

Project Report

## DOCTOR APPOINTMENT SYSTEM USING DJANGO, ANGULAR AND PYTHON

## Module Title: Web Development for Information Systems

## Module Code: B9IS109

## GROUP MEMBERS:

## SINGUPURAM NAVIN KUMAR | 10612366

## Module Leader: Dr Obinna Izima

TABLE OF CONTENTS

[1. INTRODUCTION 4](#_Toc131579637)

[How to properly Use the Doctor Appointment Booking System 4](#_Toc131579638)

[2. RESEARCH AND PLANNING 5](#_Toc131579639)

[3. CHOICE OF FRAMEWORK AND TECHNOLOGIES 5](#_Toc131579640)

[3.1. PYTHON 5](#_Toc131579641)

[3.2. DJANGO FRAMEWORK 5](#_Toc131579642)

[3.3. SQL LITE 3 6](#_Toc131579643)

[3.4. Angular 6](#_Toc131579644)

[3.5. Bootstrap Templates 6](#_Toc131579645)

[3.6. AWS 7](#_Toc131579646)

[4. UX DESIGN 7](#_Toc131579647)

[4.1 FLOW DIAGRAM 7](#_Toc131579648)

[4.2 HOME PAGE 8](#_Toc131579649)

[4.3 ABOUT US PAGE 9](#_Toc131579650)

[4.4 BOOK AN APPOINTMENT 10](#_Toc131579651)

[4.5 SERVICES 11](#_Toc131579652)

[4.6 TEAM 12](#_Toc131579653)

[4.7 CONTACT US 12](#_Toc131579654)

[4.8 FOOTER 12](#_Toc131579655)

[4.9 SIGNUP 13](#_Toc131579656)

[4.10 LOGIN 13](#_Toc131579657)

[4.11 SEARCH PAGE 14](#_Toc131579658)

[5. BACKEND 15](#_Toc131579659)

[5.1 DATABASE SETTING 15](#_Toc131579660)

[5.2 DJANGO ADMIN PAGE 17](#_Toc131579661)

[6. WEB SERVICES 20](#_Toc131579662)

[6.1 SIGNUP SCREEN 21](#_Toc131579663)

[6.2 LOGIN SCREEN 22](#_Toc131579664)

[6.3 LIST OF DOCTORS VIEW 23](#_Toc131579665)

[6.4 Book Appointment Backend 23](#_Toc131579666)

[6.5 Get Appointment Backend 24](#_Toc131579667)

[6.6 Delete Appointment Backend 24](#_Toc131579668)

[7. SECURITY AND MEASURES 25](#_Toc131579669)

[7.1. JWT BEARER AUTHENTICATION 25](#_Toc131579670)

[8. WEB APPLICATION FEATURES 28](#_Toc131579671)

[9. AWS Deployment 30](#_Toc131579672)

[10. LINKS 33](#_Toc131579673)

[11. REFERENCES: 33](#_Toc131579674)

TABLE OF FIGURES

[Figure 1: Doctor Appointment Flow Diagram 8](#_Toc131583544)

[Figure 2: Home Page 9](#_Toc131583545)

[Figure 3: About Us 10](#_Toc131583546)

[Figure 4: Appointment Booking 11](#_Toc131583547)

[Figure 5: Services 11](#_Toc131583548)

[Figure 6: Team 12](#_Toc131583549)

[Figure 7: Contact Us 12](#_Toc131583550)

[Figure 8: Footer 13](#_Toc131583551)

[Figure 9: SignUp Page 13](#_Toc131583552)

[Figure 10: Login Modal Popup 14](#_Toc131583553)

[Figure 11: Search 14](#_Toc131583554)

[Figure 12: SQL Lite 3 setting in the py setting file 15](#_Toc131583555)

[Figure 13: Database XML File 15](#_Toc131583556)

[Figure 14: Linux Virtual Box 16](#_Toc131583557)

[Figure 15: Vagrant File for Imaging 17](#_Toc131583558)

[Figure 16: Admin Login Page. 18](#_Toc131583559)

[Figure 17: Django Administration 18](#_Toc131583560)

[Figure 18: Adding the Doctors 19](#_Toc131583561)

[Figure 19: Adding the Specialists 19](#_Toc131583562)

[Figure 20: Adding the Users 20](#_Toc131583563)

[Figure 21: API Services URL 20](#_Toc131583564)

[Figure 22: SignUp Logic 21](#_Toc131583565)

[Figure 23: Token-based Logic 22](#_Toc131583566)

[Figure 24: Login JWT Authentication 22](#_Toc131583567)

[Figure 25: Form Submission for Doctor Appointment 23](#_Toc131583568)

[Figure 26: Book Appointment Logic 23](#_Toc131583569)

[Figure 27: Get Appointment Logic 24](#_Toc131583570)

[Figure 28: Delete Appointment Logic 24](#_Toc131583571)

[Figure 29: JWT Settings 25](#_Toc131583572)

[Figure 30: Tokenization 26](#_Toc131583573)

[Figure 31: Google Api OAuTH Tokenisation 26](#_Toc131583574)

[Figure 32: Login Hash 27](#_Toc131583575)

[Figure 33: Login and Logout Feature 28](#_Toc131583576)

[Figure 34: Login and Register popup 28](#_Toc131583577)

[Figure 35: Search Feature 29](#_Toc131583578)

[Figure 36: List Appointment Feature 30](#_Toc131583579)

[Figure 37: AWS EC2 Deployment 30](#_Toc131583580)

[Figure 38: Putty Console deployment 31](#_Toc131583581)

[Figure 39: Commands used for Deployment via Putty 33](#_Toc131583582)

# 1. INTRODUCTION

A doctor appointment booking system is an online tool that enables patients to conveniently schedule an appointment at a specific clinic for a health concern. Examples include dental, injury, mental health, muscle strain, obesity, arthritis, and asthma. Around 95% of the world's population suffers from multiple diseases. Several of those schedule visits with a neighboring doctor each week in order to receive the treatment. Also, you can benefit greatly from a doctor appointment booking system on your clinic's or doctor's website. It's a booking form with various fields that the patient must complete and submit in order for it to be considered an appointment. The patient and you both save a ton of time over this entire procedure. Also, you can quite effectively manage the many components of the firm using this.

The following list of advantages of a doctor booking system contains a variety of them.

* Time and money savings.
* Saves resources.
* Makes it easy to manage all the appointments and bookings.
* Proper following of the guidelines in times of health crisis.
* No rush hours and better sessions with the doctor.
* Better flow of income.
* Income from pre-bookings.
* Convenient for patients to easily book appointments.
* And much more.

## How to properly Use the Doctor Appointment Booking System

The website's front page is the best place for a medical appointment booking tool. At the top of the time is the form. So that the sick person may quickly arrange a doctor appointment without having to look through the internet once they arrive there.

Other health services pages, the footer, and the contact page on the doctor's website are additional areas you would want to take into account for adding an online doctor booking form.

Since many users have a tendency to open the contact us page right away after entering a website. Hence, failing to consider that would be a grave error in site design.

# 2. RESEARCH AND PLANNING

A system that will be user-friendly and solve a user's immediate problem was designed and built throughout a vital phase of research and planning. So, it was determined to create a system for scheduling medical appointments. Once the application was selected, the database design, framework selection, and entity design were all made by the requirements. I chose to use the open-source, simple-to-configure **SQL Lite 3 database**, which can be scaled as necessary. The framework was finished after the database was constructed. The **Django Python framework** was used to begin the development process because it offers the greatest degree of flexibility. Additionally, the **Angular Web Development Framework** was used to simplify the development of an application while gaining the advantages of using **Bootstrap** and other front-end development technologies. To streamline the development process, the app's screens, features, and routes were chosen before installation, setup, and actual work began.

# 3. CHOICE OF FRAMEWORK AND TECHNOLOGIES

# 3.1. PYTHON

The parts of a website or program that visitors don't see—the back end—are typically made with Python. For transferring data to and from servers, processing data, interacting with databases, routing URLs, and guaranteeing security, Python can be used in web development. Python offers several frameworks for web development. The application was created using the Python Django framework for the back end and the Angular Framework for the front end, both of which use the most recent version of Python, 3.11.2. For transferring data to and from servers, processing data, interacting with databases, routing URLs, and guaranteeing security, Python can be used in web development.

# 3.2. DJANGO FRAMEWORK

Model-View-Controller (MVC) architecture is used by the high-level Python web framework Django. By offering built-in components and tools for typical web development activities, it is intended to assist developers in creating online applications more rapidly and with less code.

Django has several important features, including:

Database interaction with ORM (Object-Relational Mapping)

* Admin interface built-in for data management and user authentication
* HTTP requests and responses are handled via URL routing and view management.
* engine for rendering HTML pages templates
* processing and validation of forms
* Cross-site scripting (XSS) and cross-site request forgery (CSRF) protection are examples of security features

# 3.3. SQL LITE 3

SQLite is one of the most well-liked and approachable relational database systems. It excels above other relational databases in many ways. The benefit of using SQLite3 is that it is simpler to set up and use, and the completed database is only one file that can be sent through email or kept on a USB memory stick.

The feature that makes it desirable are:

* The program is open-source.
* It works well on all platforms.
* Simple work with several sessions.

# 3.4. Angular

Utilizing Angular, one of the most widely used frameworks, for the frontend application. Angular gives developers the resources they need to create and organize complex JavaScript applications. In addition, Angular has several significant benefits over some rivals. For instance, Google employees created and maintain Angular. Along with these engineers, there is a sizable community available to assist you with problems as they arise. Angular is a code library that aims to make it easier to pair Django with AngularJS on the front end.

# 3.5. Bootstrap Templates

The most widely used CSS framework for creating responsive and mobile-first websites is Bootstrap. The templates are designed to allow for easy customization to meet the needs. It is very easy to combine and use many templates.

# 3.6. AWS

An online service called Amazon Elastic Compute Cloud (Amazon EC2) offers safe, scalable processing capability in the cloud. It is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes.

# 4. UX DESIGN

# 4.1 FLOW DIAGRAM

This Doctor Appointment System involves two parties: the admin, which makes use of the Django Framework, and the user. Admin will have the right to interact with the functionalities to add data to the application such as specialties accounted.

