Software Requirements Specification (SRS) Document

Health First Deakin

19th/JAN/2023

**Team Members**

ALI ADIL ABDULRAZZAQ AL-KINANI SHAMAIL HAIDER

HARSH PATEL YARU XU

Table of Contents

1. [Introduction 4](#_bookmark0)
   1. [Abstract 4](#_bookmark1)
   2. [Aim and objective 4](#_bookmark2)
   3. [Motivation 4](#_bookmark3)
   4. [Target Audience 4](#_bookmark4)
   5. [Scope 4](#_bookmark5)
2. [Overview 5](#_bookmark6)
   1. [System Overview 5](#_bookmark7)
   2. [Proposed System 5](#_bookmark8)
   3. [Benefits Proposed System 5](#_bookmark9)
   4. [Limitations Proposed System 6](#_bookmark10)
   5. [Future Trends 6](#_bookmark11)
3. [Business Features 7](#_bookmark12)
   1. [What 7](#_bookmark13)
   2. [Who 7](#_bookmark14)
   3. [How? 7](#_bookmark15)
4. [Analysis and Design Methodology 8](#_bookmark16)
   1. [Class Diagram 8](#_bookmark17)
   2. [Sequence Diagram 9](#_bookmark18)
   3. [Activity Diagram 10](#_bookmark19)
   4. [MVP 10](#_bookmark20)
5. [Modules Description 11](#_bookmark21)
   1. [Patients 11](#_bookmark22)
   2. [Doctors 12](#_bookmark23)
   3. [Clinic 13](#_bookmark24)
6. [Prototypes 15](#_bookmark25)

[Login 15](#_bookmark26)

[Register 15](#_bookmark27)

[Patients Profile 16](#_bookmark28)

[New Patients 16](#_bookmark29)

[User Home 17](#_bookmark30)

[User Profile 17](#_bookmark31)

[Recipes Profile 18](#_bookmark32)

[Details of Recipes 18](#_bookmark33)

[User Recipes 19](#_bookmark34)

[Clinic information Register 19](#_bookmark35)

[Clinic Profile 20](#_bookmark36)

[Clinic Booking 20](#_bookmark37)

[Upcoming Appointments 21](#_bookmark38)

[Doctor Profile 21](#_bookmark39)

1. [Non-Functionality 22](#_bookmark40)
   1. [Security 22](#_bookmark41)
   2. [Binary Compatibility 22](#_bookmark42)
   3. [Reliability 22](#_bookmark43)
   4. [Maintainability 22](#_bookmark44)
   5. [Portability 22](#_bookmark45)
   6. [Extensibility 22](#_bookmark46)
   7. [Reusability 22](#_bookmark47)
   8. [Application Affinity/Compatibility 22](#_bookmark48)
   9. [Serviceability 23](#_bookmark49)
2. [Project scenario 23](#_bookmark50)

# 1. Introduction

* 1. **Abstract**

A Diet app that helps an individual to self-regulate their eating habits and minimize their risk of health disease.

* 1. **Aim and objective.**

The main objective of system is to create a diet app for people that provides users several services to users according to their height and weight with the proper consultation with the doctors and help them to achieve desire results.

* 1. **Motivation**

We analysed several data and statistics for the internet and found that Australia ranks 9th position in obesity and people from young age starts facing so much problem like heart attack, cholesterol, diabetes, and many other health related issues which is because they are engaged with poor diet and irrational eating habits. So, to minimize the health risk and help people to gain knowledge and get proper guidance about what to eat in what quantity We have come up with diet app that helps users to consult with doctor and get them proper suggestions related to health and their diets habits.

* 1. **Target Audience**

Today many people are unfit as they lack knowledge regarding diet, so our application is for everyone who is keen into fitness and has a goal to live a happy and healthy life.

* 1. **Scope**

The scope of the web application is to reduce the death rate and increase awareness among the people to be fit and live a healthy lifestyle also provide a detail overview of their eating habits which will be corrected by having right consultation from the doctors.

# 2 Overview

* 1. **System Overview**

Deakin first health web application has three different modules the heart of the system the most important module for a system is its users so when users tries to login into system with their given Username and password they can have a look for doctors who can suggest them diet they can check for various diet recipes consult doctors the other important feature is doctor from where users can get proper guidance regarding their diet users can book appointments from the system and doctors can give them their availability. doctors can even recommend certain basic stretching exercise or vitamins to both the functions together third module clinic is made that will create an interaction point between user and doctor moreover will take care of various management related work for doctor as well as patients.

* 1. **Proposed System**

**Patients Details**: The new proposed system stores and maintains all the Patients details.

**Doctor Details:** The new proposed system stores and maintains all the Doctor details.

**Clinic Details:** The new proposed system stores and maintains all the Clinic details.

**Speed**: The new proposed system is very fast with 100% accuracy and saves time

**Efficiency**: The new proposed systems complete the connects user to doctor in less time.

**Reduces redundancy**: The most important benefit of this system is that it reduces the redundancy of data within the data.

**Interactive:** The system is more interactive as it connects all modules, and they can book appointments and check their history or profile and edit as per there convenience

* 1. **Benefits Proposed System**
     + The system works on latest technologies so if there are further changes it can be easily manageable.
     + User can interact with doctor and get proper guidance regarding their diet can also get certain recommendations.
     + Secure authentication system each user of the system has their unique login ID and password.
     + Online booking for patients can reduce the time and Patents can get flexibility.
     + A Git repository is created so developers can work together and integrate their work moreover help for future versions.
  2. **Limitations Proposed System**
     + It is a small project so there will be no maintenance and will have less updates.
     + System does not have blood related details, so patients need to get the report from pathology labs.
     + Internet connection is required to get the profile related data.
  3. **Future Trends**
     + Pathology will be a module so that patients can view and submit their blood samples with the applications.
     + A recommendation related to diet and tracking related to calories can be tracked by the patients.
     + A chat board will be provided so users and doctor can interact virtually, and a pdf of report will be generated.
     + User can post their recipes and can make them verified with doctors so that other patients can also grab benefits.
     + Doctors will be provided rating according to their patient count and their patient’s feedback.

# 3 Business Features

* 1. **What**

## Value Proposition

* + - **Diet details** –Patients will be able to view the details regarding his/her diet and deficiencies.
    - **Profile details** – will display Profile details of the Patients. Patients view the feedback generated by the doctor on his profile with their diet suggestions.
  1. **Who**

## Modules

* + - Clinic
    - Patient
    - Doctor

## Customer Relationships

* + - On registration Patients are provided email address, we will keep them updated through emails.
    - Patients can email us on help email to get a general help or for any queries which system does not provides.
  1. **How?**

## Key Features

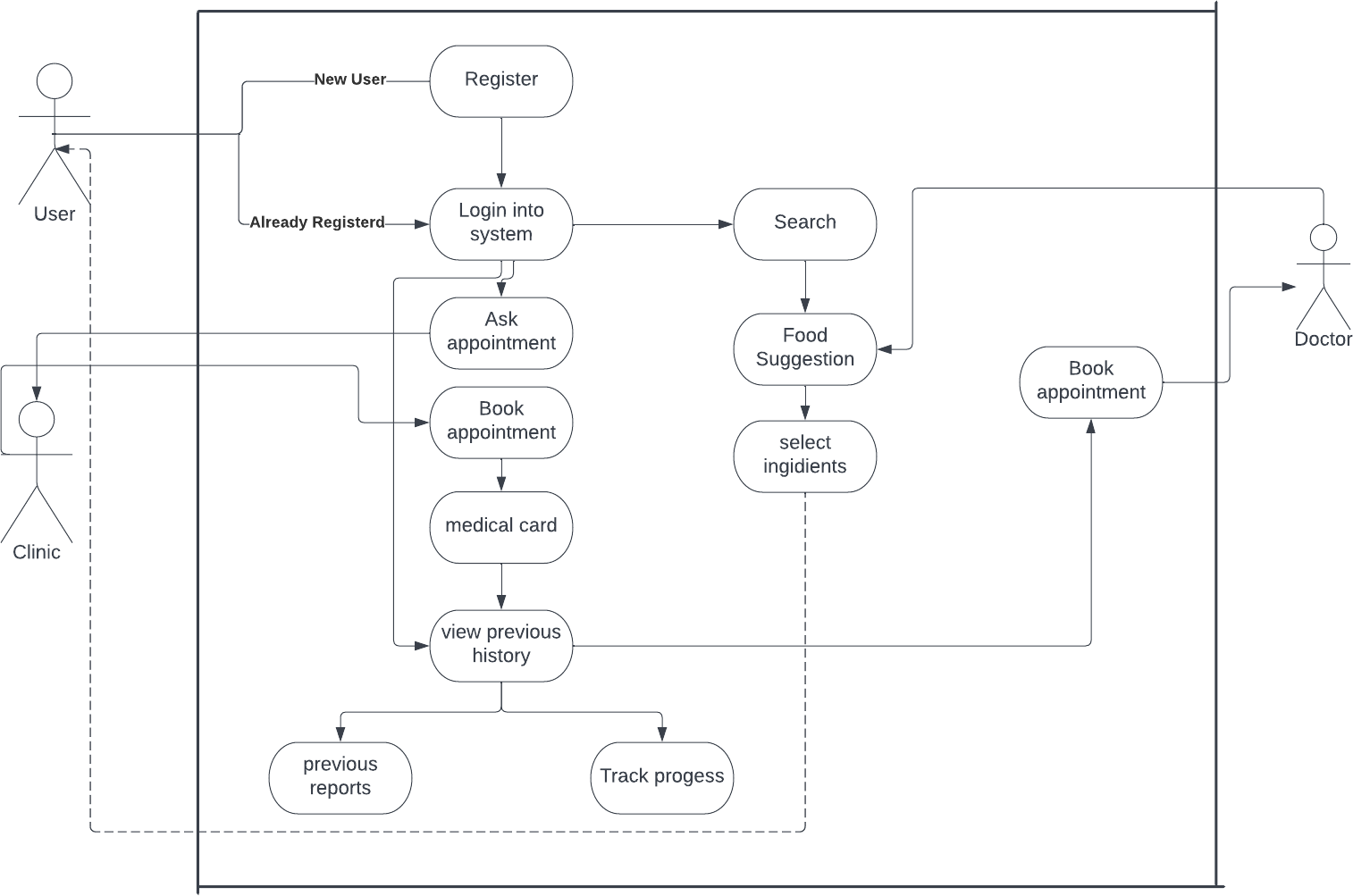
* + - Booking appointments for user and doctor
    - User, Doctor, and Clinic have certain CRUD operations.
    - Various range of diet options provided by experience doctors.

