lambda x:x[1], reverse = True)
        for item in sorted_similar_names:
          if item[1] == 1:
              sorted_similar_names.remove(item)

        #print(sorted_similar_names)
        # print the name of similar persons based on the index
        print('We suggest you the best roommates for you : \n')


        i = 1

        for person in sorted_similar_names:
          index = person[0]
          name_from_index = data[data.id==index]['name'].values[0]
          gender_from_index = data[data.id==index]['gender'].values[0]
          if(data[data.id==index]['roommate'].values[0]==False):
            if (gender_from_index == data[data.id==index_of_the_name]['gender'].values[0]):
              if (i<4):
                print(i, '.',name_from_index , gender_from_index )
                i+=1

In [21]:
recommend('Kendrick Trickey')

you aleardy have a roommate :)
```

Figure 4.7 Model to production

4.8 Conclusion

In this chapter we have gone trough each signle step of the data science project's life cycle. In order to know the different technologies we used in detail, the next chapter will be devoted to the Implementation phase of our application.

Chapter 5

Implementation

5.1 Introduction

After having made a design that best meets the needs of our application, we begin the implementation part of the application that we have developed, by exposing the different development tools and languages used during the production of our application as well as the results obtained.

5.2 Tools Used:

5.2.1 HTML CSS:

HTML, HyperText Markup Language, provides structure and meaning to the content by defining this content by principle of markup like, for example, titles, paragraphs or images.. etc.

CSS, or Cascading Style Sheets, is a presentation language created to style the appearance of content, using, for example, fonts or colors.

5.2.2 TypeScript:

TypeScript is a programming language that is a superset of JavaScript, meaning that it is a variant of JavaScript that includes additional features. It was developed and is maintained by Microsoft.
One of the main features of TypeScript is its static typing, which allows developers to specify the data types of variables and function argu- ments at compile time, rather than at runtime as in JavaScript.
This can help to catch errors and improve the maintainability of code. TypeScript also includes features such as interfaces, classes, and dec- orators, which can make it easier to write object-oriented code.
It is often used in the development of large-scale applications and is supported by a number of frameworks and tools, including Angular and React.

5.2.3 Angular:

Angular is a frontend web development framework used for building single-page applications (SPAs). It is based on TypeScript, a superset of JavaScript, and follows the Model-View-Controller (MVC) architectural pattern. Angular provides a set of tools and features for building interactive and dynamic user interfaces, including templates, components, and dependency injection. It also has a built-in support for routing, form validation, and handling HTTP requests. Angular is maintained by Google and is widely used in the development of web applications.

5.2.4 Spring Boot:

Spring Boot Spring Boot is a popular Java-based framework used to build production-grade web applications and services. It is built on top of the Spring Framework and takes an opinionated view of the Spring ecosystem, meaning that it includes pre-configured versions of tools and libraries that work well together.

5.2.5 Python :

Python is particularly popular for data analysis and Machine Learning, but also for backend web development. This language is also used to scrape web data and develop productivity tools. Python comes with a built-in package called json to encode and decode JSON data.

The process of encoding JSON is usually called serialization. This term designates the transformation of data into a series of bytes (therefore in series) to be stored or transmitted on a network.

5.2.6 Flask :

Figure 5.1: Logo - Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.
Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

5.3 Development environment

5.3.1 VisualStudioCode:

Figure 5.2: Logo - VS Code

Visual Studio Code is an extensible code editor developed by Microsoft for Windows, Linux, and macOS2.
Features include debugging support, syntax highlighting, smart code completion, snippets, code refactoring, and Git built-in. Users can change the theme, hotkeys, preferences, and install extensions that add additional functionality.

5.3.2 Jupyter Notebook :

Figure 5.3: Logo - Jupyter Notebook

Jupyter Notebook (formerly IPython Notebooks) is a web-based, interactive programming environment for authoring Jupyter Notebook documents. The term "notebook" can refer to many different, context-appropriate entities, such as Jupyter web application, Jupyter Python web server, or Jupyter document format.

5.3.3 XAMPP:

Figure 5.4: Logo - XAMPP

XAMPP is a free, open-source software package that provides a complete web server environment for local development and testing purposes. It includes Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

5.3.4 Versionandcollaborationmanagement

Figure 4.5: Logo - Git

Git is a version control system that allows developers to track changes to source code and collaborate with other developers on software projects. It allows developers to work on multiple versions of a codebase simultaneously, without fear of overwriting each other's changes.

Figure 5.6: Logo - Github

GitHub is a web-based platform that allows developers to store and manage their Git repositories online. It provides a range of tools and features for collaboration, code review, and version control. It is a popular choice for hosting open-source projects and has become an essential tool for many developers.

5.4 Web Platform Interfaces

This part lists the presentation of the application scenarios of the «Claim Book» web platform.
We will present in the following are screenshots of the main Pages for a User WorkFlow.

5.4.1 Homepage

This interface allows the user to access the different pages of the platform.

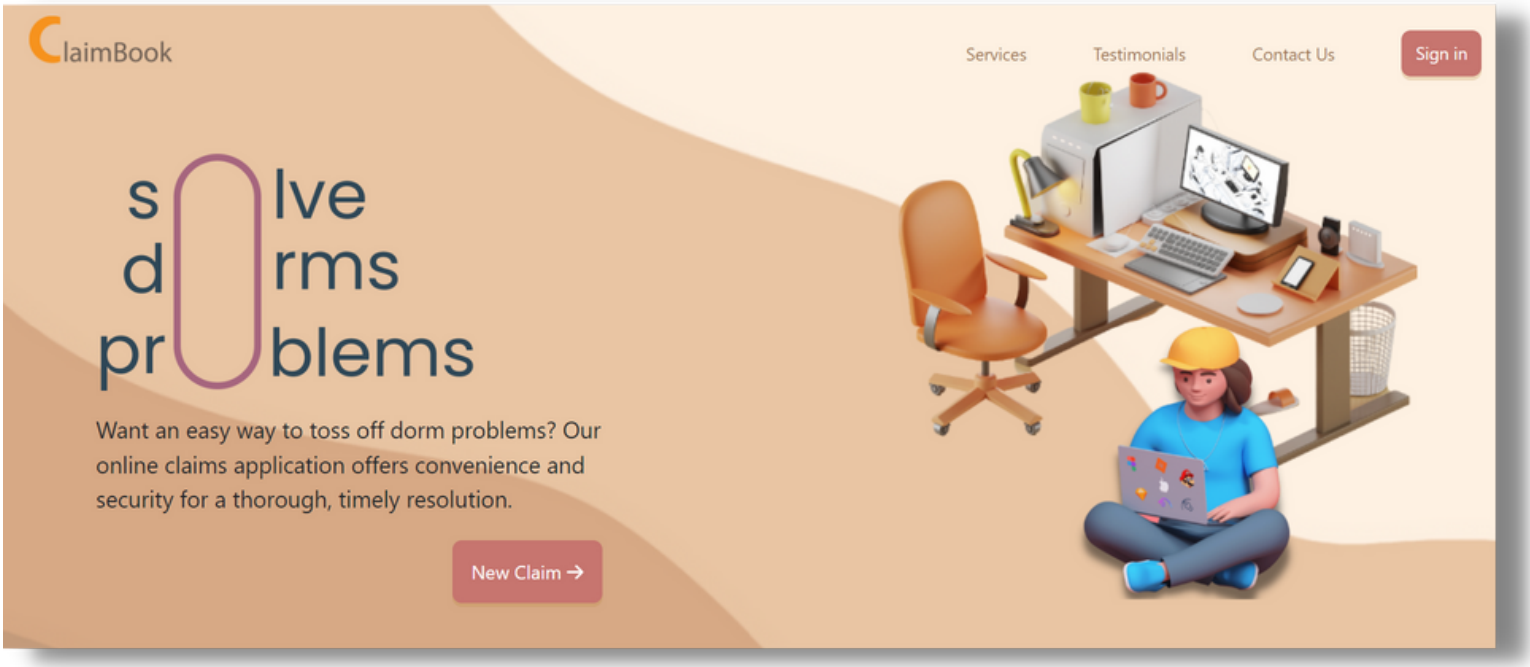


Figure 5.8: Web Platfrom - Landing Page

5.4.2 Student login page

This interface allows the studentto connect with his email address and password.