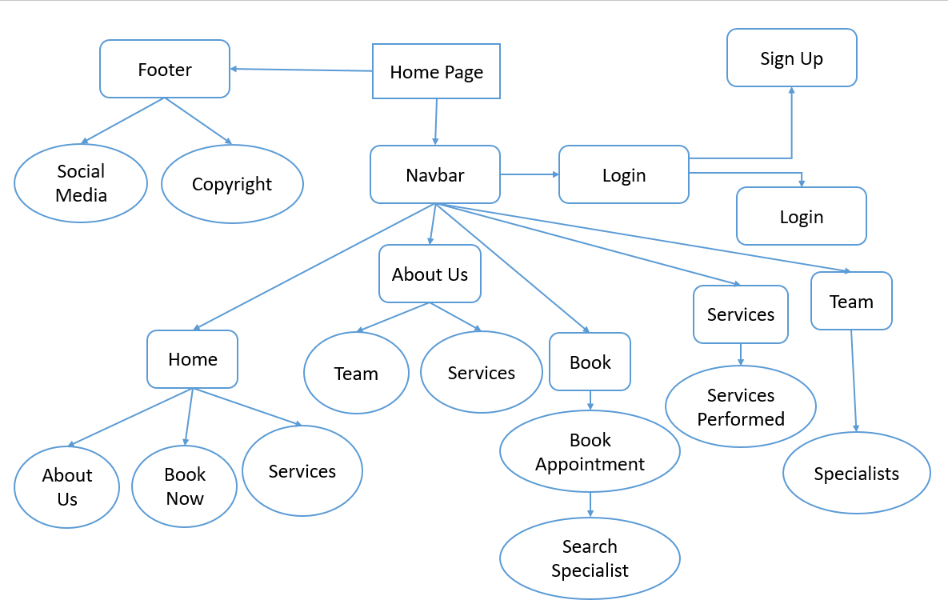
Let’s understand the flow:

User flow:

* Register to the Doctor Appointment Management System by adding a username and password.
* Log in with the Credentials
* Create a new appointment using the application by adding Full name, email date, phone, timing, and the specialist.
* Newly created appointments would be there in the view status and also an option to delete when needed.
* View Contact information on the website
* View About us, our services, and about team information on the website

Admin Django Flow:

* Sign up for the Django administration as a super user to manage appointments, authentication, and authorization.
* Enter the credentials to log in.
* Access appointments, physician, and specialty histories.
* View the history of groups and users as part of the authentication, and authorization.
* Add doctors and their specialties via the admin panel.



**Figure 1: Doctor Appointment Flow Diagram**

# 4.2 HOME PAGE

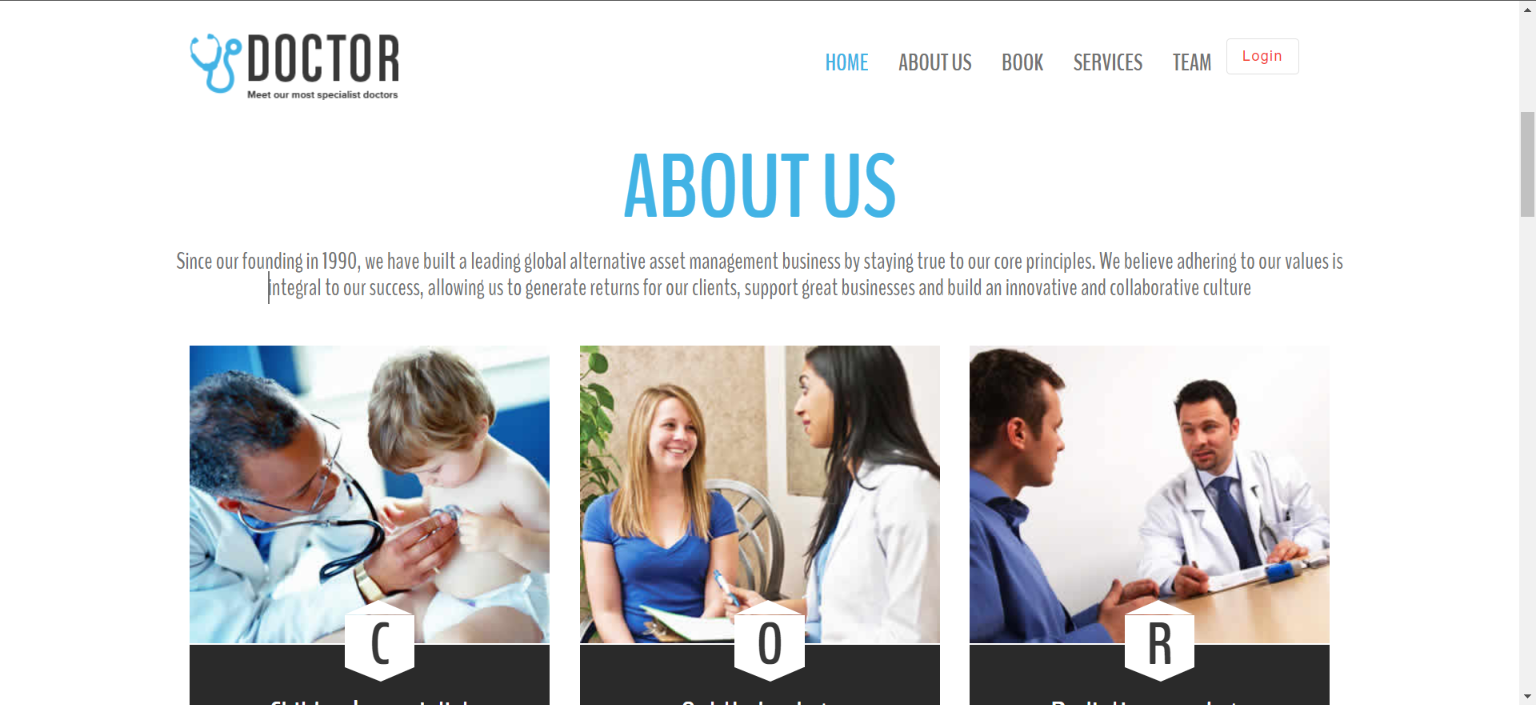
The homepage consists of six sections as it is a single-page application. The user has options for navigating around the home page, about us, book, services, and team using the Navbar. The user can also book an appointment from the book now section by providing details such as name, details, contact number, time slot needed, and specialist required.

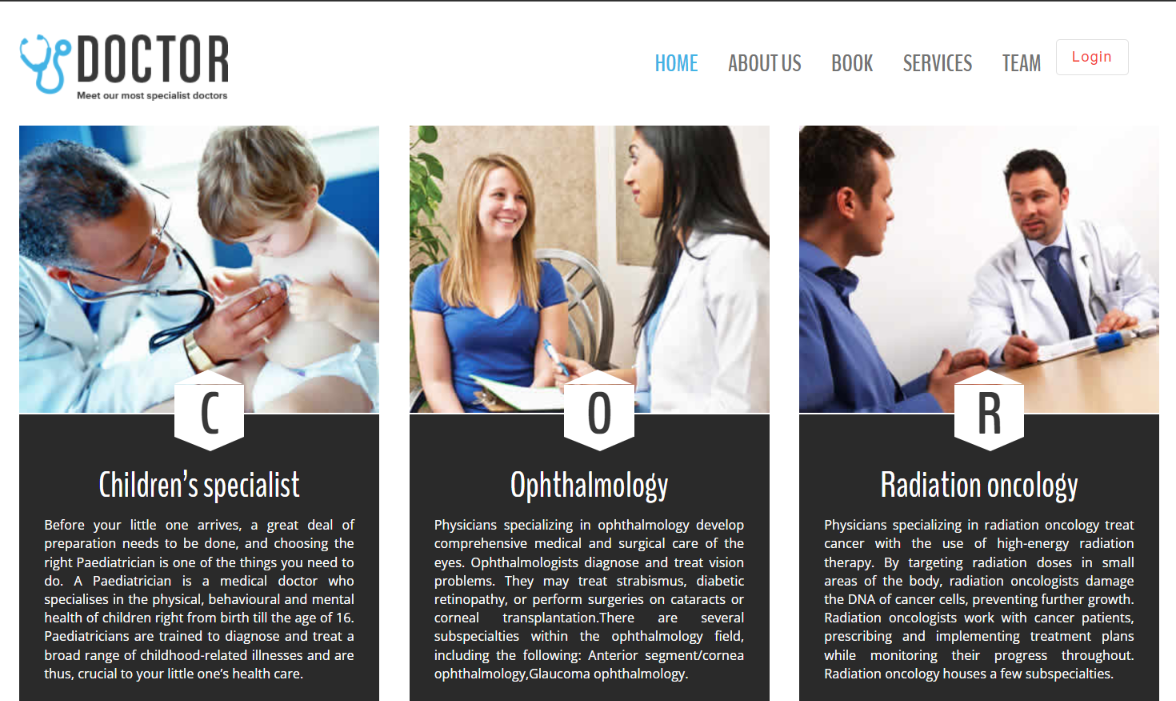


**Figure 2: Home Page**

# 4.3 ABOUT US PAGE

This section consists of the details of the doctors and the specialists provided in the application such as the Children’s specialist, ophthalmology, radiation oncology, etc. Also, the description of the different specialists providing services.