## Key Resources

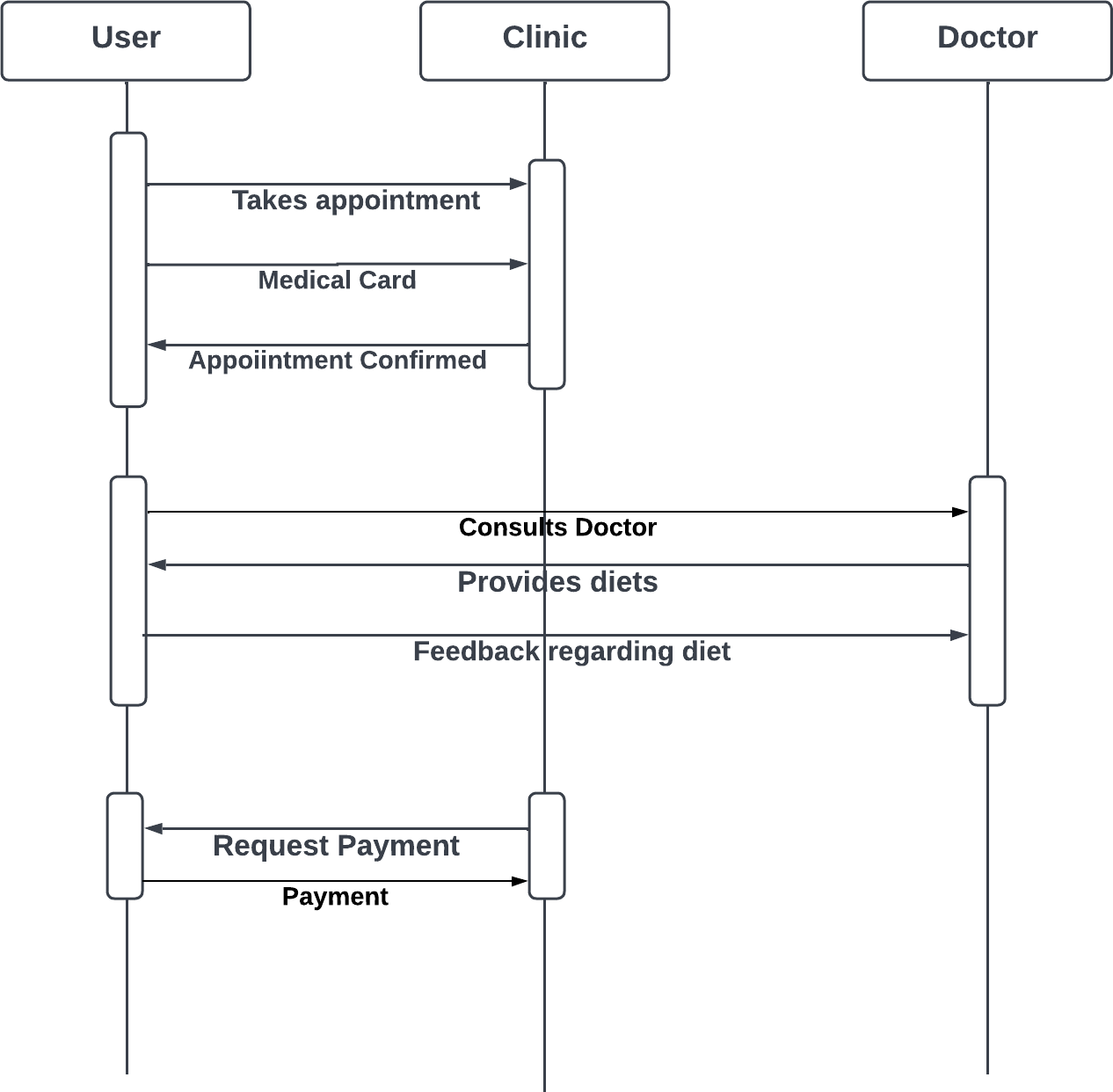
* + - The Data generated by the system will be more helpful to generate revenue and keep a log of patients.
    - Patients are the one of the biggest parts of the system.

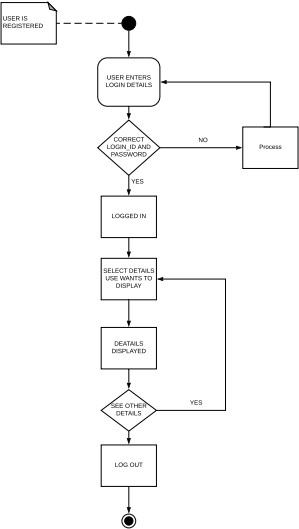
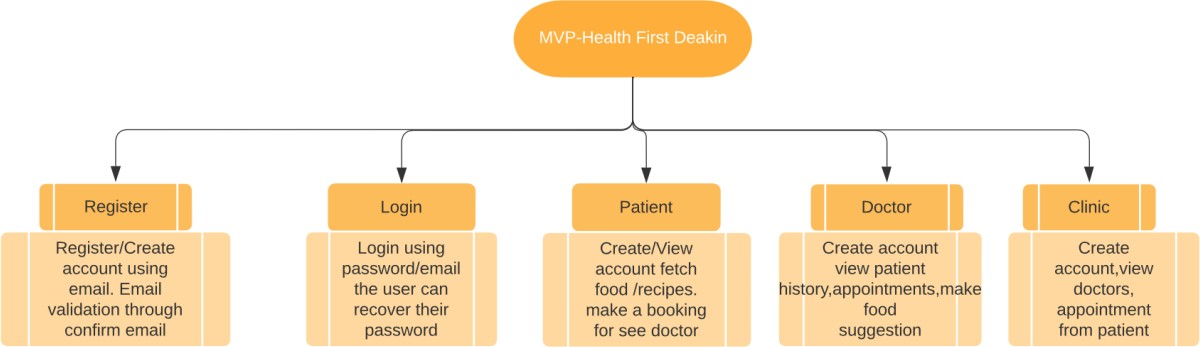
# 4 Analysis and Design Methodology

* 1. **Class Diagram**



* 1. **Sequence Diagram**



* 1. **Activity Diagram**
  2. **MVP**

# 5 Modules Description

* 1. **Patients**

## User Stories

Being a patient, I could do a search for diet food/packages by its name, and I could also view its nutritional value.

Use case - Search food products.

**Primary actor:** User/Patient

## Secondary actor: None

**Description:** The use case, Patient can search food products using some keywords.

**Precondition:** User should have access to the web application at the application URL.

## Basic flow

* + - 1. User chooses the option to search for a product.
      2. Health Frist Deakin provides option to enter search keywords.
      3. User enters the keywords.
      4. Health Frist Deakin displays the list of products with those keywords in their names and the list of nutrients and their quantities in them.
      5. Use case ends successfully.

## User Login

**Primary actor:** Login user

**Description:** In this use case, a registered user can log in to Health First Deakin system.

**Precondition:** The registered user should have access to Health First Deakin on the web.

## Basic flow

* + - 1. Registered user chooses the option to log in.
      2. Health First Deakin displays the options to enter username and password.
      3. Registered user enters username and password.
      4. Health First Deakin checks the username and password, and if correct, shows new features for registered users.
      5. Use case ends successfully.

## View Diet

**Primary actor:** Patients

**Description:** In this use case, a registered Patient can view their diet log.

**Precondition:** The registered user should be logged in.

## Basic flow

* + - 1. Registered Patient chooses the option to view diet log.
      2. Health First Deakin displays the list of meals entered so far starting with most recent.
      3. Health First Deakin gives options to filter the list for a given date range.
      4. Registered user enters the required inputs.
      5. Health First Deakin filters the list for a given date range.
      6. Use case ends successfully.
  1. **Doctors**

## Doctor Login Primary actor: Doctor

**Description:** In this use case, a registered Doctor can log in to Health First

Deakin system.

**Precondition:** The registered Doctor should have access to Health First Deakin on the web.

## Basic flow

* + - 1. Registered Doctor chooses the option to log in.
      2. Health First Deakin displays the options to enter username and password.
      3. Registered user enters username and password.
      4. Health First Deakin checks the username and password, and if correct, shows new features for registered users.
      5. Use case Login successfully.

## Doctor Profile

**Primary actor:** Doctor

**Description:** In this use case, a registered Doctor can View, Update, and delete to Health First Deakin system.

**Precondition:** The Doctor should access and customise his/ her details to Health First Deakin on the web.

## Basic flow

* + - 1. Doctor chooses the option to Profile.
      2. Health First Deakin displays various options to doctor profile.
      3. Update Doctor can update their personal and professional details.
      4. Delete Doctor can even delete some of the detail’s personal details.
      5. Doctor Details should be Update/delete successful.

## Doctor Patient List Primary actor: Doctor

**Description:** In this use case, a doctor can view their list of patients and their cases into Health First Deakin system.

**Precondition:** The Doctor should have access to patients details of Health First Deakin on the web.

## Basic flow

* + - 1. Doctors can visit patient’s options from navigation bar.
      2. A List of patients with their current summery and details will be displayed.
      3. Doctors can view history of patients and give them certain update regarding their health.
      4. User case ends successfully.
  1. **Clinic**

## Clinic Login Primary actor: Clinic

**Description:** In this use case, a registered Clinic can log in to Health First

Deakin system.

**Precondition:** The registered Clinic should have access to Health First Deakin on the web.

## Basic flow

* + - 1. Registered Clinic chooses the option to log in.
      2. Health First Deakin displays the options to enter username and password.
      3. Registered user enters username and password.
      4. Health First Deakin checks the username and password, and if correct, shows new features for registered users.
      5. Use case Login successfully.

## Clinic Profile Primary actor: Clinic

**Description:** In this use case, a registered Clinic can Update or delete their

profile details with Health First Deakin system.

**Precondition:** The registered Clinic should have access to Health First Deakin on the web.

## Basic flow

* + - 1. Clinic reception chooses the option to Profile.
      2. Health First Deakin displays various options to Clinic profile.
      3. Update Clinic reception can update their personal and professional details.
      4. Delete Clinic reception can even delete some of the detail’s personal details.
      5. Clinic Details should be Update/delete successful.

## Clinic Booking Primary actor: Clinic

**Description:** The use case, a registered Clinic will be able to check patient’s

appointments to Health First Deakin system.