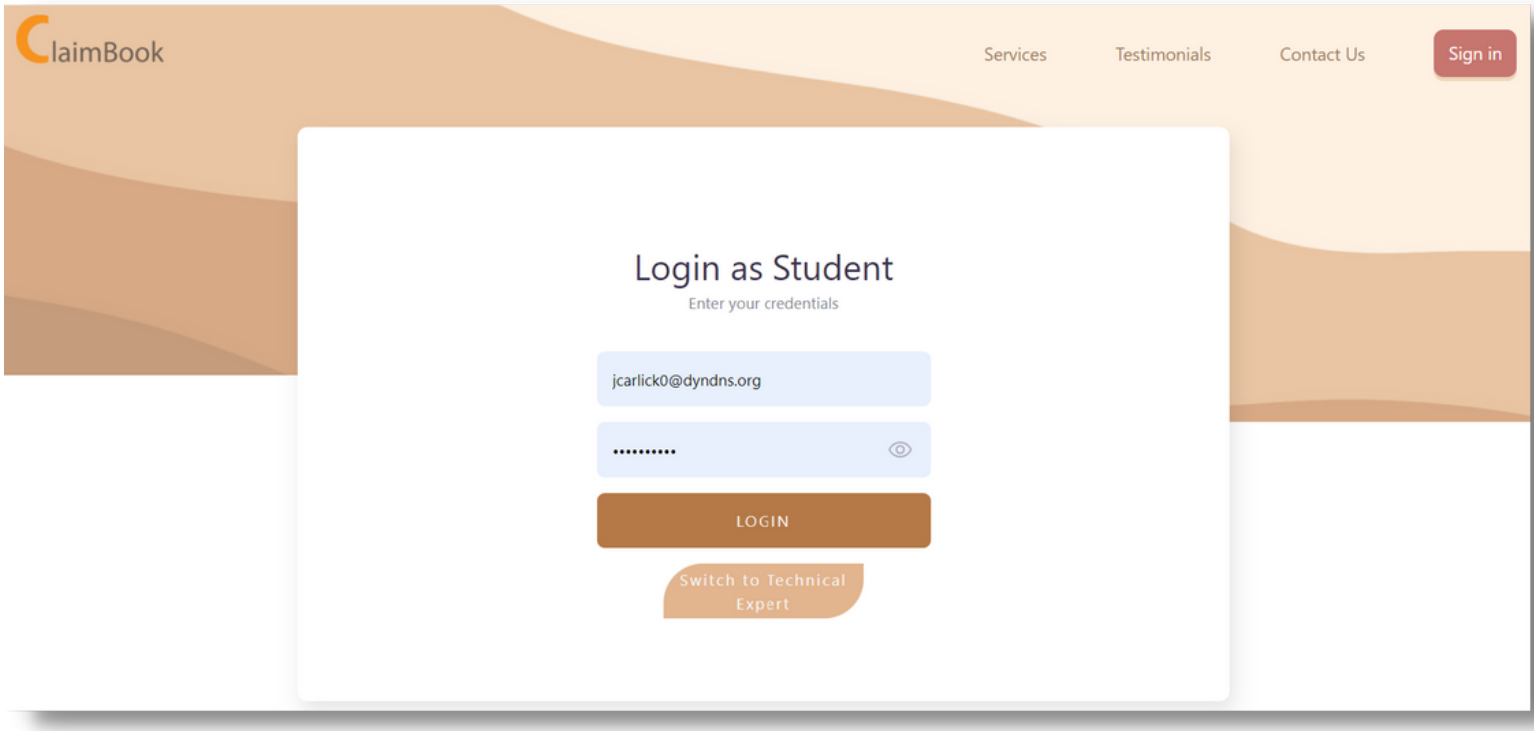


Figure 5.9: Web Platfrom - log Page

5.4.3 Student Profile

This interface allows the student to consult his profile and he can get help to find roommate if he doesn't have one.