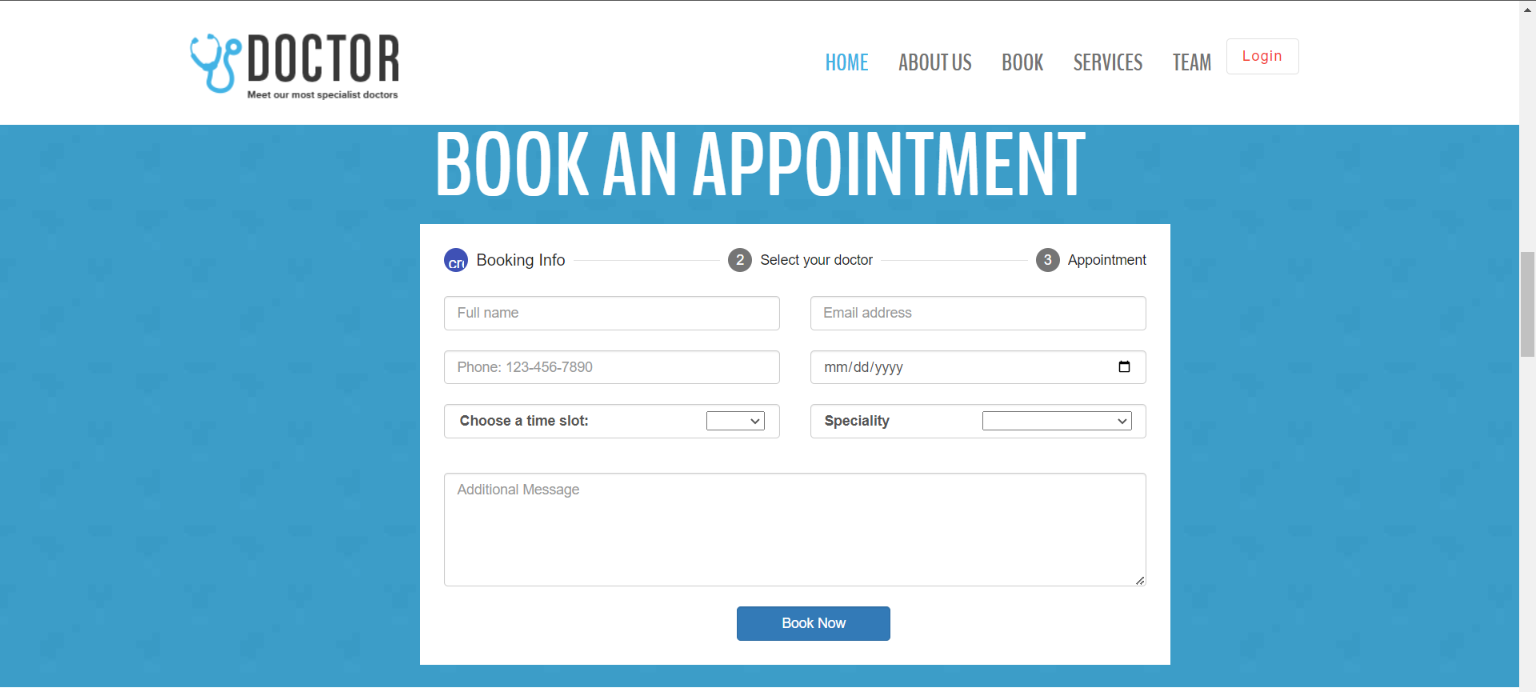




**Figure 3: About Us**

# 4.4 BOOK AN APPOINTMENT

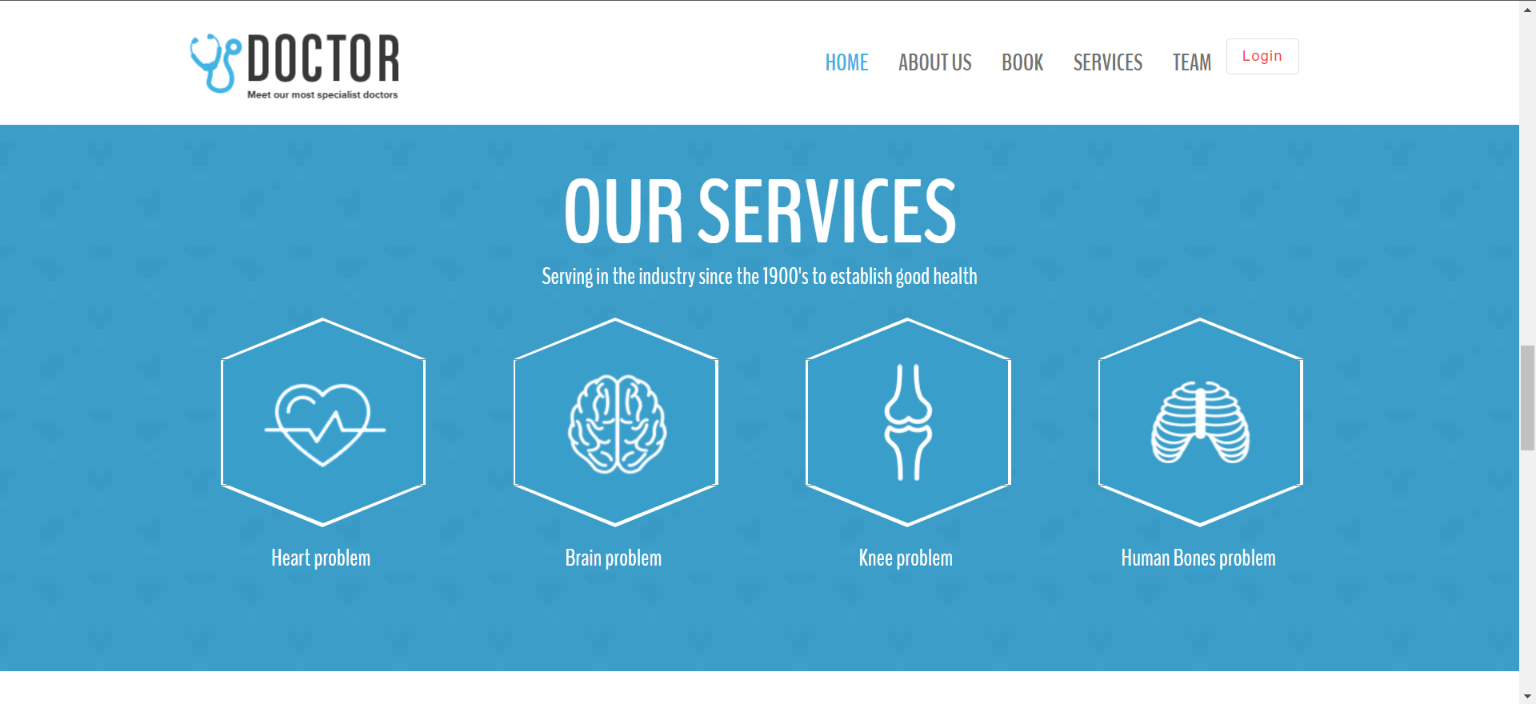
This section consists of a few fields to punch in and book an appointment on a specific day and time with a specialist.



**Figure 4: Appointment Booking**

# 4.5 SERVICES

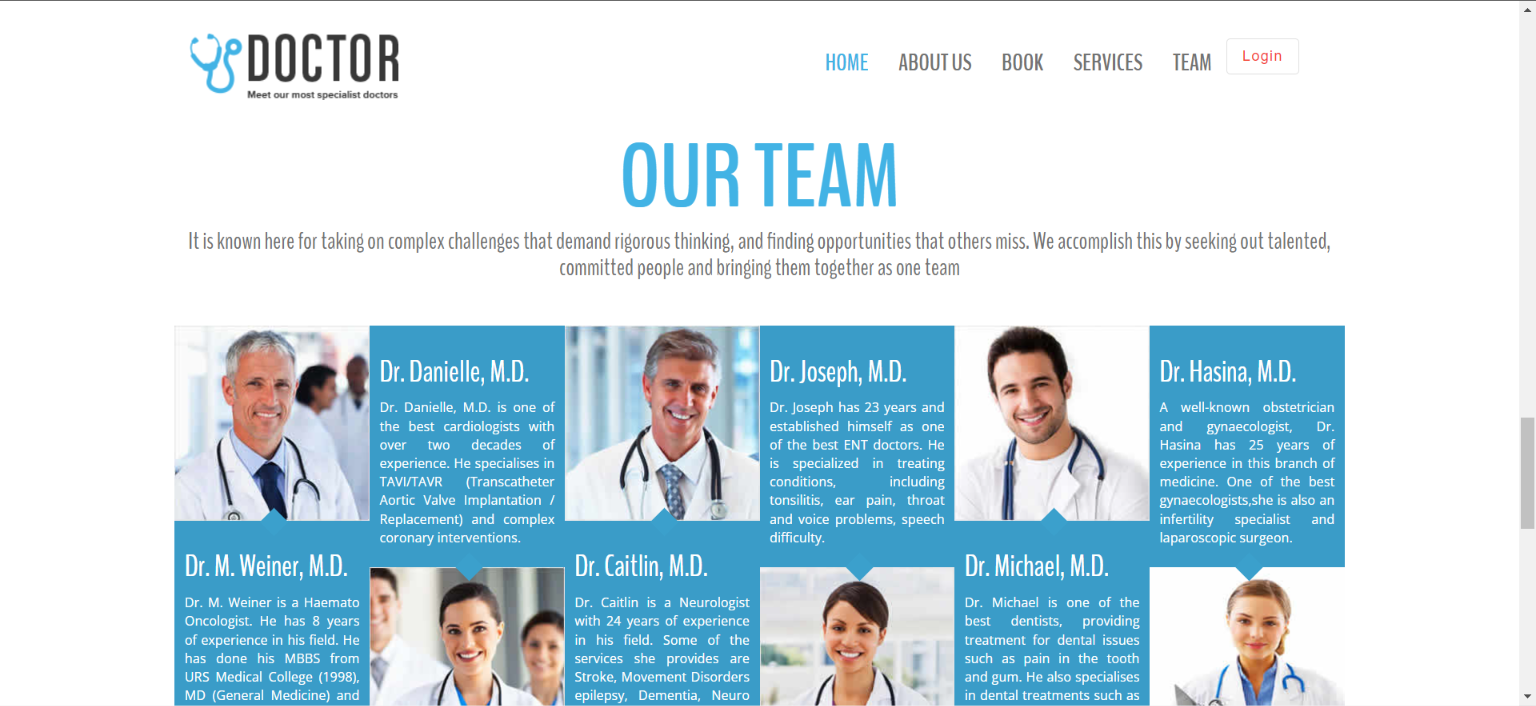
This section consists of different services provided by the particular hospital and can use the services wisely with an appointment.