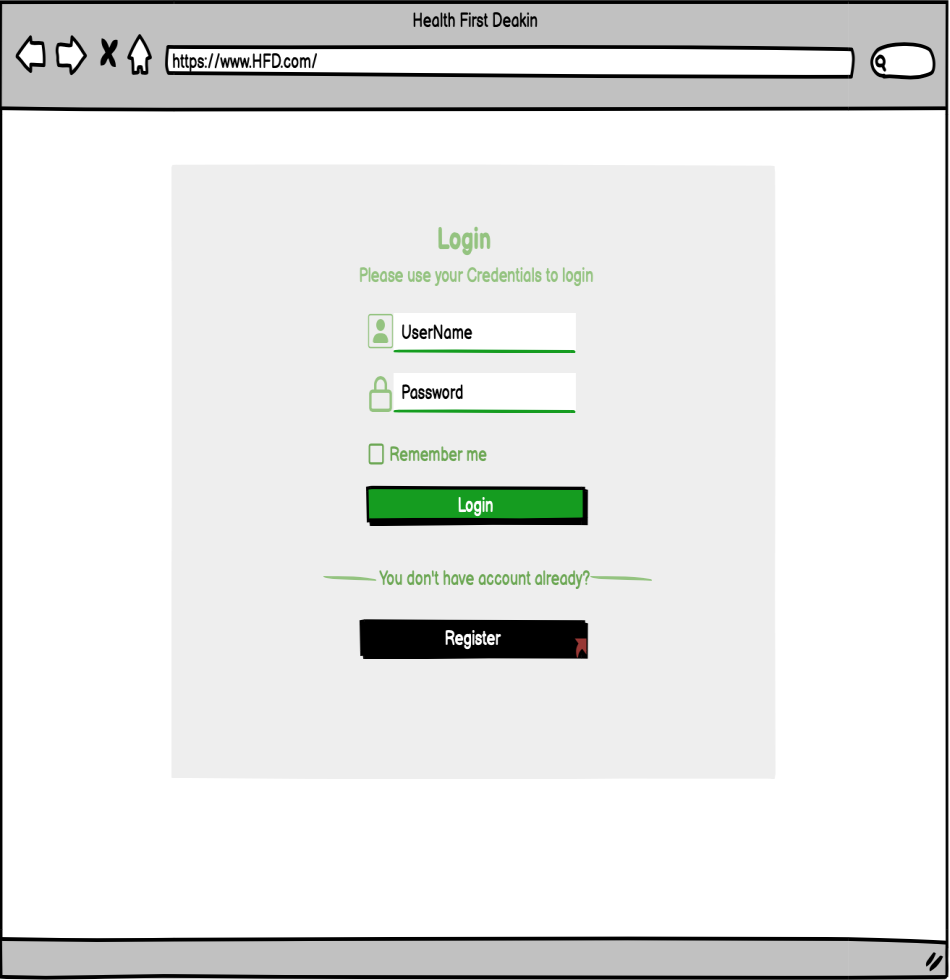
**Precondition:** The registered Clinic should have access to Health First Deakin on the web.

## Basic flow

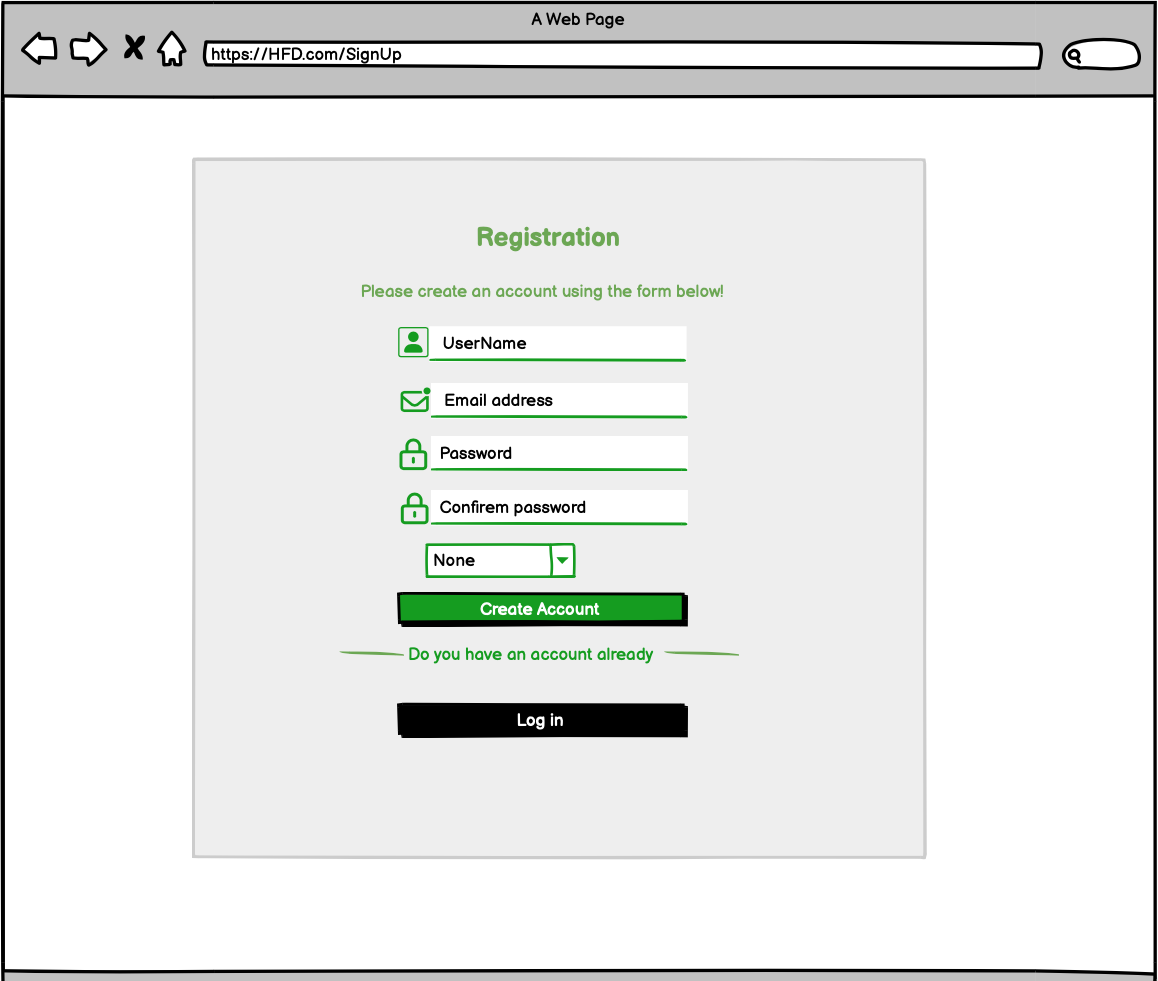
* + - 1. Clinic reception chooses the option to Booking.
      2. A list of doctors and their appointments will be view on the page.
      3. There will be list of present doctors who are available to check patients.
      4. User case ends successfully.