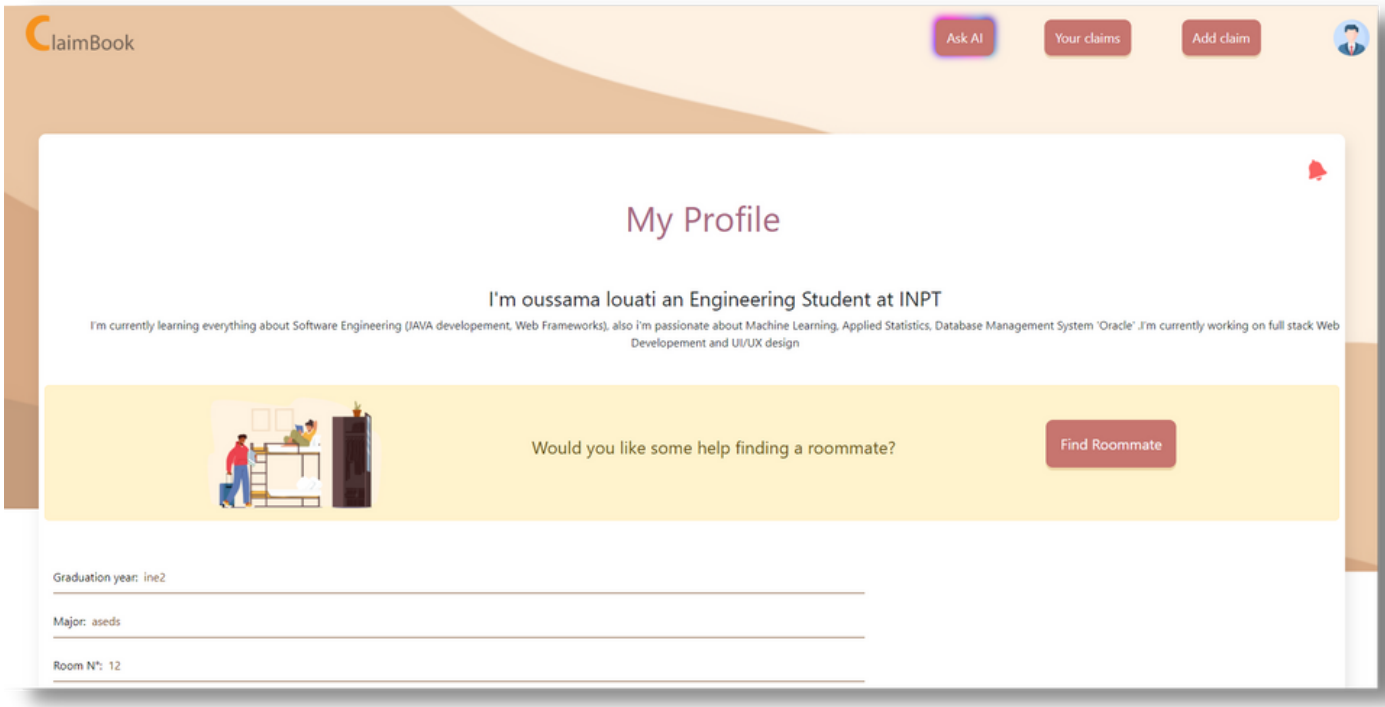


Figure 5.10: Web Platfrom - Student Profile Page

5.4.4 Recommendations Interface

When he click o find roommate he get three recommended roommates based on his profile

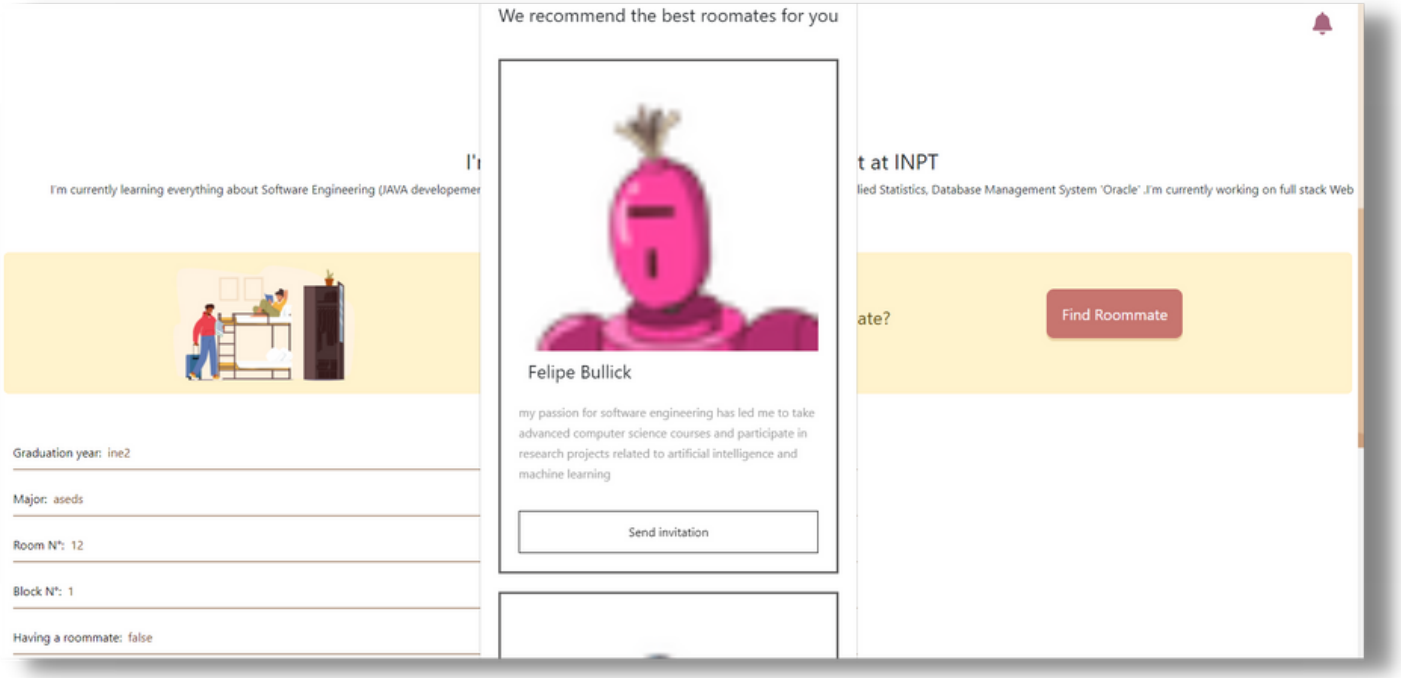


Figure 5.11: Web Platfrom - Main Interface

Once the user has choose a roommate, an invitation that contain user information will be sent to that roommate.

5.4.5 get invitation notification

this interface show the invitation to user , it contain the invitation owner information , if the user accept or reject the invitation a notification will be sent to the invitation owner .

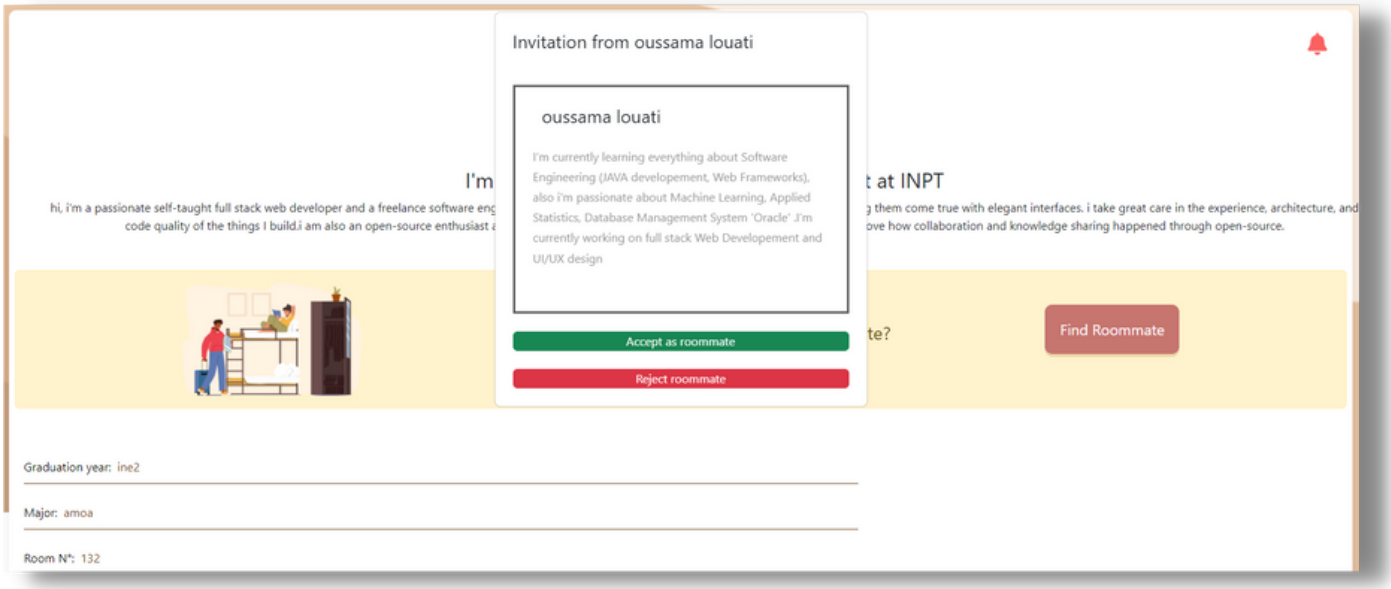


Figure 5.12: Web Platfrom - get invitation interface

5.4.6 accepted invitation notification

this interface show the invitation has been accepted .