**Figure 5: Services**

# 4.6 TEAM

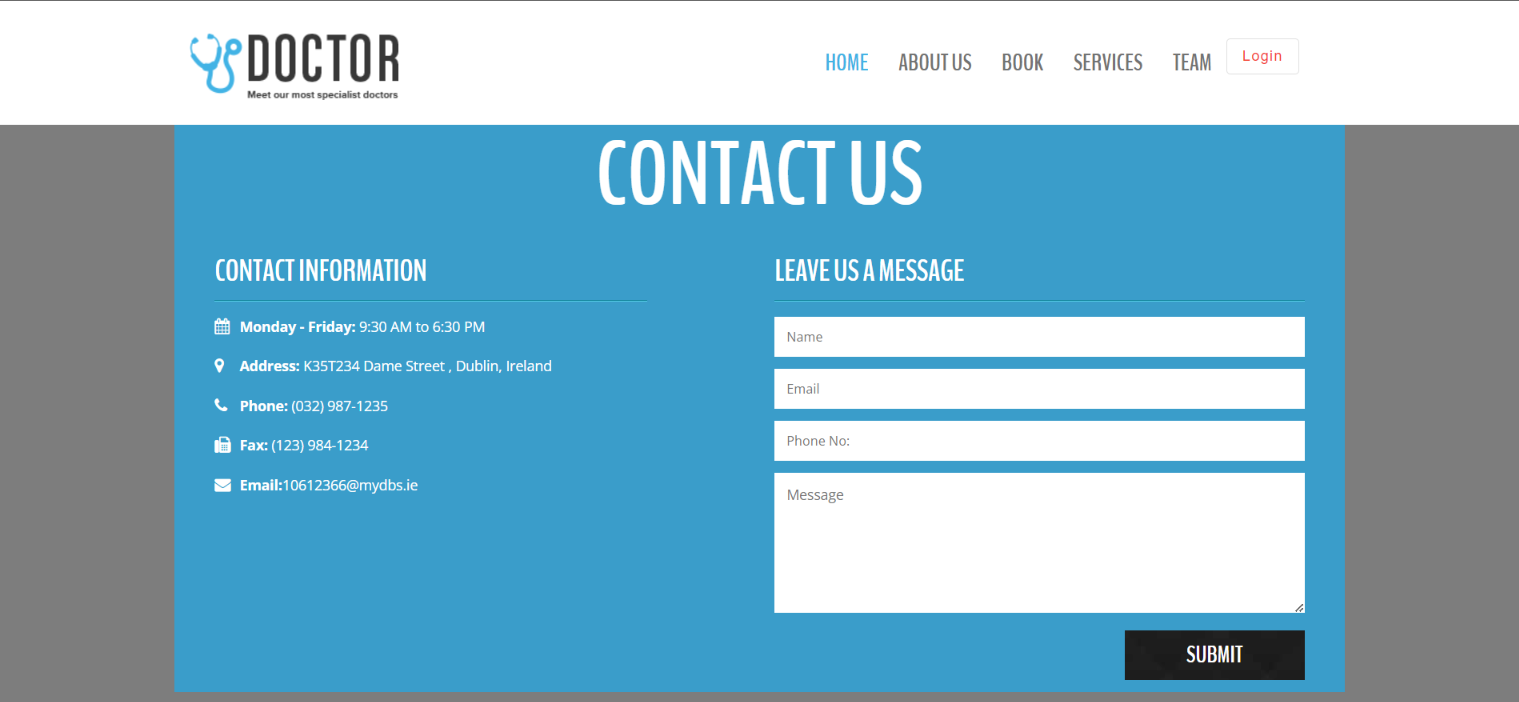
This section consists of a team of doctors and specialists working as part of the hospitality and on different diseases.



**Figure 6: Team**

# 4.7 CONTACT US

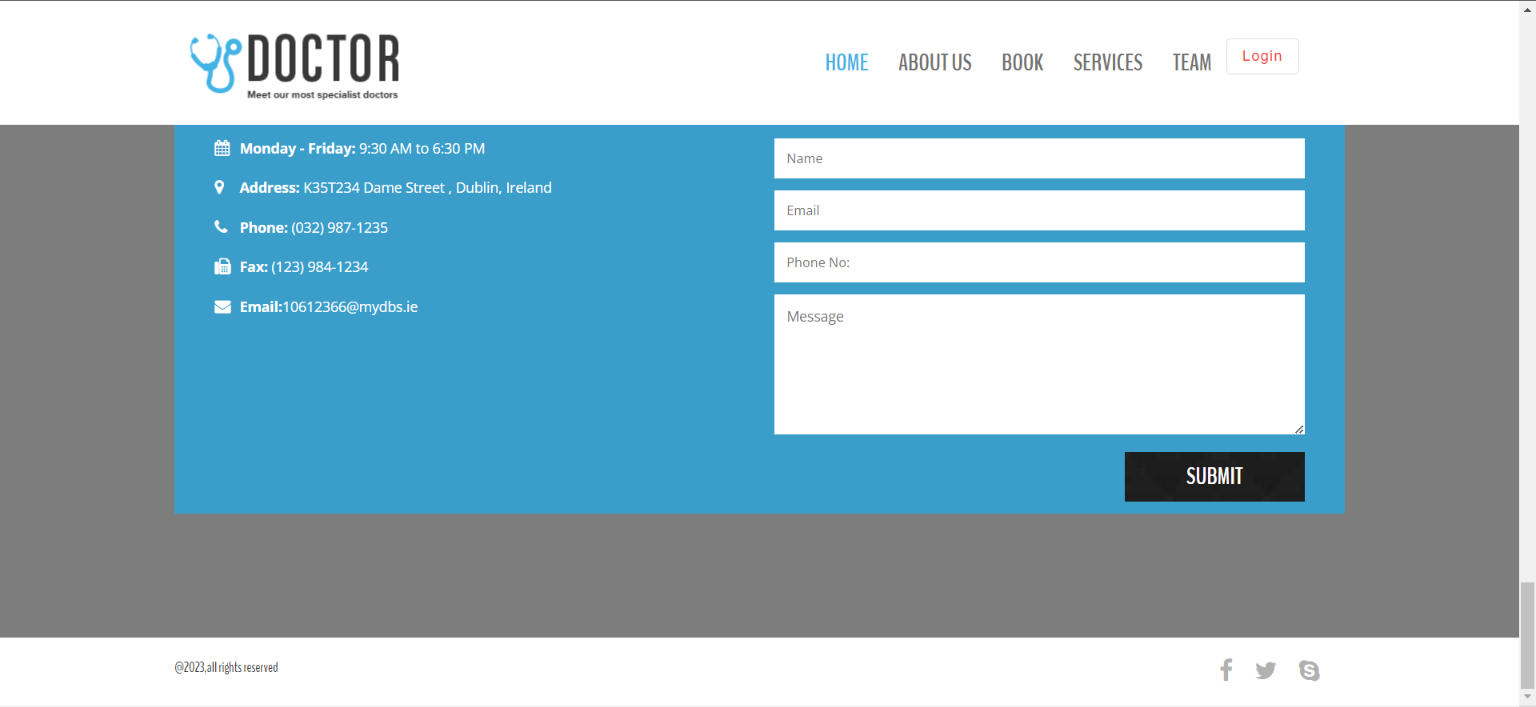
This section has the contact details for any inquiry and other details



**Figure 7: Contact Us**

# 4.8 FOOTER

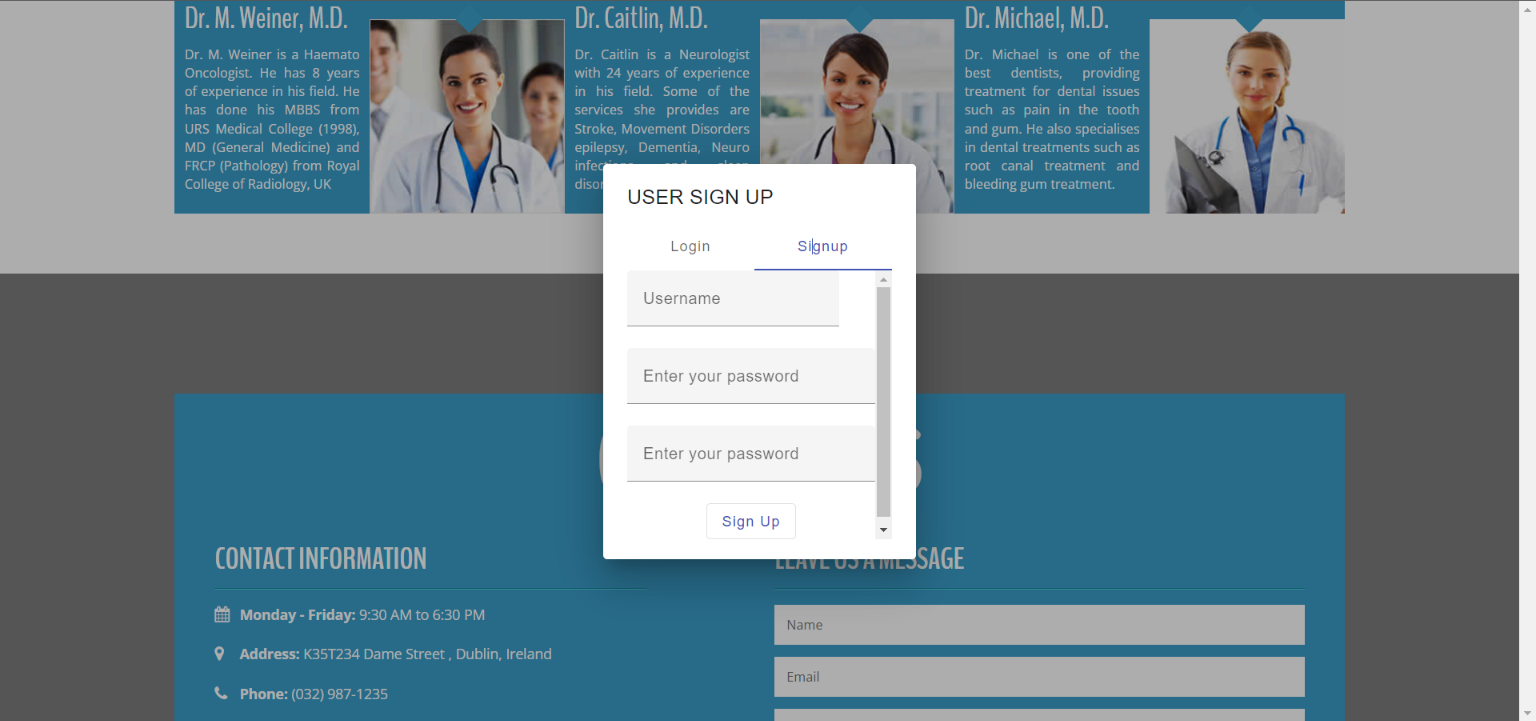
The footer section consists of the social media and the copyrights reserved.