# 6 Prototypes

# Login



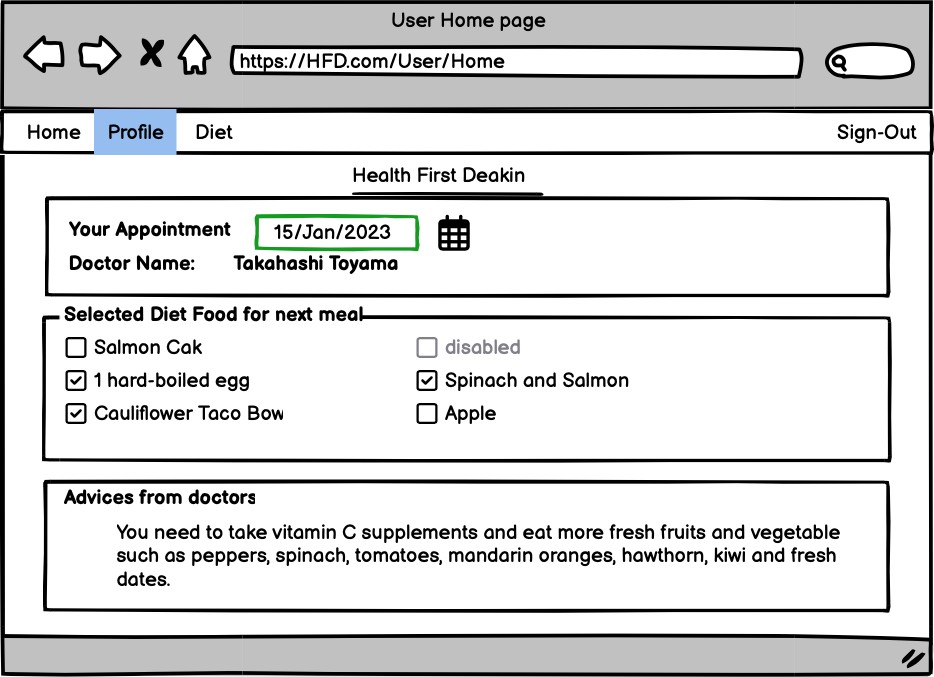
# Register



# Patients Profile

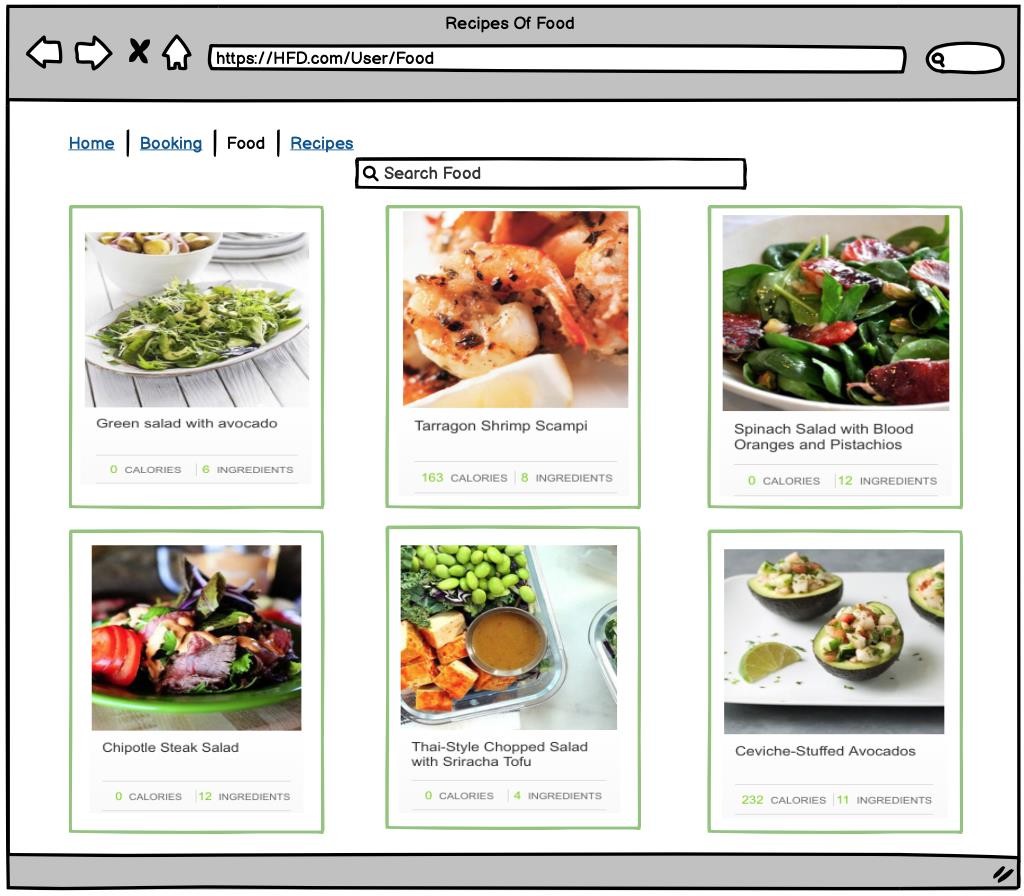
# New Patients

# User Home

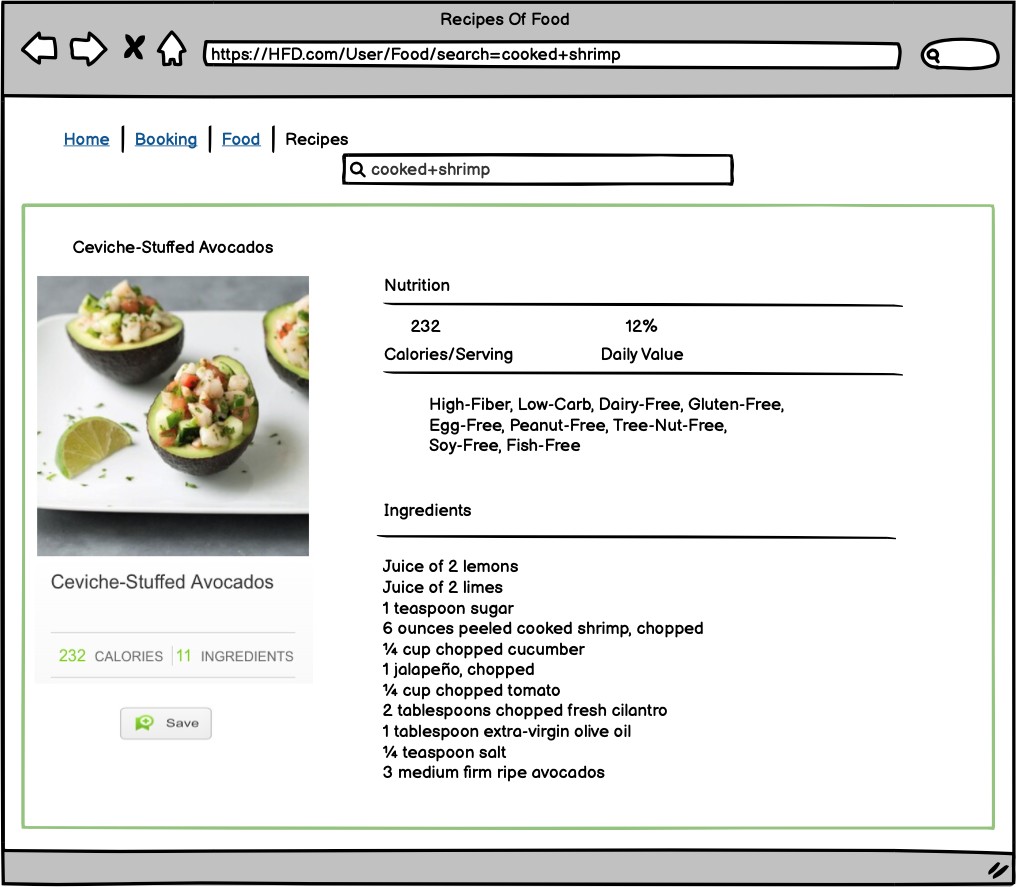


# User Profile

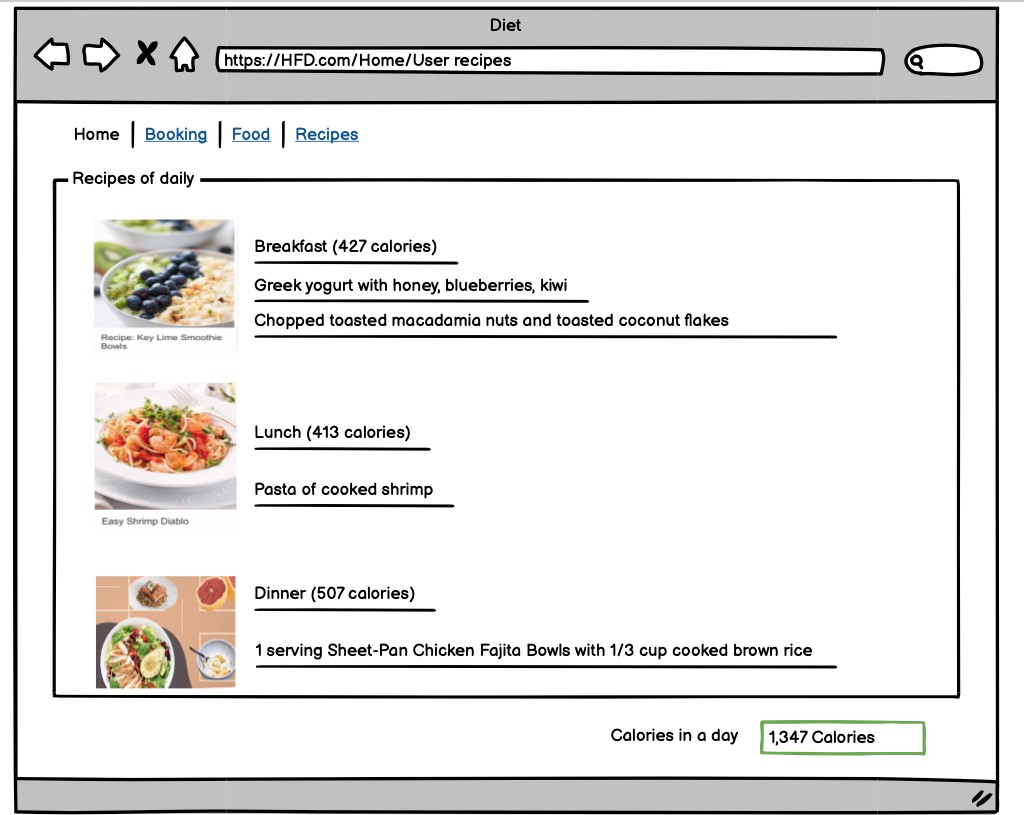
# Recipes Profile



# Details of Recipes

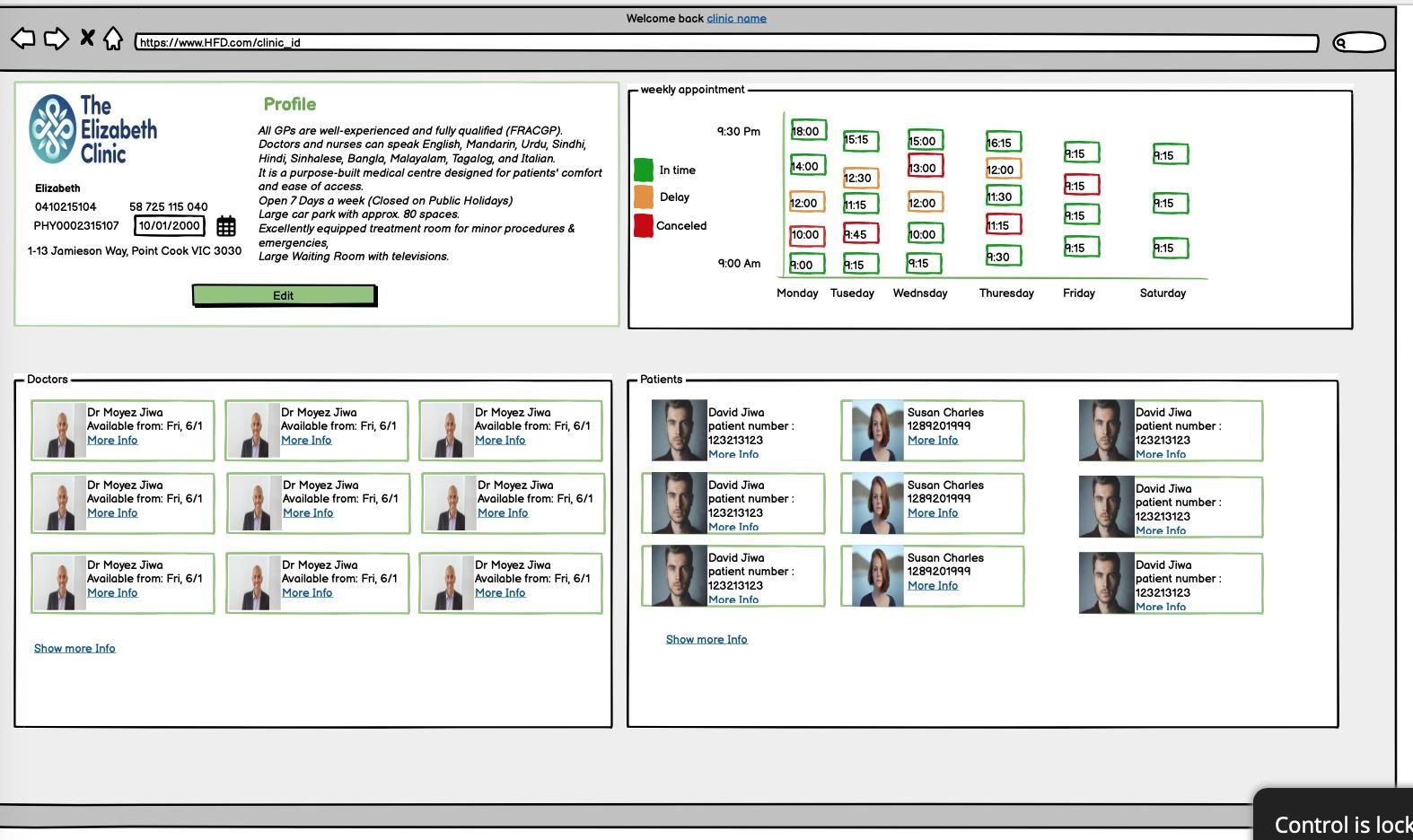


# User Recipes



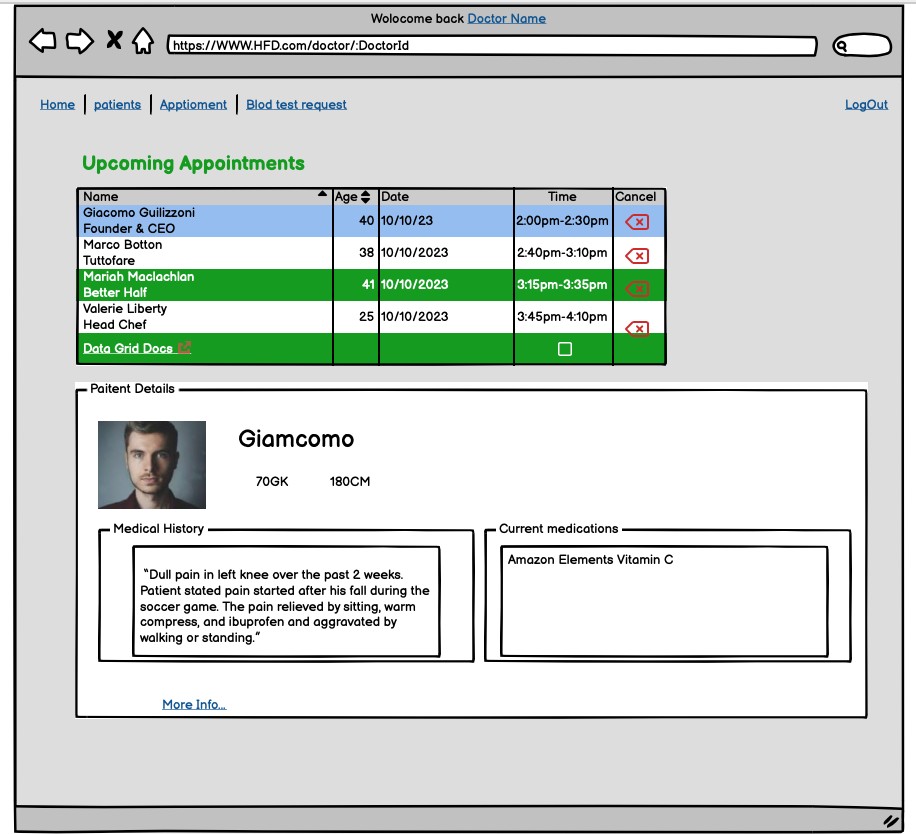
# Clinic information Register

# Clinic Profile

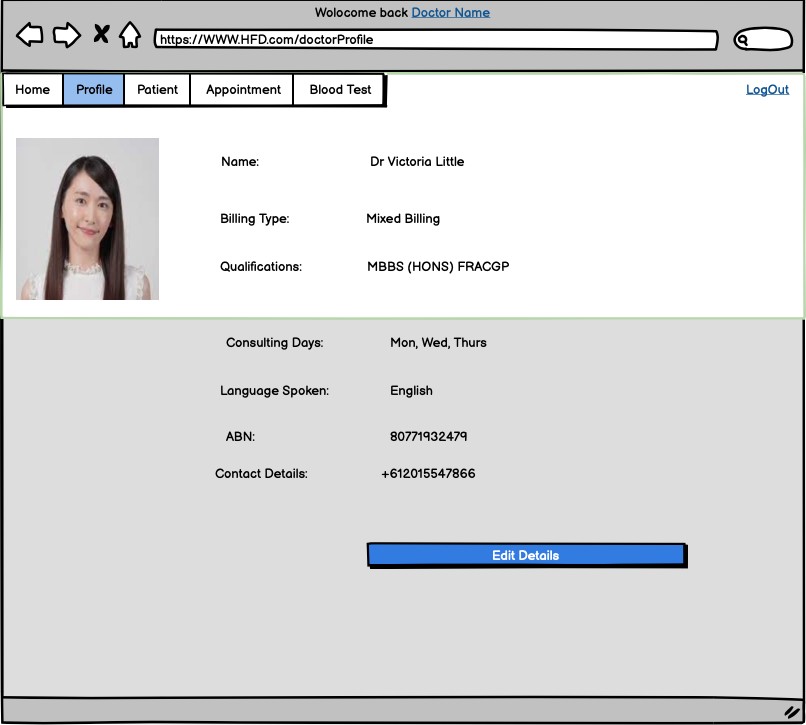


# Clinic Booking

# Upcoming Appointments



# Doctor Profile



# 8 Non-Functionality

* 1. Security

The system shall be designed with a level of security appropriate for the sensitivity of information enclosed in the database. More interaction is needed with client about the volatility of the information. Since there is no obvious information that is of a high security level such as credit card information, the only requirements that could be implemented are encrypting the database and/or making the database password-protected, by user’s request.

* 1. Binary Compatibility

This system will be compatible with any computer that has Microsoft Office Professional 2007 or later installed (whether PC or Mac), and will be designed with more than one computer in mind.

* 1. Reliability

Reliability is one of the key attributes of the system. Back-ups will be made regularly so that restoration with minimal data loss is possible in the event of unforeseen events. The system will also be thoroughly tested by all team members to ensure reliability.