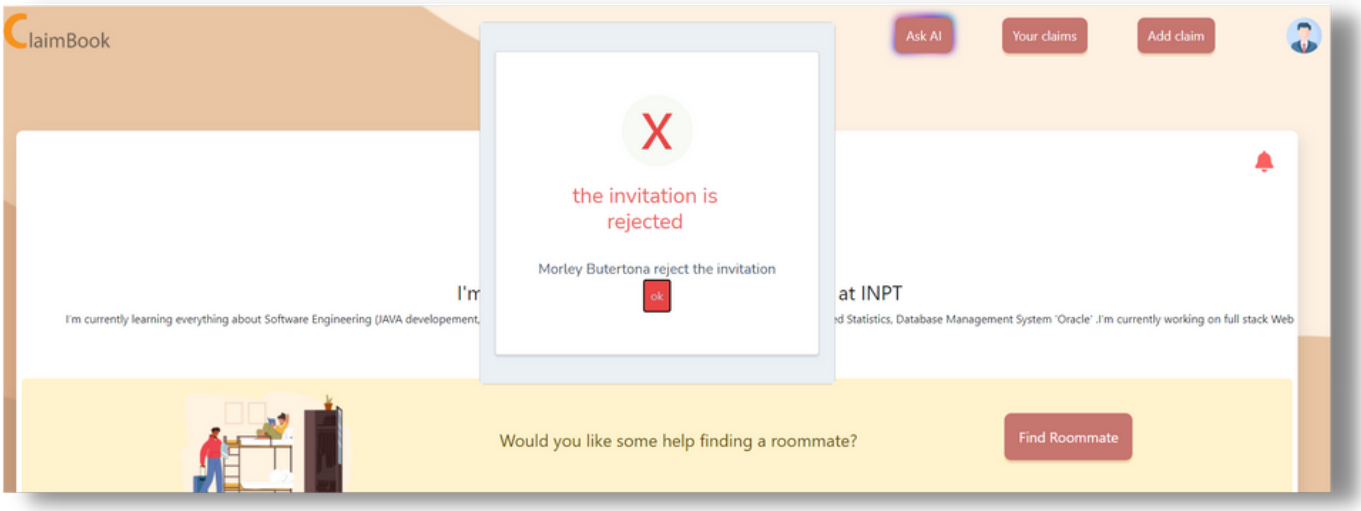


Figure 5.13: Web Platfrom - accepted invitation interface

5.4.7 rejected invitation notification

this interface show the invitation has been rejected.

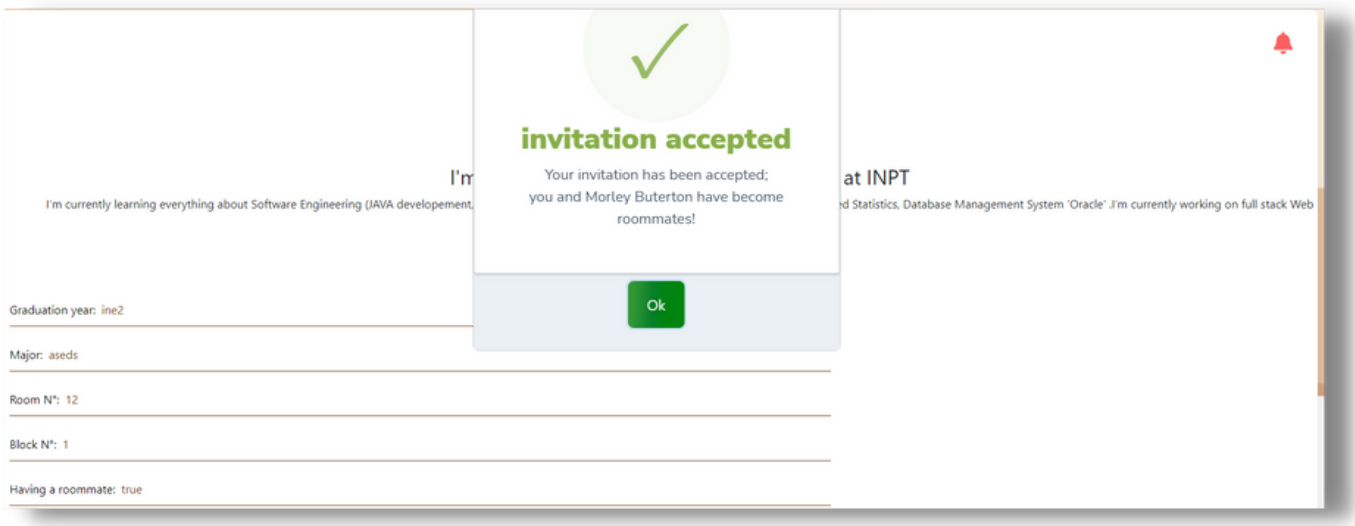


Figure 5.14: Web Platfrom - rejected invitation interface

5.4.8 claim lists Interface

This interface allows the student to consult the list of his claims with the status of each one .