**Figure 8: Footer**

# 4.9 SIGNUP

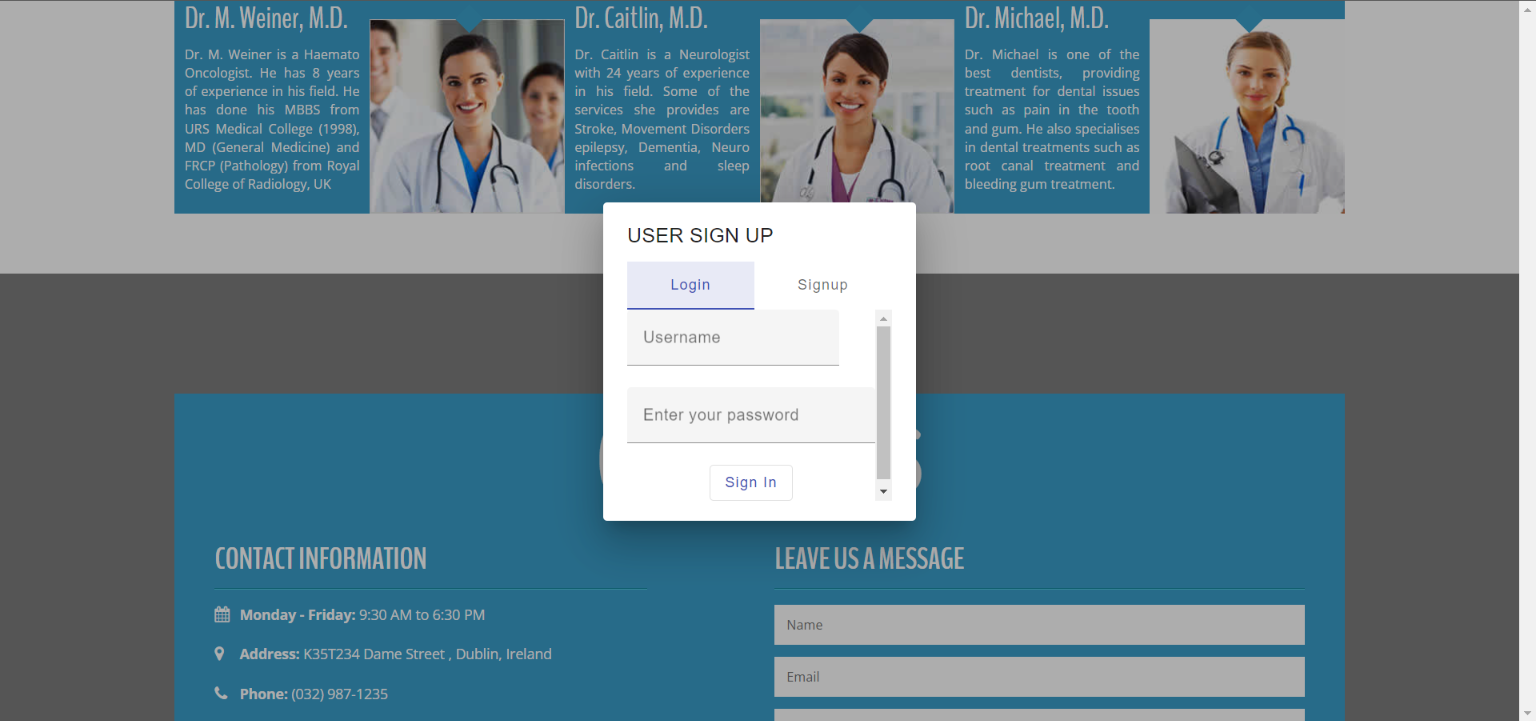
The footer section consists of the signup username and password to get into the application.



**Figure 9: SignUp Page**

# 4.10 LOGIN

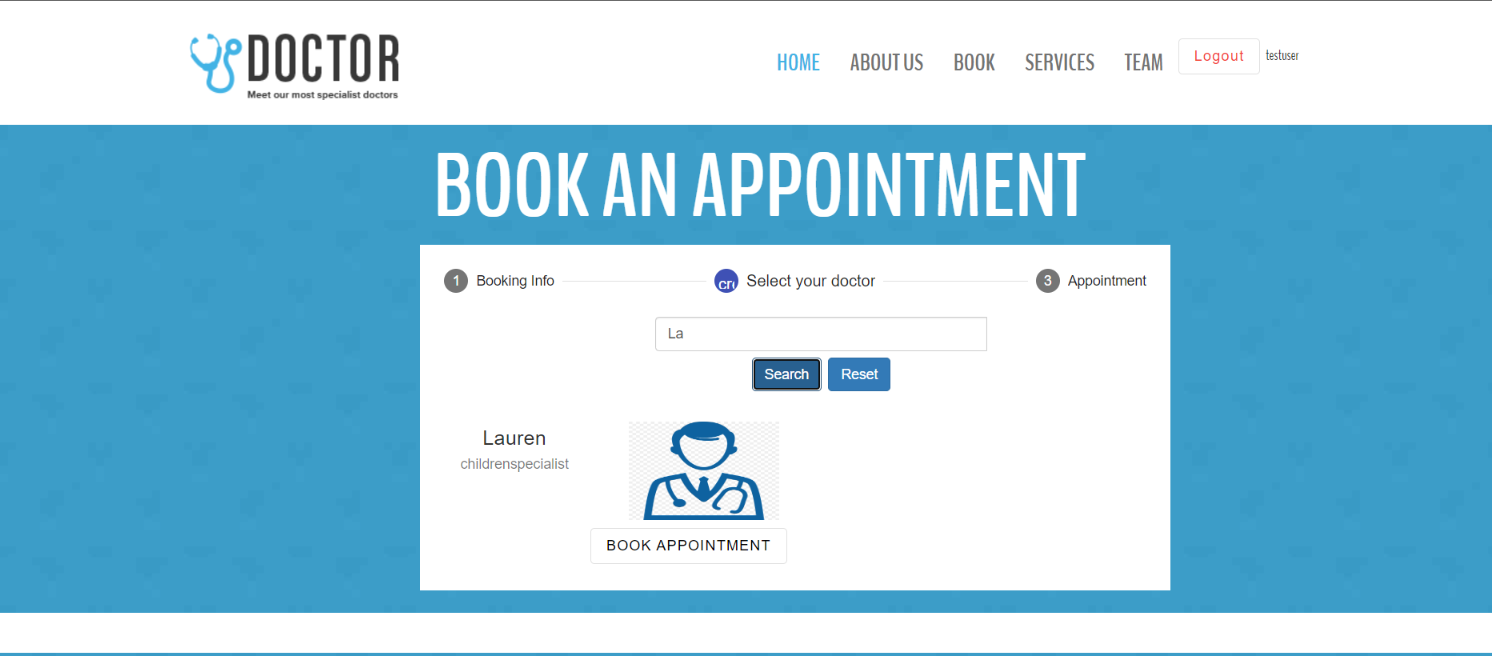
Users can be able to login with the registered details in the below screen.



**Figure 10: Login Modal Popup**

# 4.11 SEARCH PAGE

The section consists of the search option to select any doctor to search on the availability.



**Figure 11: Search**

# 5. BACKEND

# 5.1 DATABASE SETTING

As the below screenshot shows the engine used was a sqllite3 for the database. In the settings.py file, the database configuration has been developed.

**Figure 12: SQL Lite 3 setting in the py setting file**

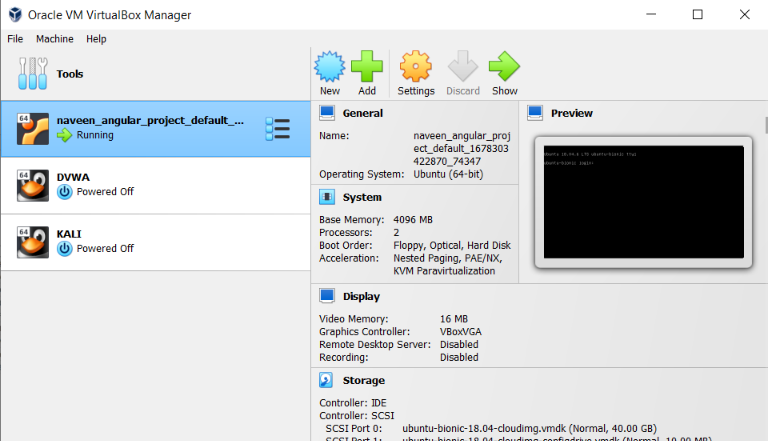
Below is the XML file of the connection of the database which is the sql lite 3



**Figure 13: Database XML File**

# 5.2 LINUX SETTING

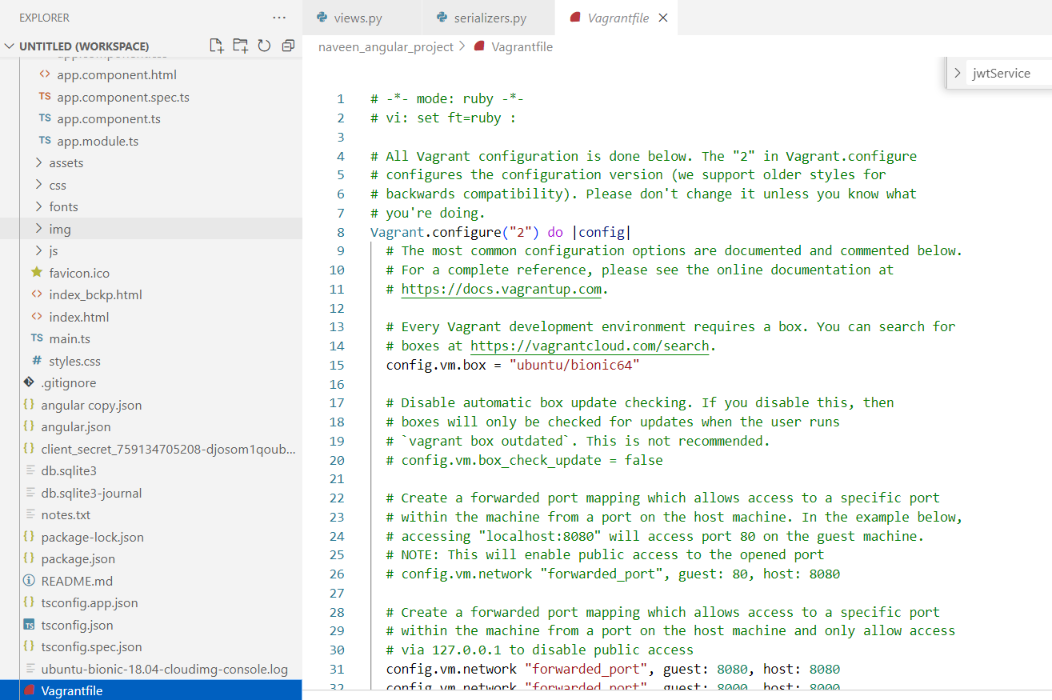
Created the Ubuntu Linux to run the code through the vagrant file



**Figure 14: Linux Virtual Box**

# 5.3 VAGRANT FILE

The below vagrant file is used to image the actual Windows angular, and python code to the Linux Ubuntu and then can be used properly during the deployment of the Amazon EC2.