* 1. Maintainability

The system shall be maintained by Health First Deakin, of the G’ Jo company, or delegated to another employee.

* 1. Portability

The system shall be designed in a way that shall allow it to be run on multiple computers with Microsoft Office Professional 2007 or later installed.

* 1. Extensibility

The system shall be designed and documented in such a way that anybody with an understanding of Microsoft Access shall be able to extend the system to fit their needs with the team’s basic instructions**.**

* 1. Reusability

The system should be designed in a way that allows the database to be re-used regularly for the various silent auctions that the organization shall hold.

* 1. Application Affinity/Compatibility

This system requires the Microsoft Office Professional 2007 suite or later, as it operates primarily through Microsoft Access, in conjunction with Microsoft Excel.

* 1. Serviceability

The maintenance of the system should be able to be sufficiently performed by any person with a basic understanding of Microsoft Access.

## Scenario A:

# 9 Project scenario

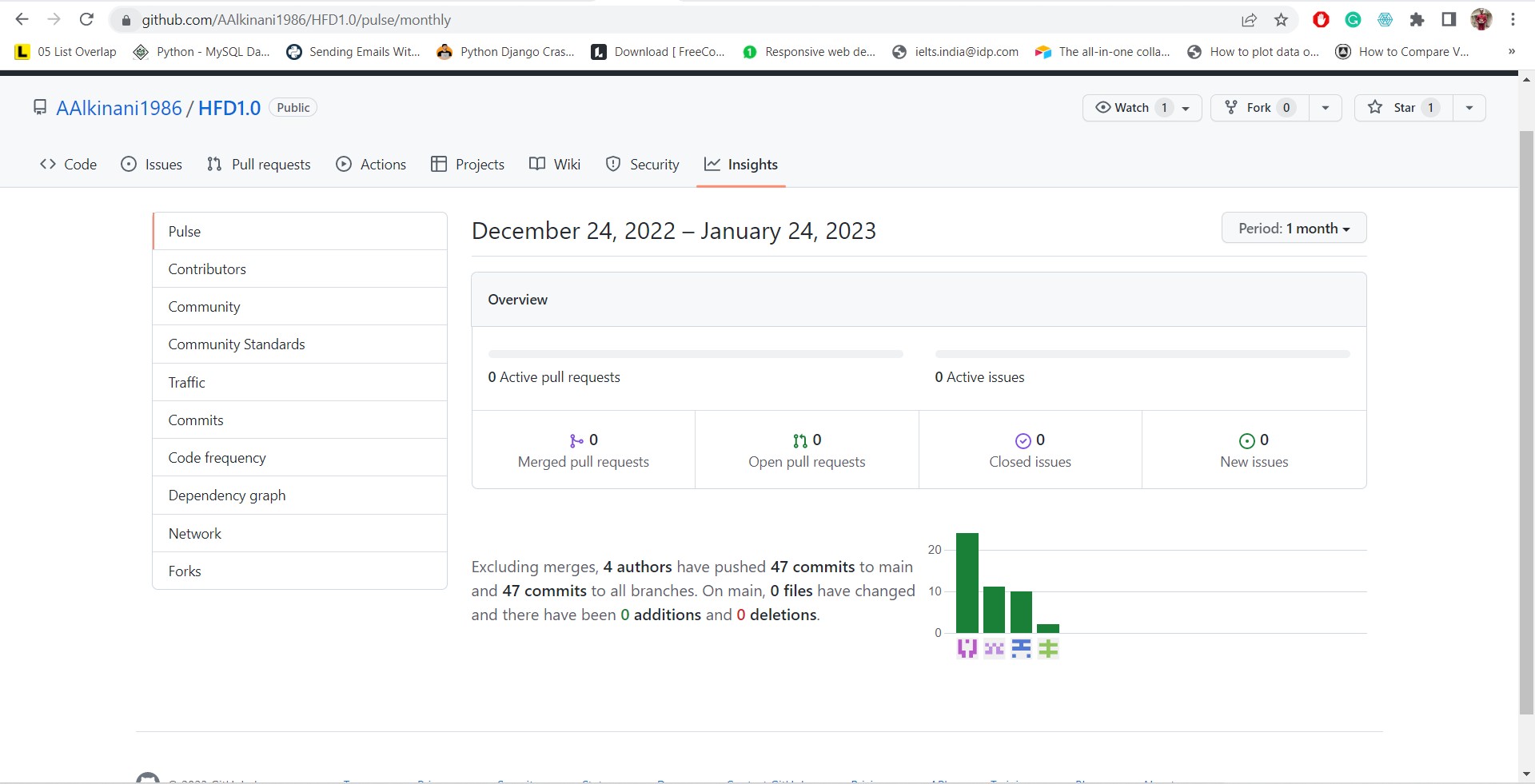
Initial Item Definitions the Patients shall enter the information about the items into the database for its initial construction and evolution. The fields will be completed via a form that will manipulate the data.

## Scenario B:

Database Maintenance The user may want to alter/delete information after the auction is over, in this case they will need to be able to remove the data that has been entered.