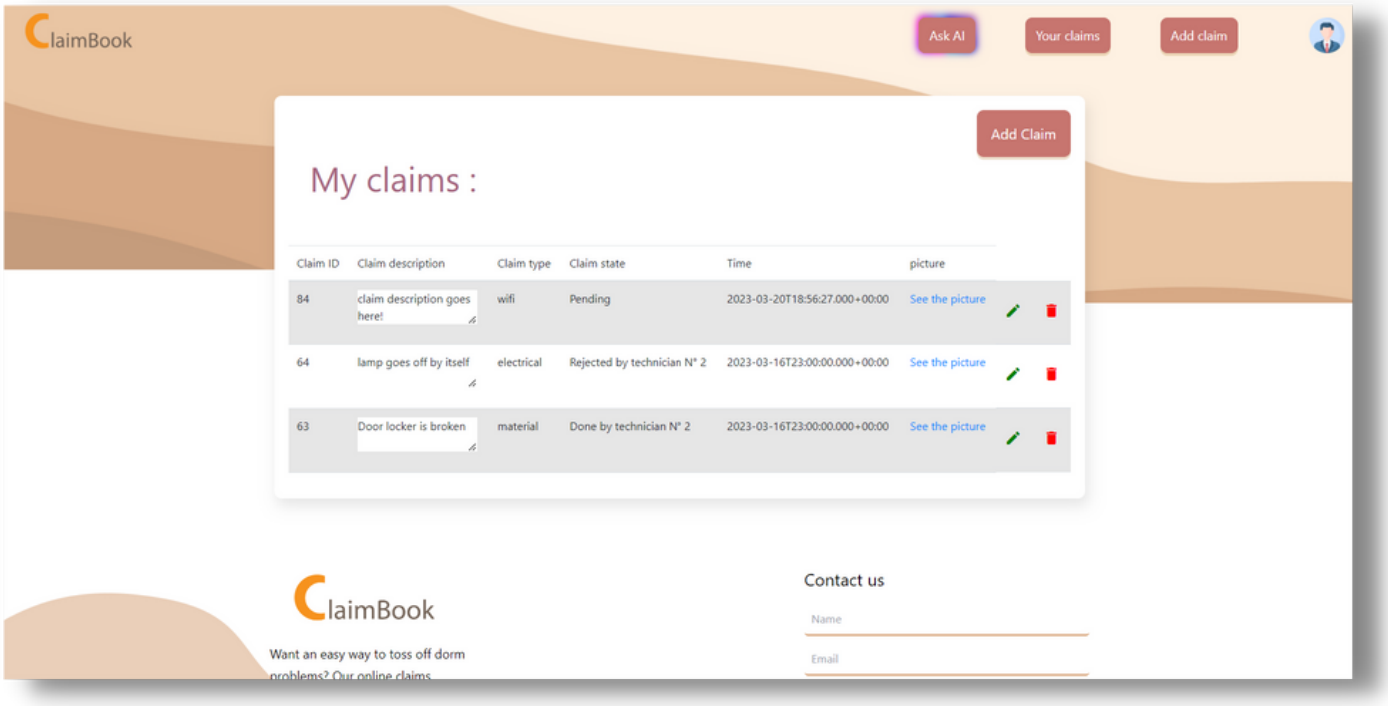


Figure 5.15: Web Platfrom - claim list interface

5.4.9 add claim Interface

This interface allows the student to add new claim . he can specify the type of the problem and add a photo and the description of the claim .

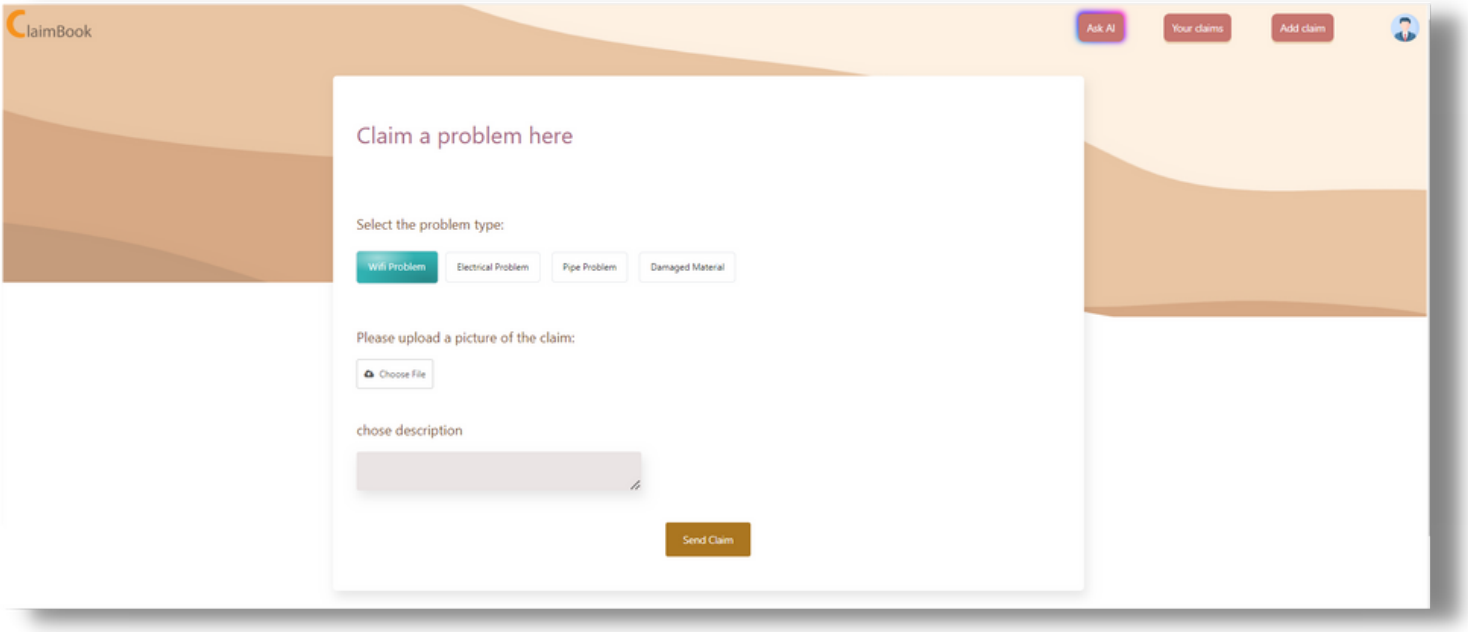


Figure 5.16: Web Platfrom - add claim interface

5.4.10 edit claim Interface

This interface allows the student to edit a claim .