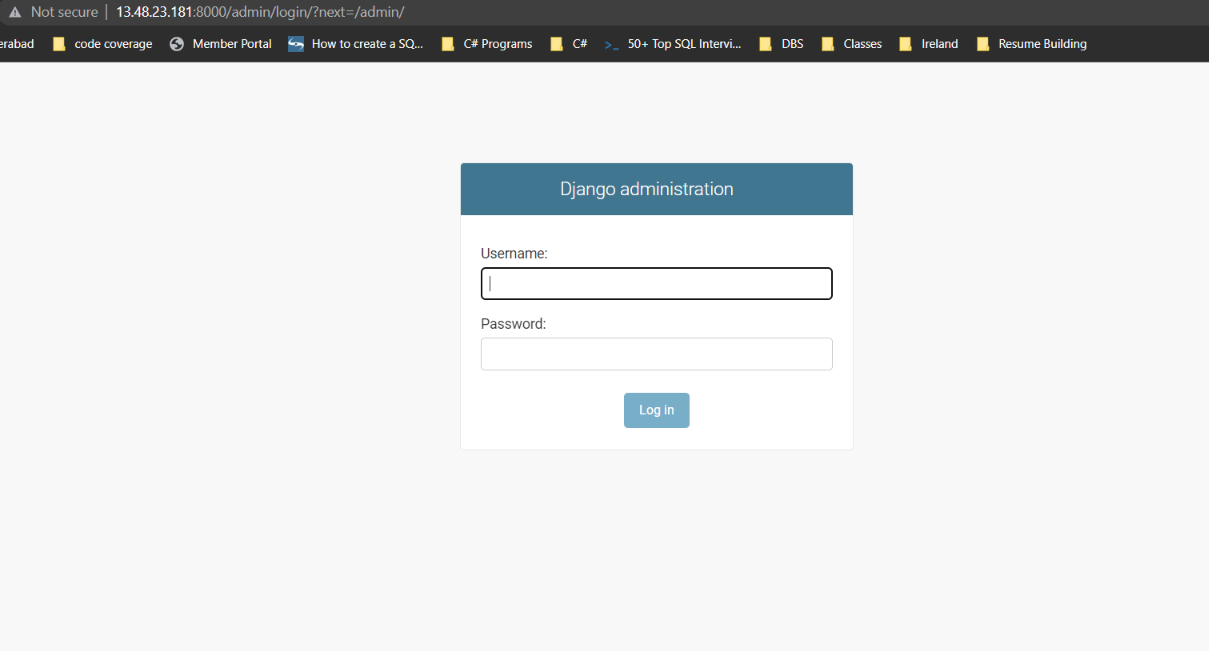


**Figure 15: Vagrant File for Imaging**

# 5.4 DJANGO ADMIN PAGE

In this, data such as the users, specialists, and doctors can be viewed, edited, and deleted as necessary.

Below is the screenshot from the deployed version of the Django

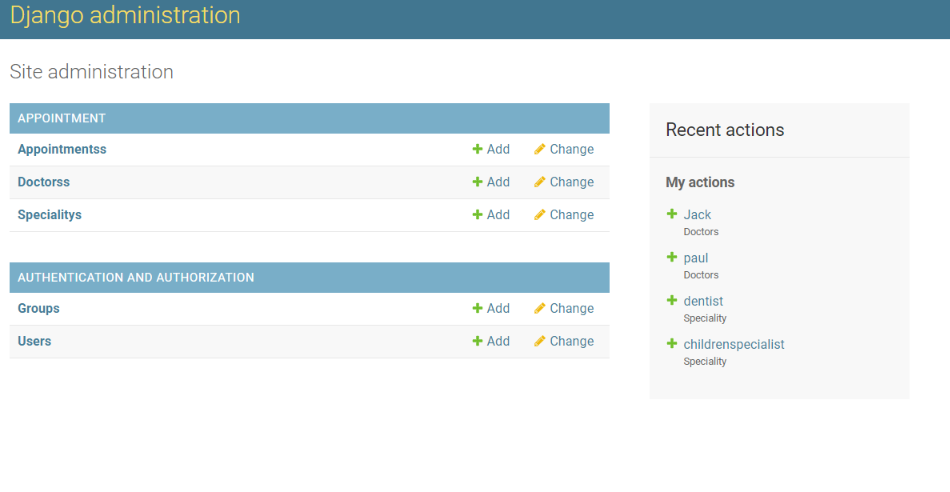


**Figure 16: Admin Login Page.**

Use the provided link to access the admin page - “http://13.48.23.181:8000/admin/login/?next=/admin/”. The login credentials for local SQLLite3 are the same as the username and password.

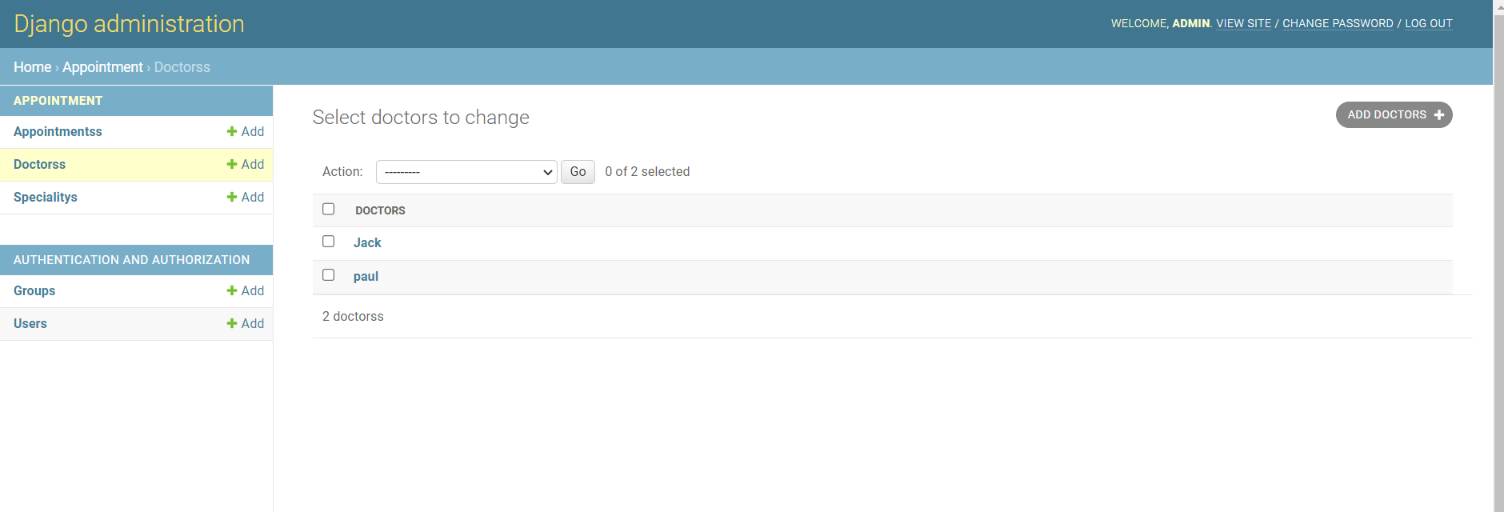
**SuperUser Credentials:**

**Username**: admin, **Password**: password

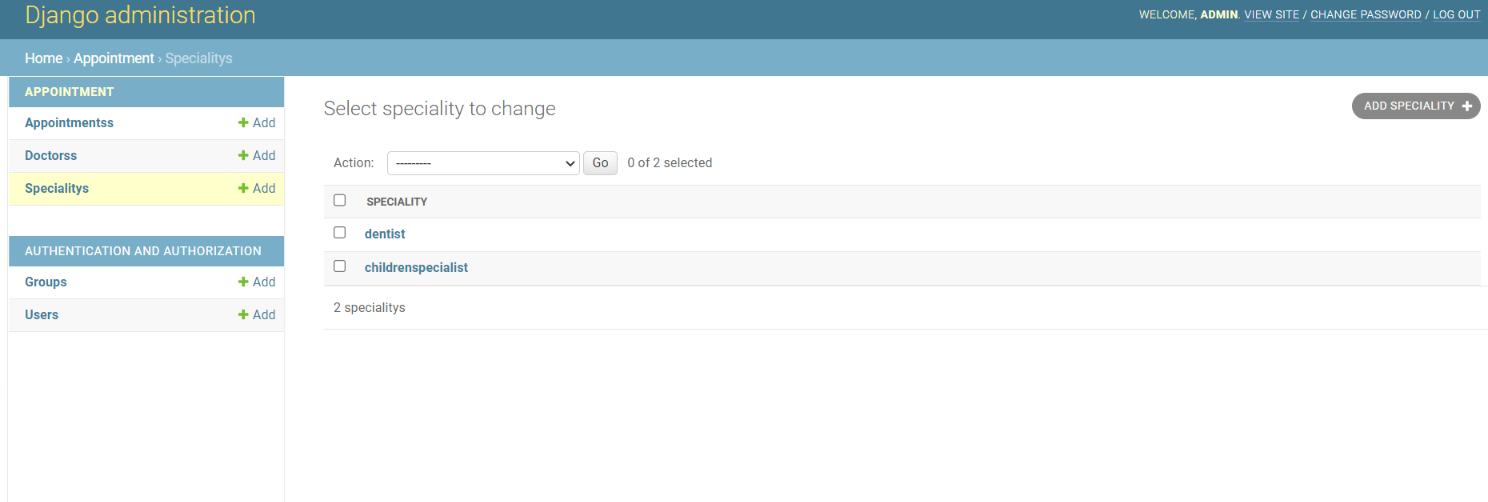


**Figure 17: Django Administration**

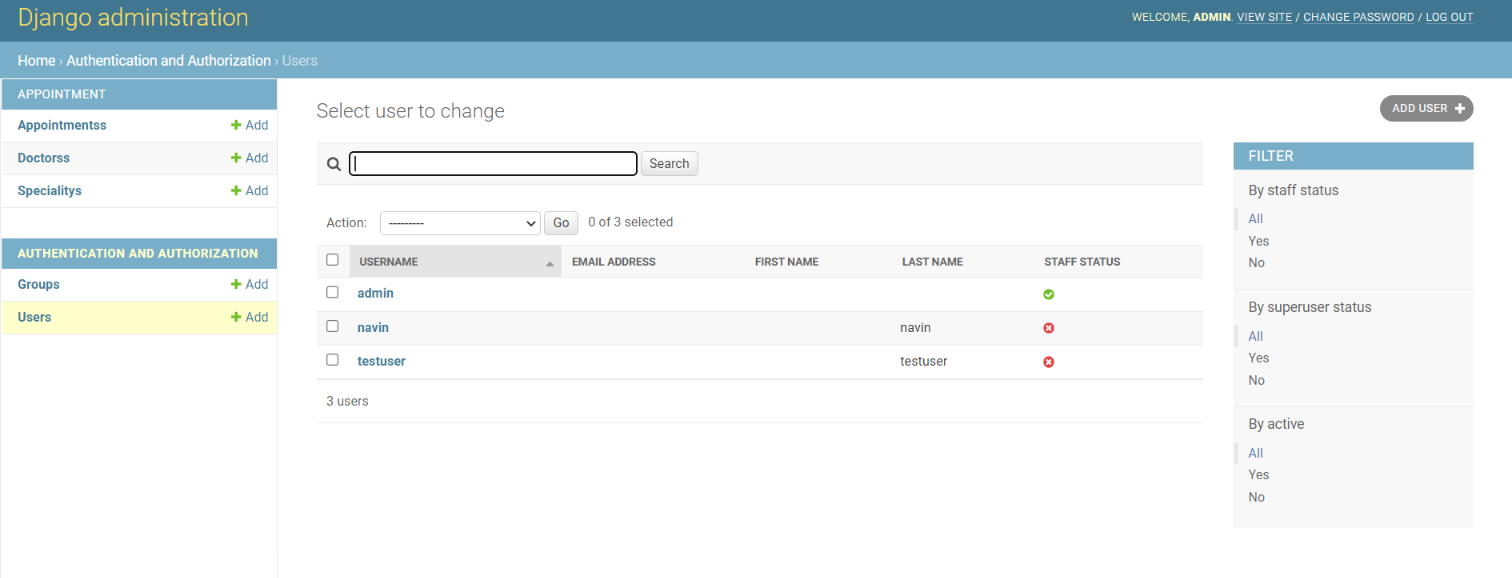
The administrator can view all the data and add any new data that is required for the web application on this page.