10 Appendix: Sprint 1

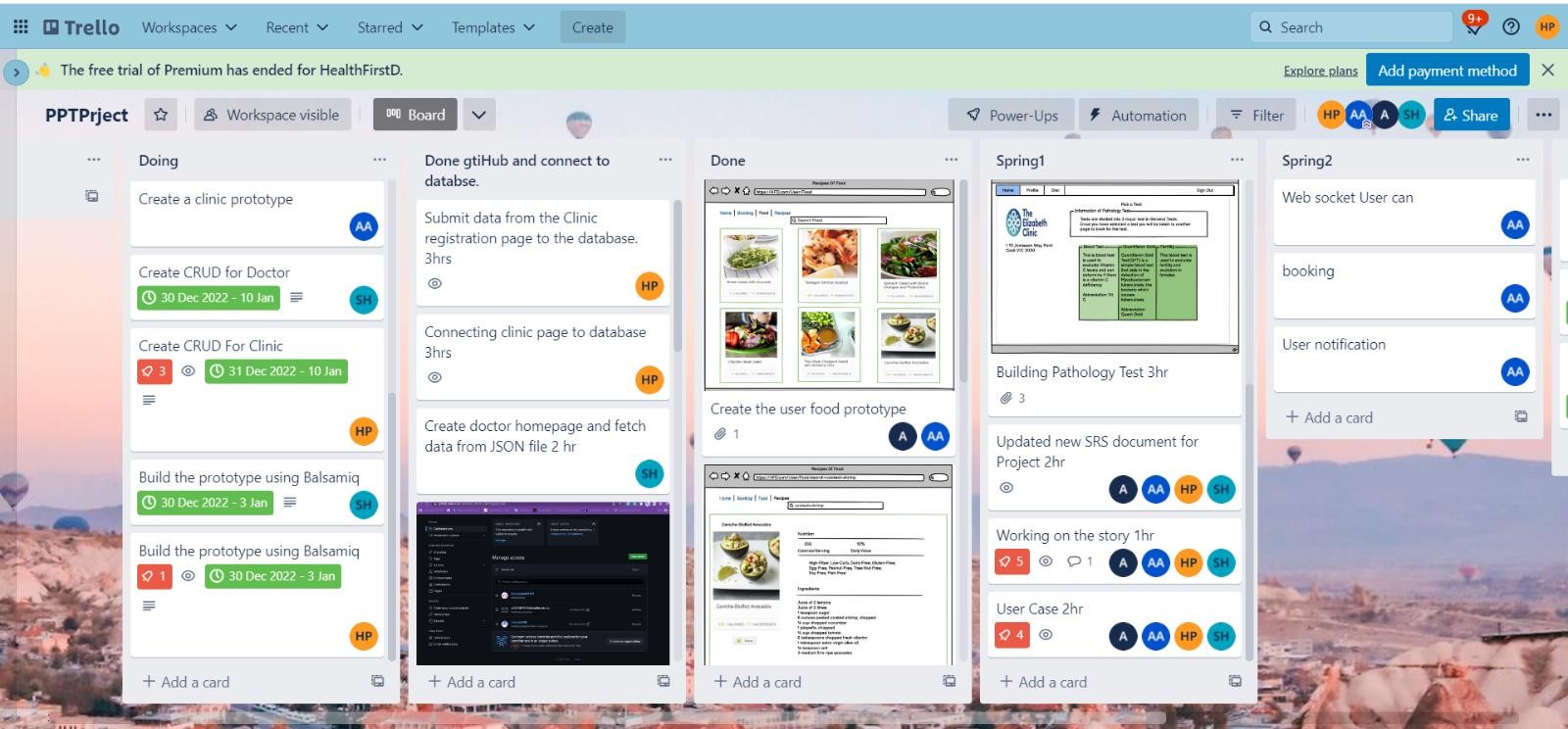
Git Pulse and Git Contribution



Graphical user interface

Description automatically generated

Trello Board



Task and Member

## Ali Akkineni

* Created User Prototype.
* Setting up environment for Git
* Creating authentication for login and logout
* Adding patients details to SRS.

## Harsh Patel

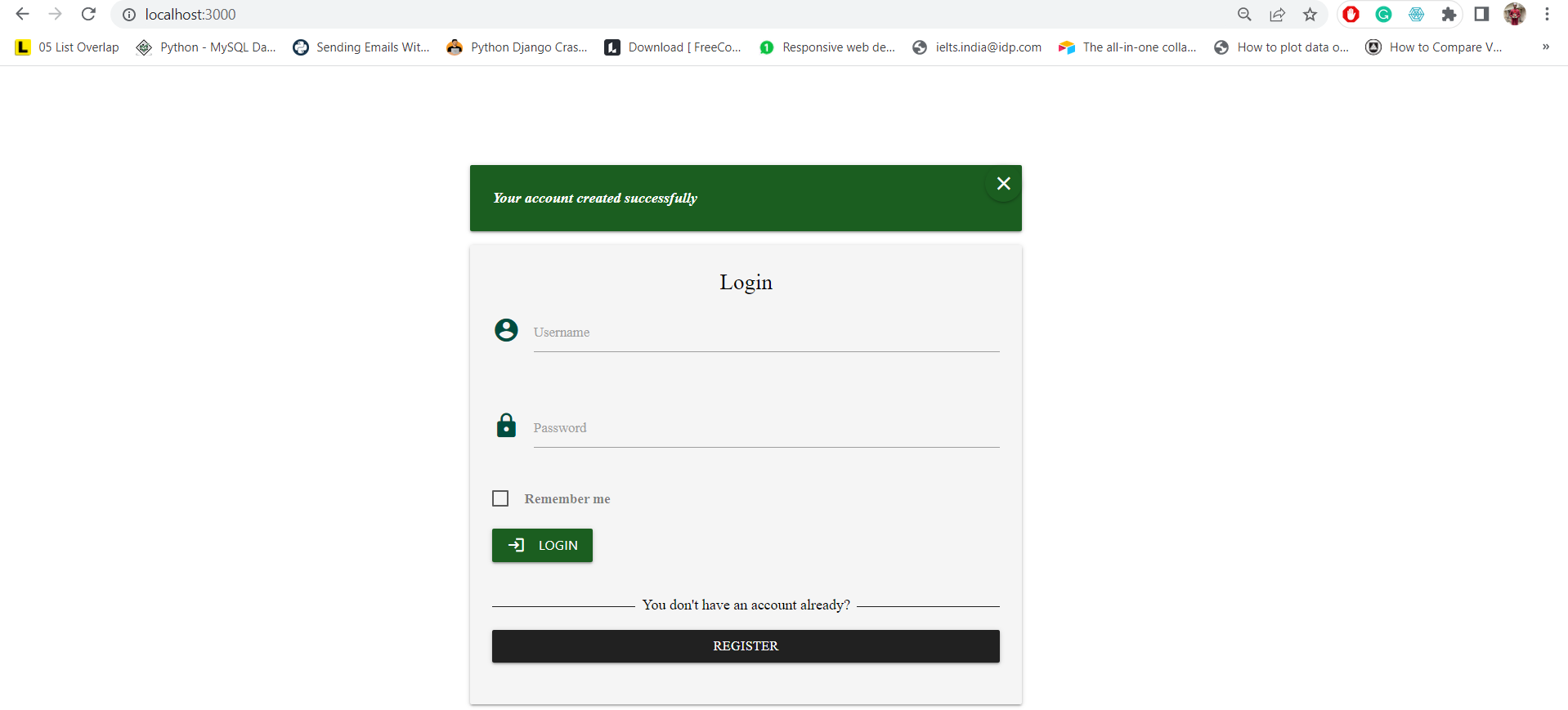
* Created Pathology Prototype.
* Working on Clinic Module.
* Creating Use cases and diagram of system and adding Clinic details
* Working on database for clinic module.

## Shamail Haider

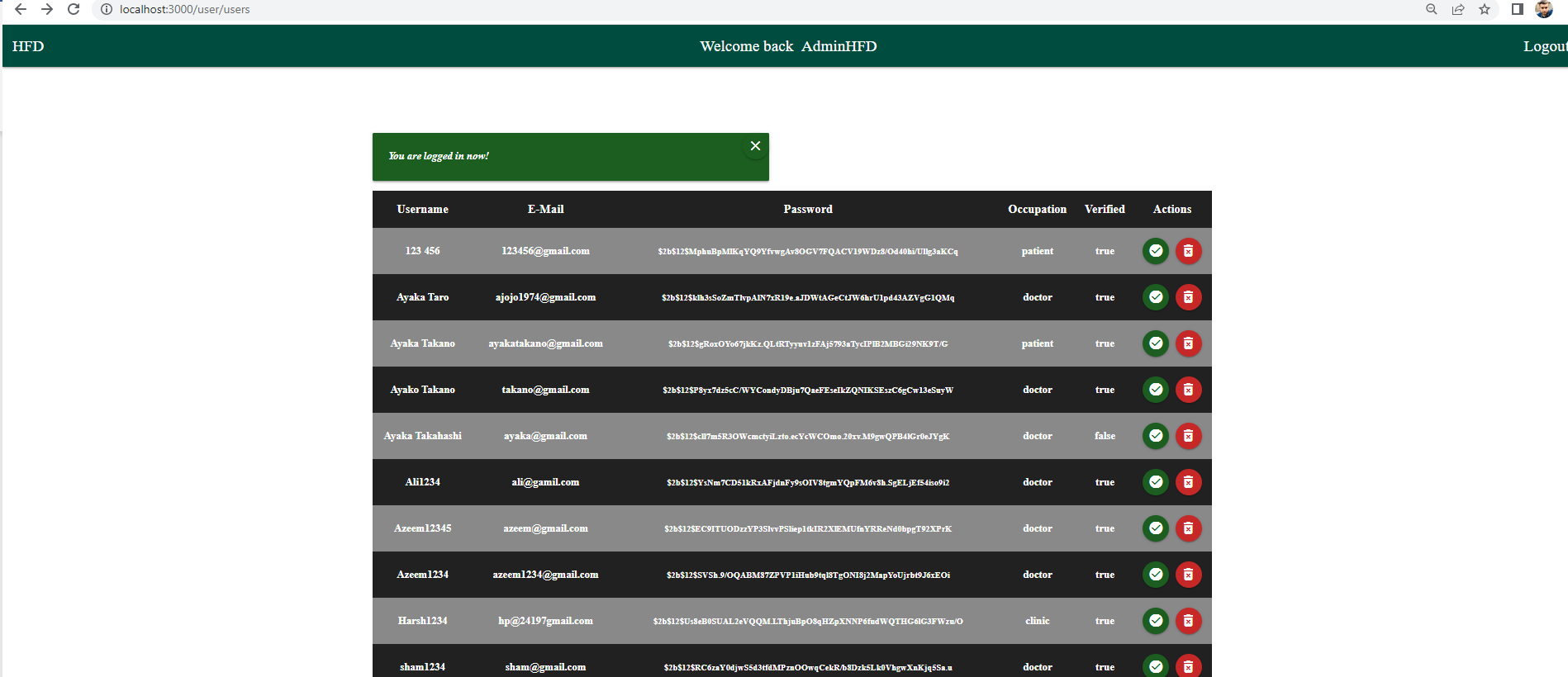
* Created Doctor Prototype.
* Created Doctor pages like registration page and connected it to the database.
* Fetched data from database and displayed it on page.
* Added everything related to Doctor Module on SRS.
* Created use case diagram, sequence diagram.