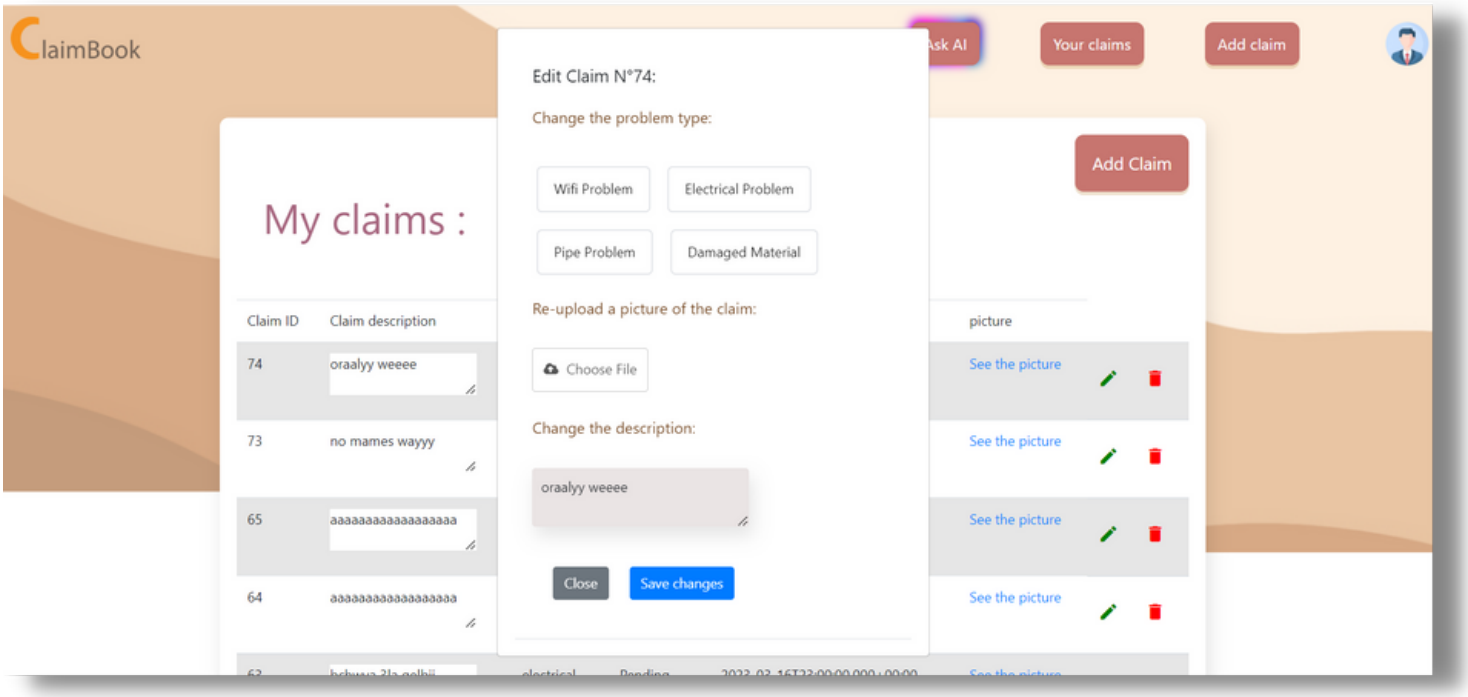


Figure 5.17: Web Platfrom - edit claim interface

5.4.11 delete claim Interface

This icon delete allows the student to delete a claim .

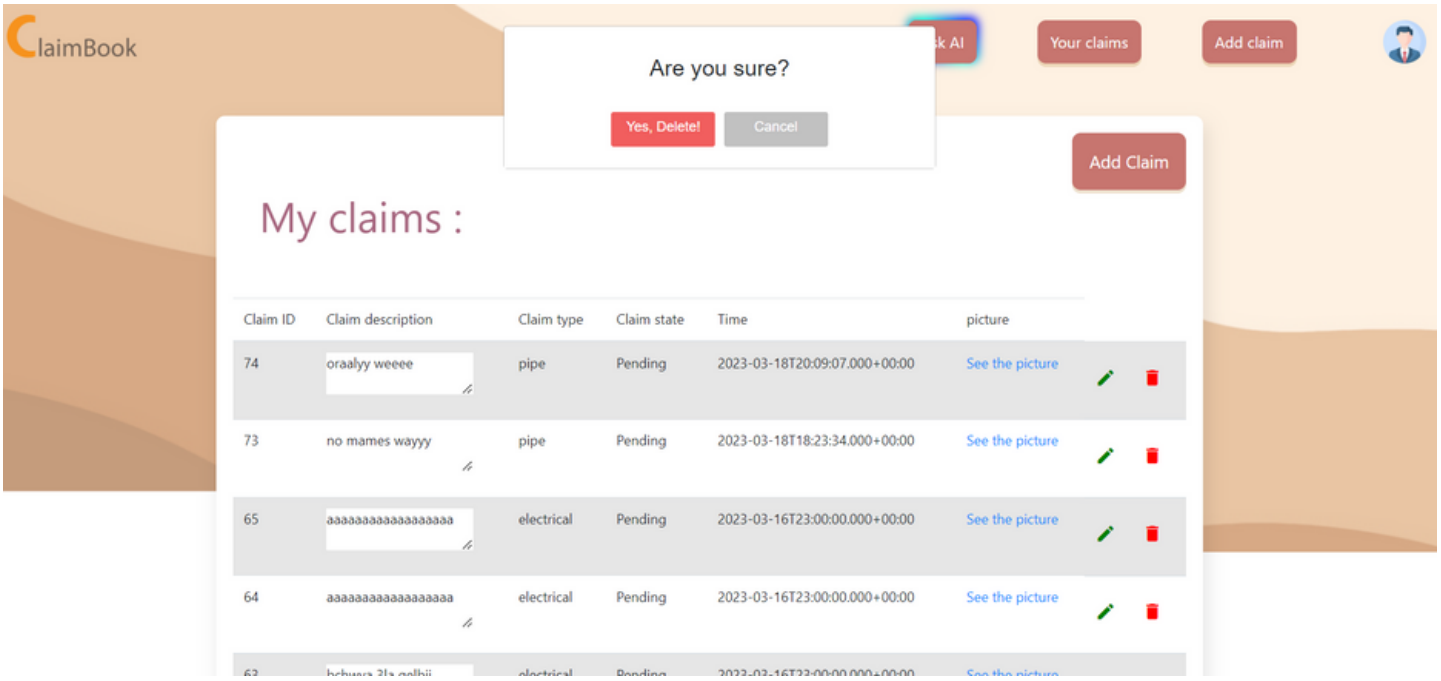
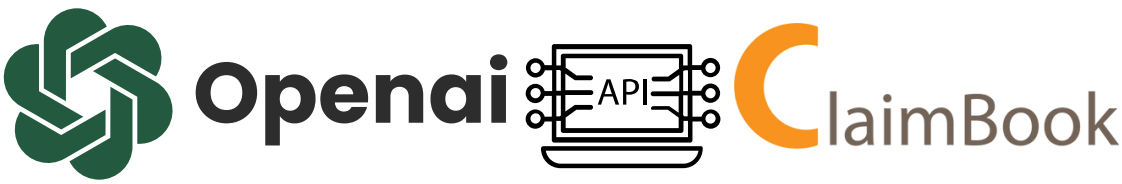


Figure 5.18: Web Platfrom - delete claim interface

5.4.12 ask ai Interface

This interface contains an input field where the user can paste his question, a button beside the seach bar which directs make the request to the server.

After Clicking the search button the user will be prompted with the waiting interface.