**Figure 18: Adding the Doctors**



**Figure 19: Adding the Specialists**



**Figure 20: Adding the Users**

# 6. WEB SERVICES

There is a specific route linked with each screen. Let's examine the APIS list.



**Figure 21: API Services URL**

# 6.1 SIGNUP SCREEN

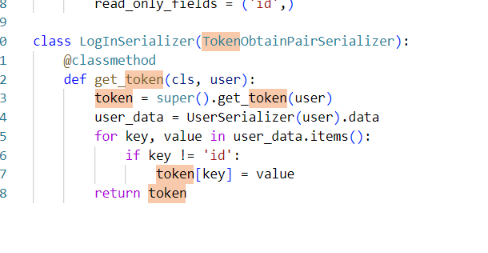
Users can register themselves on the sign up page so they can access the website for scheduling a doctor's appointment.

The system ensures that the password and valid email id are created when the user creates the credentials. The password is saved in an encrypted format during storage to ensure that the user's security is not compromised and that no one else can see the password.



**Figure 22: SignUp Logic**

Instead of keeping the original password, the generate password bearer function encrypts the password and stores the encrypted password in the database.





**Figure 23: Token-based Logic**

# 6.2 LOGIN SCREEN

The user must have a working email address and password to access the login page. Since both fields are required, the user must input accurate and legitimate credentials.



**Figure 24: Login JWT Authentication**

# 6.3 LIST OF DOCTORS VIEW



**Figure 25: Form Submission for Doctor Appointment**

# 6.4 Book Appointment Backend



**Figure 26: Book Appointment Logic**

# 6.5 Get Appointment Backend



**Figure 27: Get Appointment Logic**

# 6.6 Delete Appointment Backend



**Figure 28: Delete Appointment Logic**

# 7. SECURITY AND MEASURES

# 7.1 JWT BEARER AUTHENTICATION

An open standard (RFC 7519) called JSON Web Token (JWT) is used to safely transport data between parties in the form of JSON objects. It is small, readable, and digitally signed by the Identity Provider using a private key or a public key pair (IdP). As a result, the token's legitimacy and integrity can be confirmed by other parties. Using JWT serves to assure data authenticity rather than to conceal data. JWT is not encrypted; it is signed and encoded.

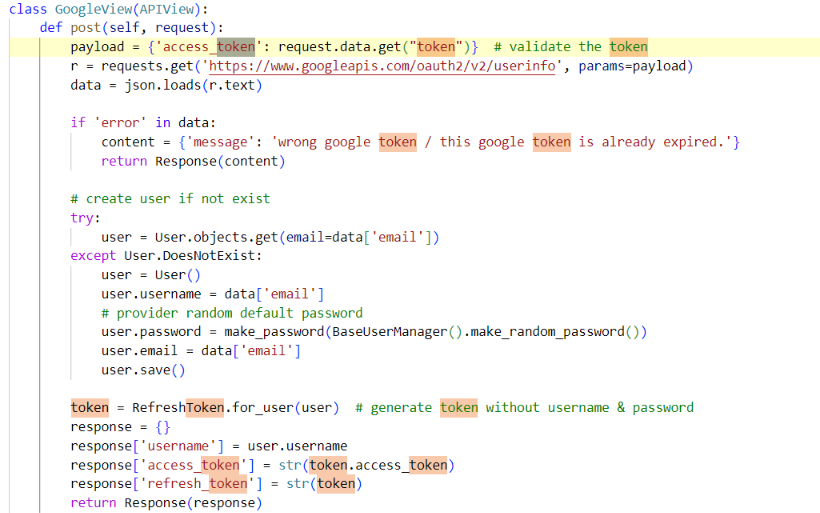
JWT is a stateless, token-based authentication method. As the session is client-side based and stateless, the server does not entirely rely on a datastore (database) to store session data.



**Figure 29: JWT Settings**



**Figure 30: Tokenization**

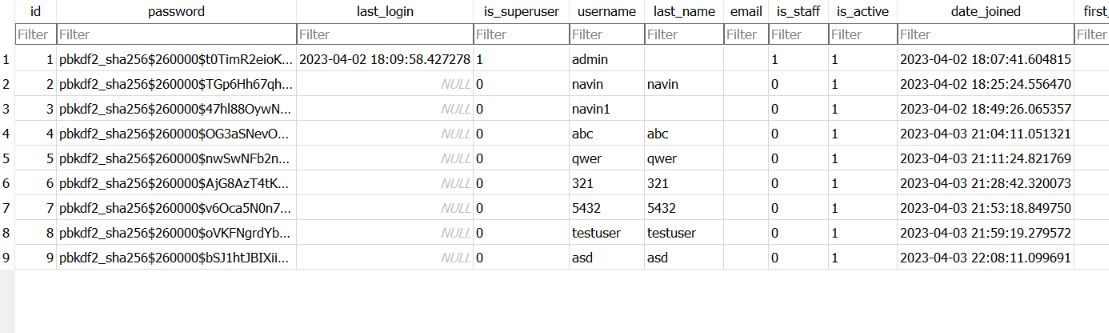


**Figure 31: Google API OAuth Tokenisation**

Below is the screenshot which indicates the bearer-based authentication that is used for the web services and the code below is for the submission of the form.



**Figure 32: Tokenisation Used for Api Services**



**Figure 33: Login Hash**

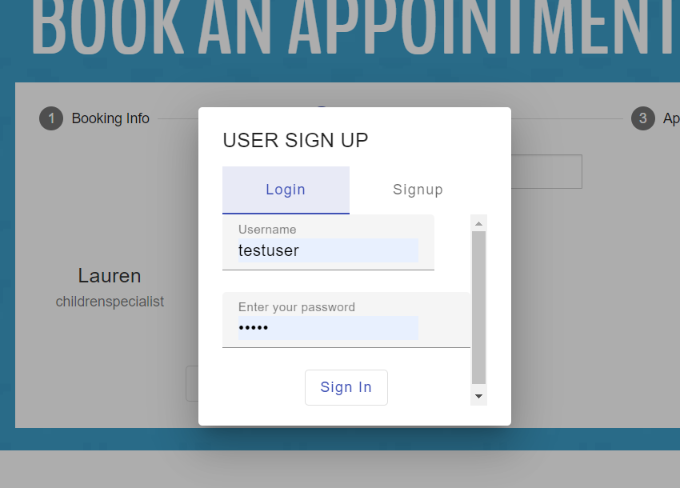
# 8. WEB APPLICATION FEATURES

**Login and Logout Feature**:

Every page of the web application's navigation bar features a login and logout button. By pressing this button, the user can sign up for an account.



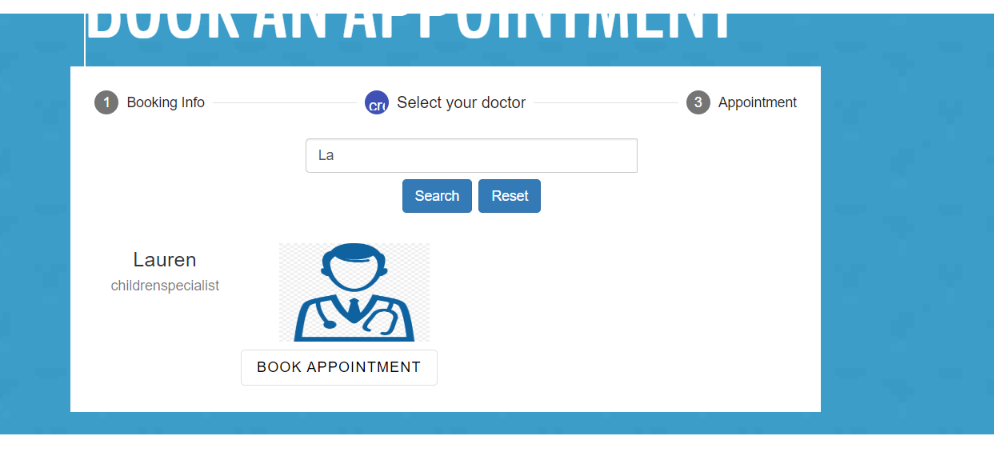
**Figure 34: Login and Logout Feature**



**Figure 35: Login and Register popup**

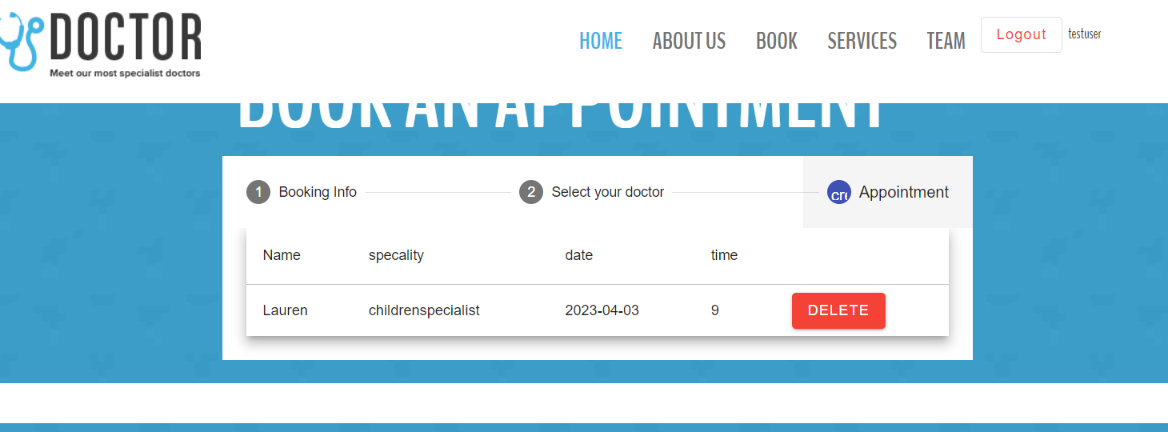
**Search Feature**:

This application consists of search functionality which can be navigated through the book appointment section while filtering the data of the doctors and the user can look through all of the physicians.