## Yaru Xu

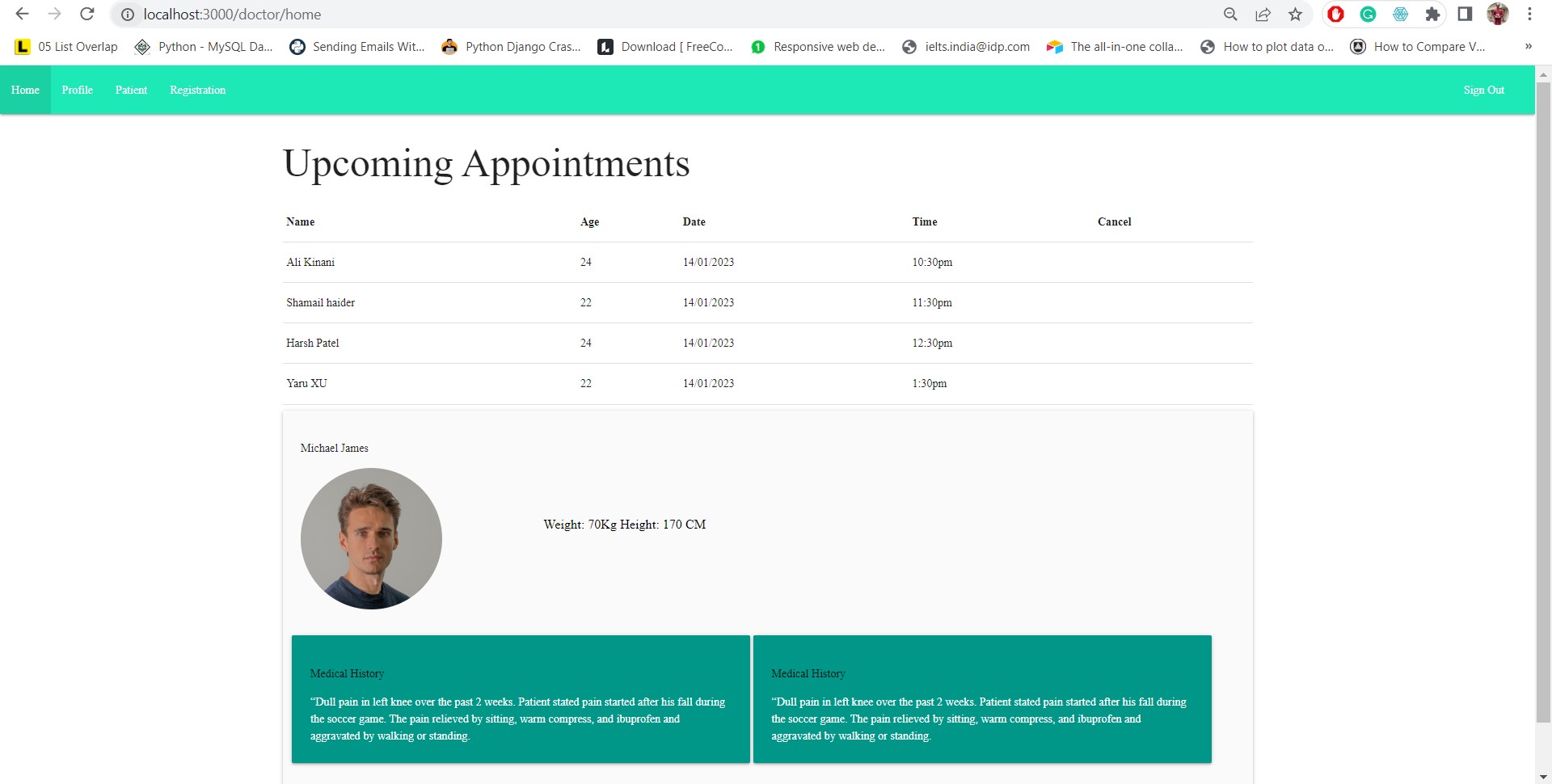
* Created Pathlogy Prototype.
* Created pages related to Diets.
* Currently working on API’s related to diet recopies.

Snapshots of applications Login Page

Users List



Doctor Home page

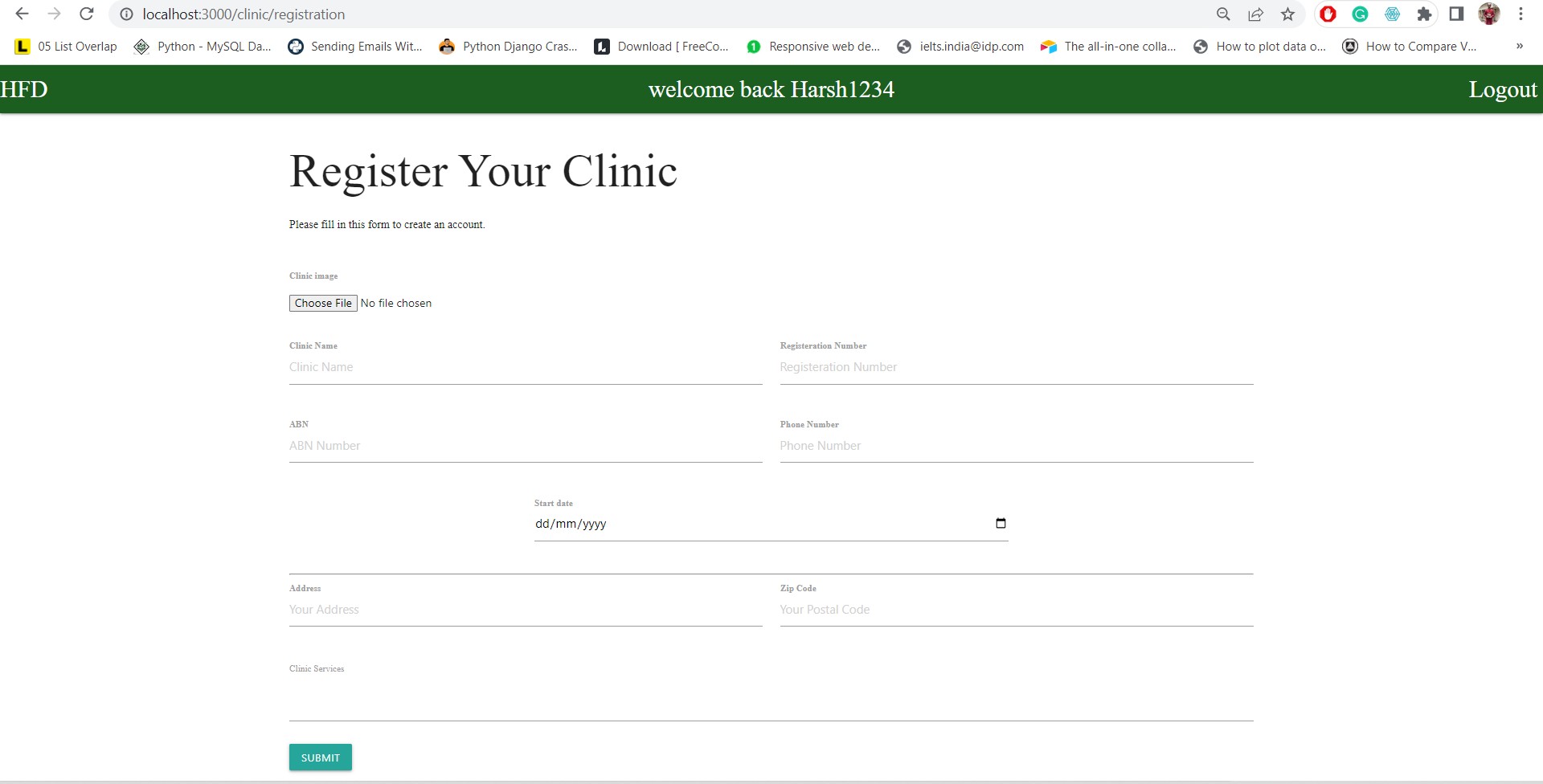


Doctor Registration Page

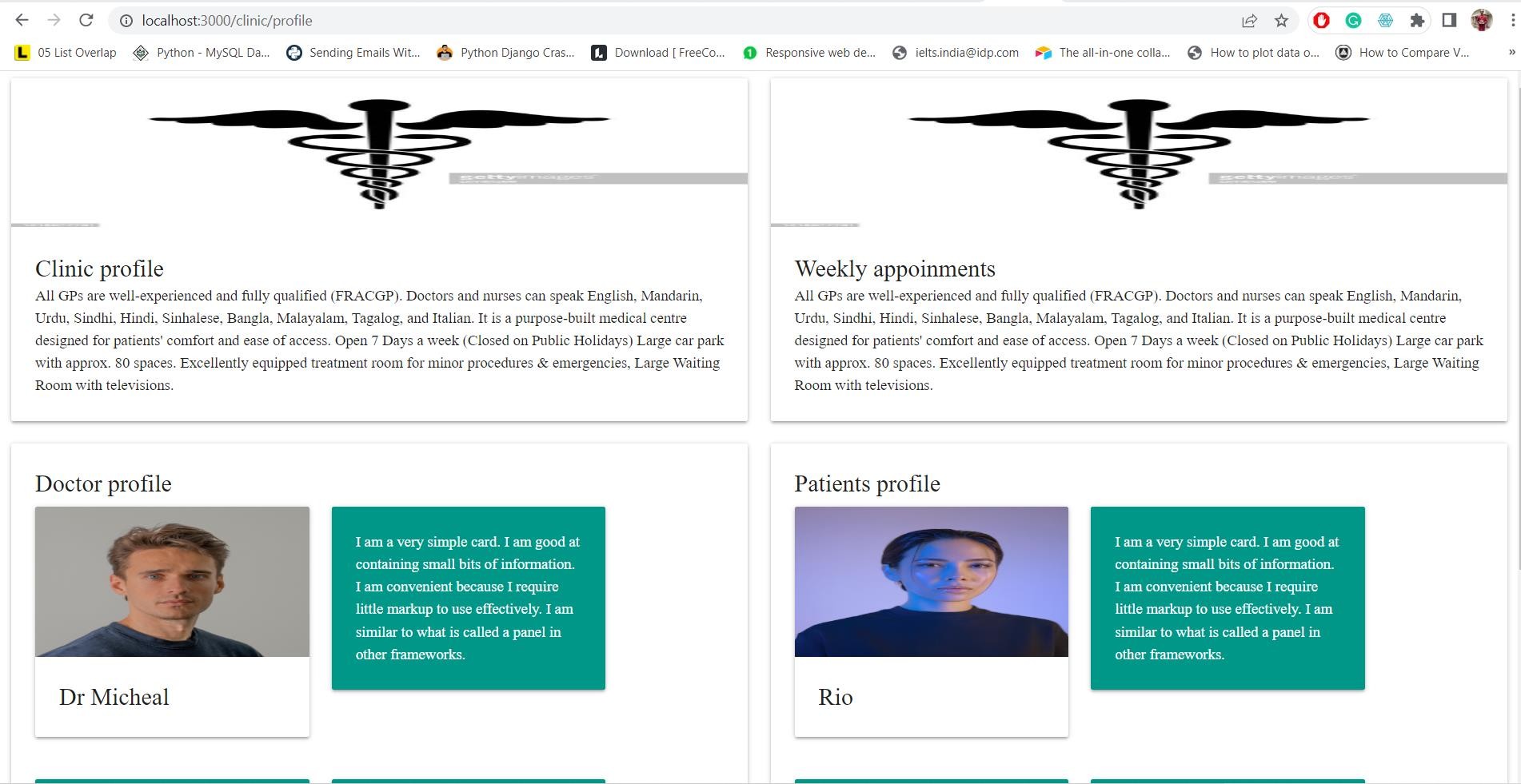
Graphical user interface, text, application, email

Description automatically generated

Clinic register page.