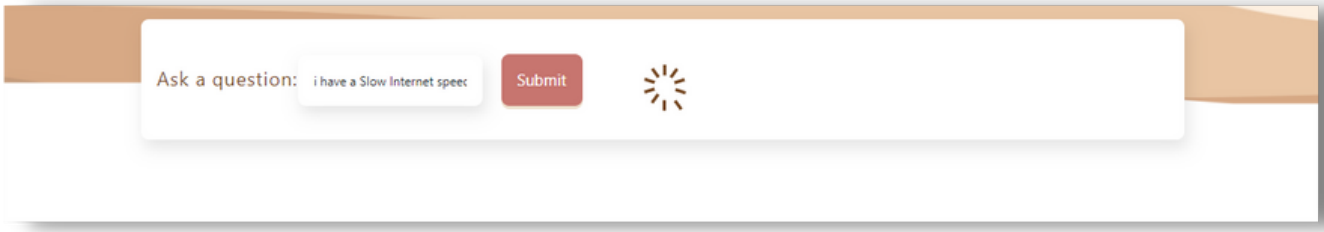


Figure 5.19: Web Platfrom - Waiting Animation

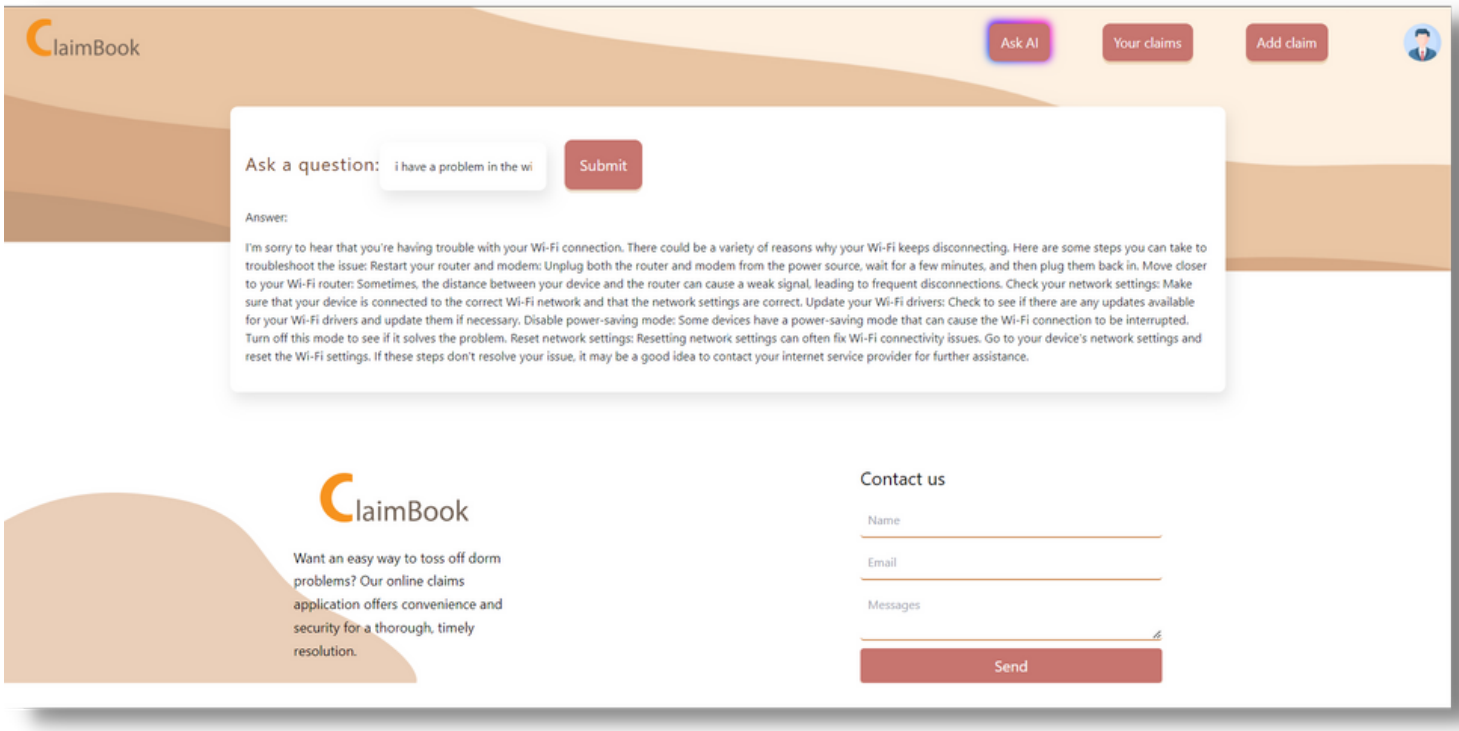


Figure 5.20: Web Platfrom - Result

Once the server has processed the request, our frontend will receive the response and display the result of the response

5.4.13 Technician login page

This interface allows the technician to connect with his email address and password.

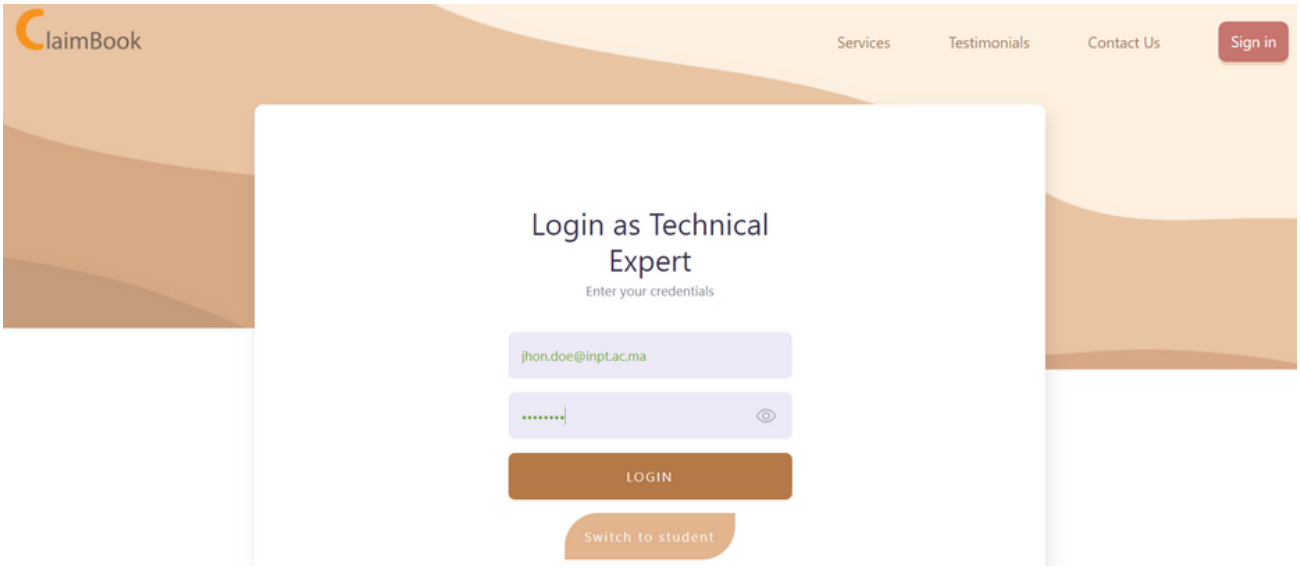


Figure 5.21: Web Platfrom - technician login interface

5.4.14 Claim list page

This interface allows the technician to consult all student's claims.