Clinic Profile



**Reflection by Shamail Haider:**

While working on this project as a group, I had a very good time, and it gave a very good opportunity of learning from others and gaining an experience of how to work with a team. We all connect with each other every morning for 10mins and takes an update from each other and discuss today’s goals. This project gave me an idea of how work is done in professional environment. Node js is very much in demand and gaining skills over it will help me a lot in my carrier. I also learnt about various stages involved in a successful project delivery.

**Reflection by Harsh Patel:**

Learning latest technology like node js and working as a team gave me a good experience to work on Various task and gain knowledge from different group members moreover, I also came up with various industrial project management skills for instance creating brainstorm, making interactive prototypes having frequent catchup meetings for designing gave me a good opportunity to work on different ideas and implement them. I also improved my skills related to development skill with the help of different team members. To keep it in a nutshell a better experience to work as a team helped me improve in all the aspects of project development process.

**Reflection by Yaru Xu:**

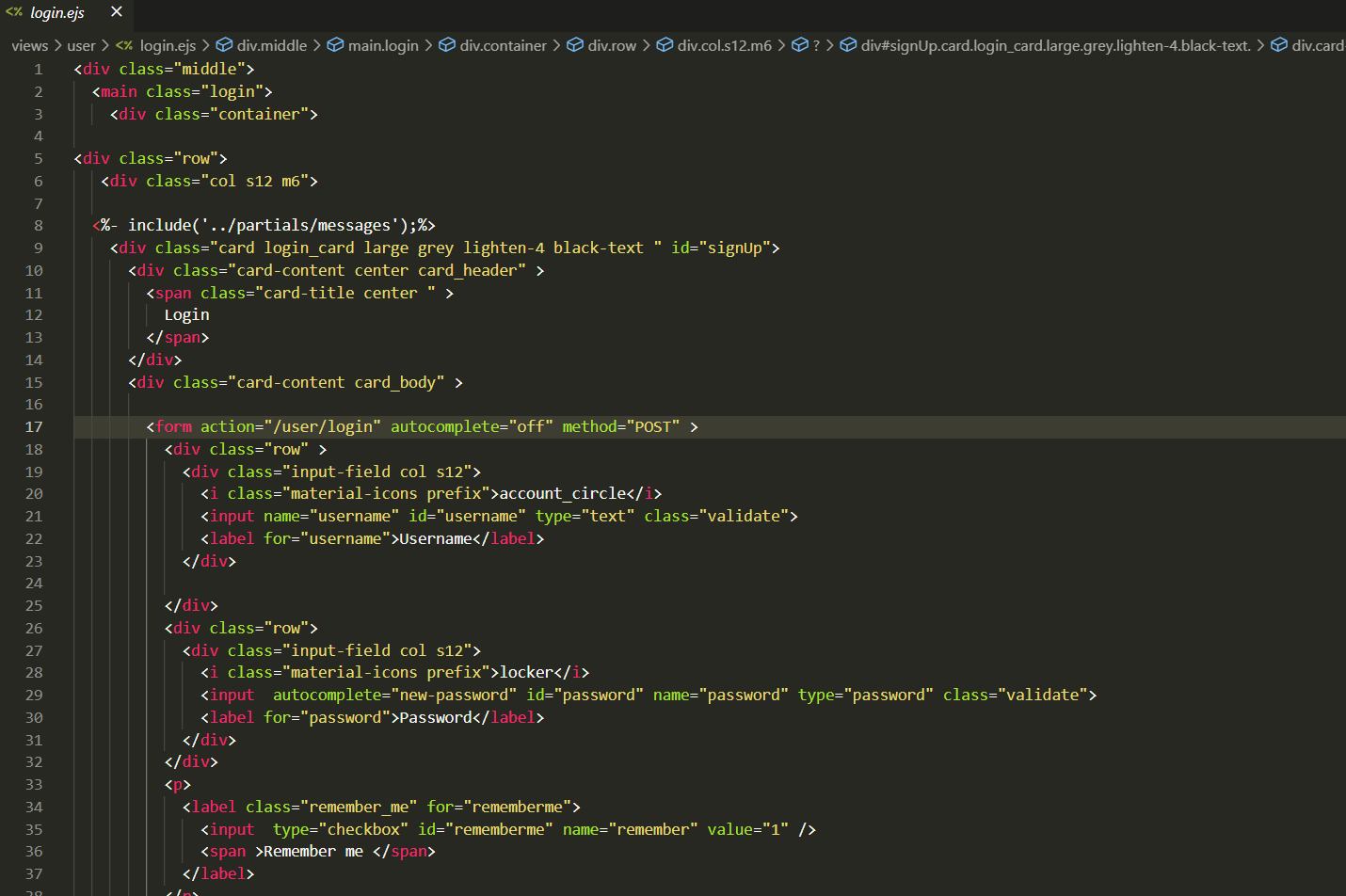
Working with a team, helped me very much, I worked with the api’s, fetched and posted data. While working on this project, I got a smell of professional working or industrial environment. There were some challenges but with each other we always came with a perfect solution. Node js is a very powerful language and it is also very beneficial for my future aspirations. I learnt about how to plan a project and what are steps involved in a project from start to delivery.

**Reflection by Ali Adil Abdul Razzaq:**

Working in a team. Gave me a good exposer to learning about different project ideas and how to implement them at a professional project as everyone carries a different working experience. I also came across many challenges but with the support of different team member I was able to learn how to coordinate and work as a team other than working in a team it also developed my skills regarding management which will be a boon for me when I will work in IT industry as a team member

**Code Snippets:**

**User Login Page:**



Text

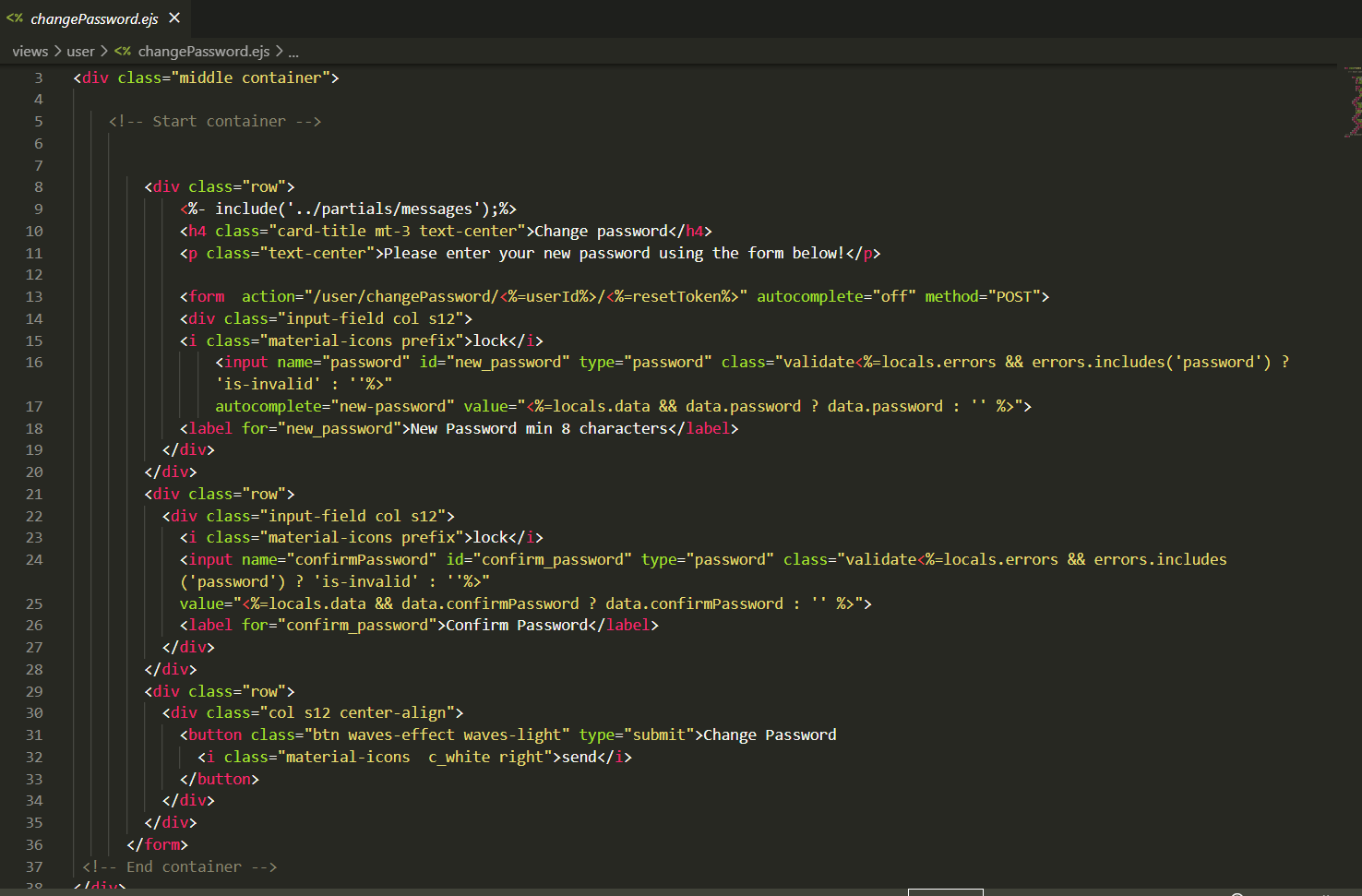
Description automatically generated

**User List Page:**

Text

Description automatically generated

**Change Password:**



**Registration Page For User:**

**Text

Description automatically generated**

**User Controller:**

**Text

Description automatically generated**

**User Model:**

**Text

Description automatically generated**