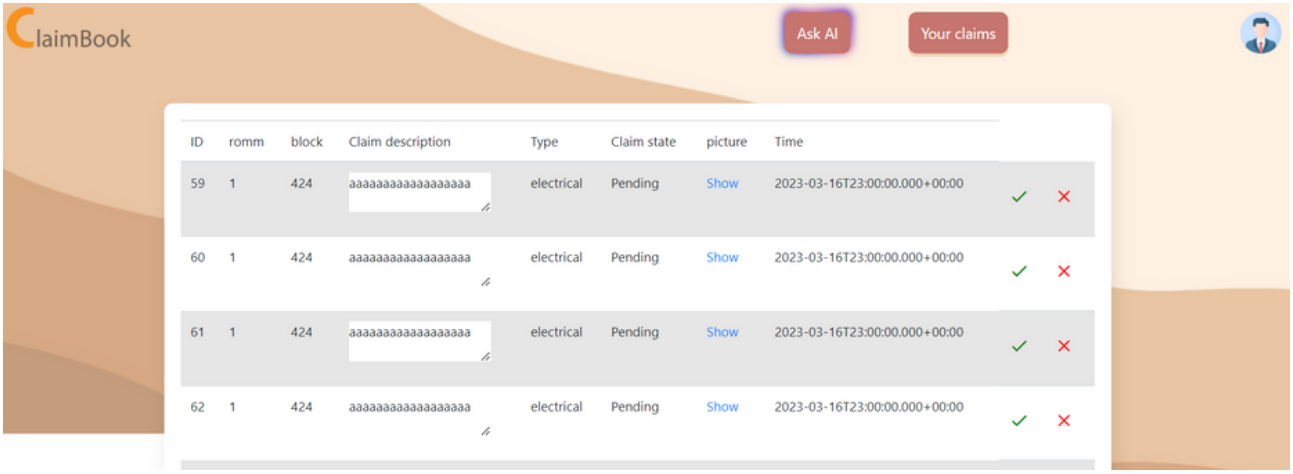


Figure 5.22: Web Platfrom - technician claim list interface

5.4.15 resolve claim page

This interface allows the technician to confirm that he solve the dorm problem.

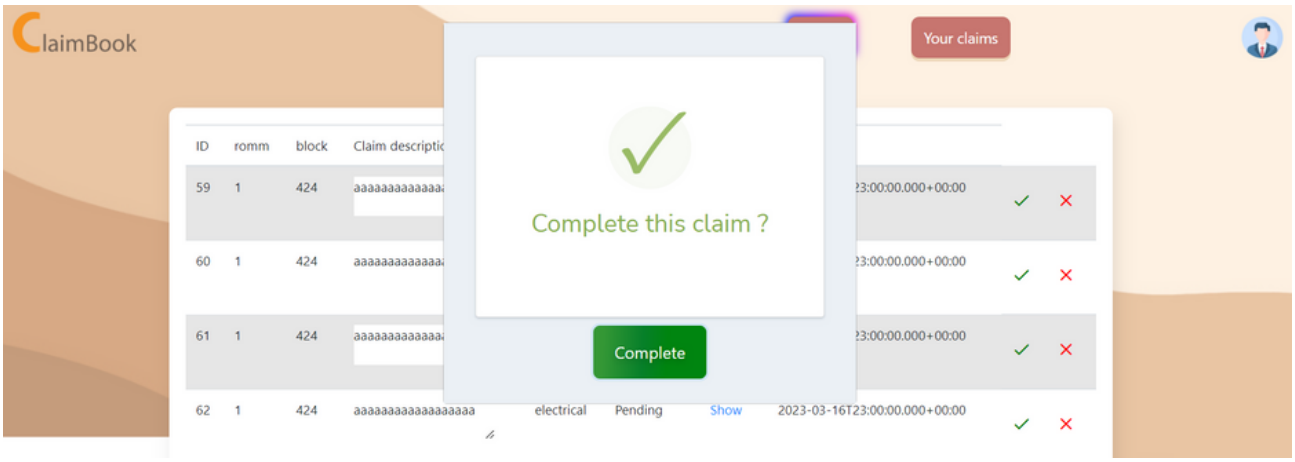


Figure 5.23: Web Platfrom - resolve claim interface

5.4.16 reject claim page

This interface allows the technician to reject a claim if he can't handle the problem or the claim does not contain enough detail.

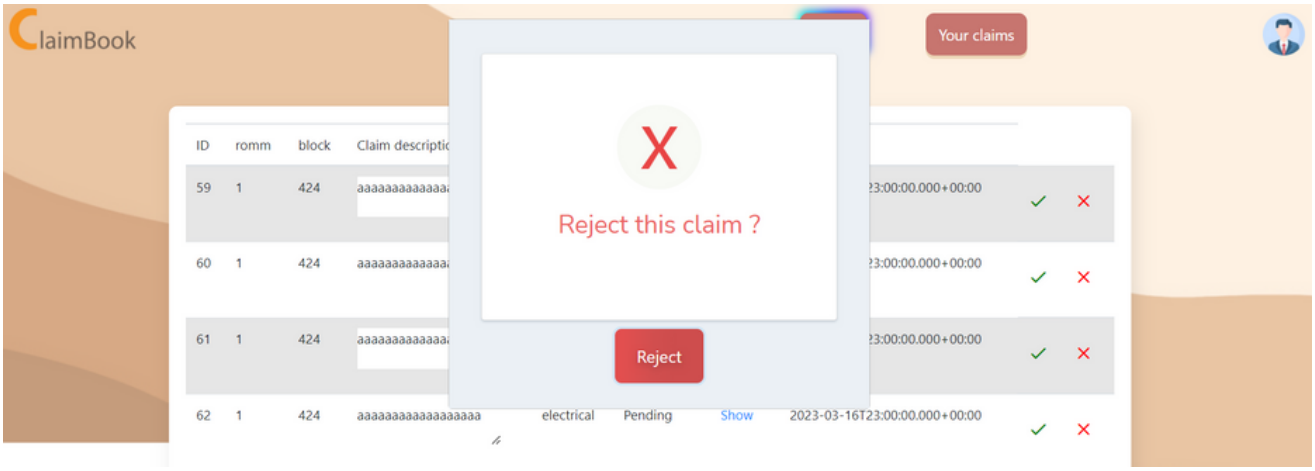


Figure 5.24: Web Platfrom - reject claim interface

General Conclusion

The work we did consisted of creating a web application that digitize claim management and roommate choosing . We did a little analysis of user situations. we also provided the general architecture of our application, including the workflow of user interaction with the application and the details of our Machine Learning model.

This project allowed us to put into practice the theoretical concepts that we learned throughout the year. The realization of this project was also an opportunity to obtain a clear vision of the progress of a professional project and discover the various difficulties and obstacles that can be encountered. It also allowed us to learn the importance of teamwork, which appears essential for the realization of a web application.

Bibliography

[1] Angular Documentation. https://angular.io/tutorial.
[2] Spring Boot Documentation. https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/.
[3] Angular Documentation. https://angular.io/tutorial.
[4] Flask Documentation. https://flask.palletsprojects.com/en/2.2.x/
[5] OpenAI Documentation https://platform.openai.com/docs/introduction
[6] StackOverflow. http://www.stackoverflow.com.