**Figure 36: Search Feature**

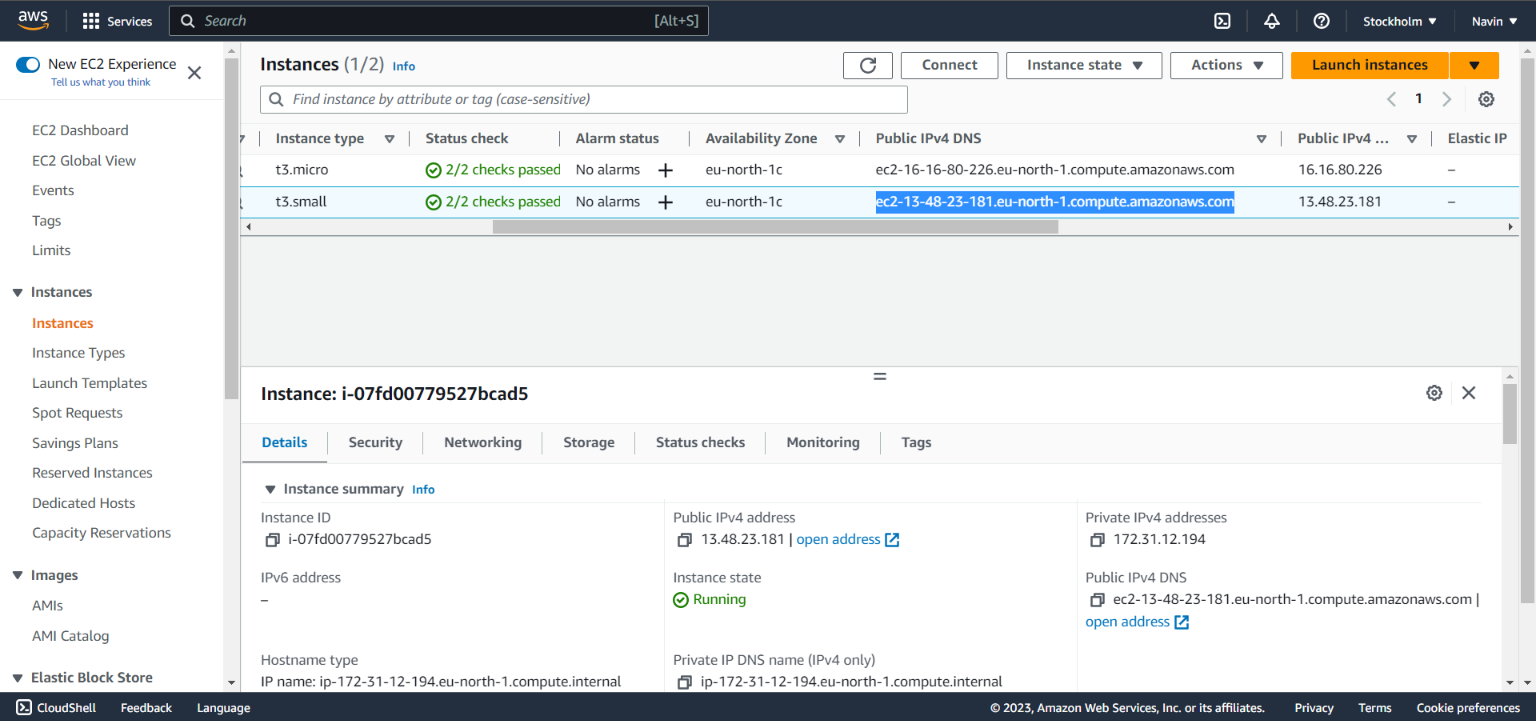
List of Appointments Feature: This is used to view the details using the rest API services



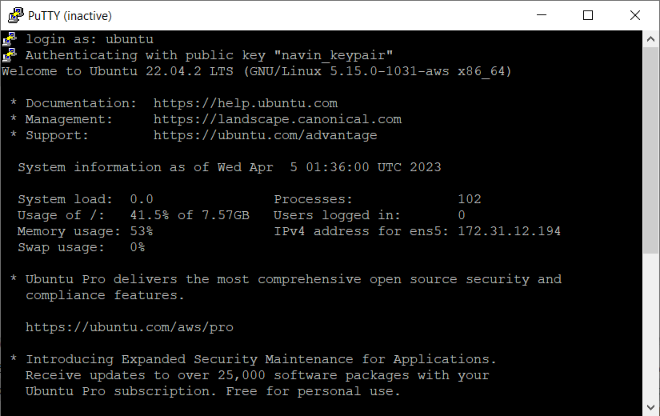
**Figure 37: List Appointment Feature**

# 9. AWS Deployment

In this section, we can see the instance created as part of the deployment in the Amazon EC2



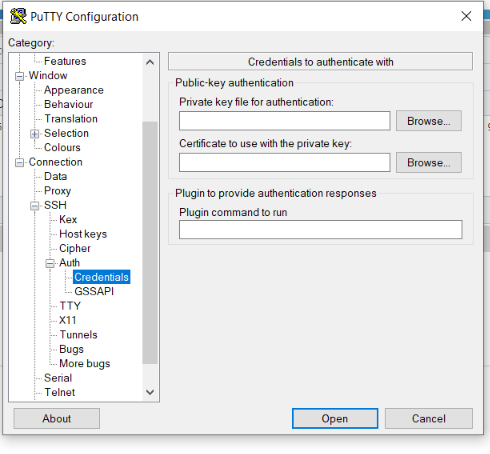
**Figure 38: AWS EC2 Deployment**



**Figure 39: Putty Console deployment**

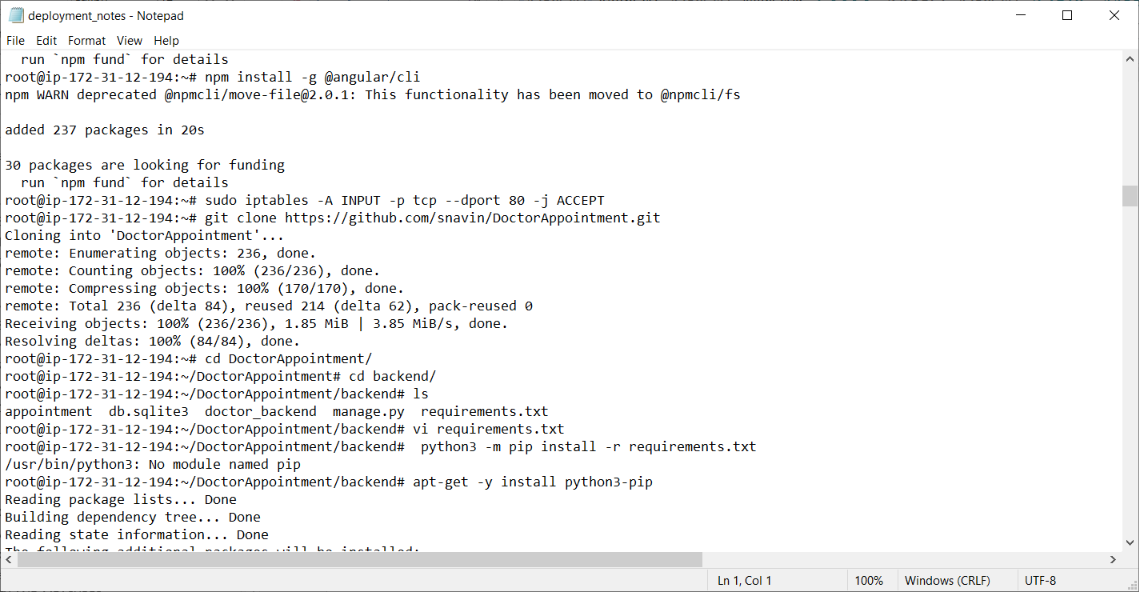
Login as: ubuntu

Authenticating with public key "keypair" added as part of the artifact.



**Figure 40: Putty Configuration**

The deployment notes are added as part of the artifacts folder



**Figure 41: Commands used for Deployment via Putty**

# 10. LINKS

GitHub Link: https://github.com/snavin/DoctorAppointment.git

AWS Link: <http://13.48.23.181/>

DJANGO Admin Link: http://13.48.23.181:8000/admin/login/?next=/admin/

# 11. REFERENCES:

1. T Haerder and A Reuter, "Principles of transaction-oriented database recovery", *CSUR*, vol. 15, pp. 287-317, Apr. 1983.

2. F Ramisch and M Rieger, "Recovery of SQLite3 Data Using Expired Indexes", 9th International Conference on IT Security Incident Management & IT Forensics, pp. 147-152, Nov. 2015.

3. MDB - Material Design for Bootstrap 5 & 4 (no date). Available at: <https://mdbootstrap.com/> (Accessed: 5 April 2023).

4. Contributors, M.O., Jacob Thornton, and Bootstrap (no date) Accordion. Available at: <https://getbootstrap.com/docs/5.2/components/accordion/> (Accessed: 5 April 2023).

5. Youtube.com. 2021. How to use the vagrant box to create AWS EC2 instance? (2021). Available at: https://www.youtube.com/watch?v=im-shBCnP2E (Accessed: 5 April 2023).

6. Download PuTTY - a free SSH and telnet client for Windows (no date). Available at: https://www.putty.org/ (Accessed: 5 April 2023).

7. Python Release Python 3.11.0 (no date) Python.org. Available at: https://www.python.org/downloads/release/python-3110/ (Accessed: 5 April 2023).

8. Secure and resizable cloud computing – Amazon EC2 – Amazon Web Services (no date). Available at: https://aws.amazon.com/ec2/ (Accessed: 5 April 2023).

9. Team, A.C. (no date) Angular Material, Angular Material. Available at: https://material.angular.io/ (Accessed: 5 April 2023).

10. Free CSS | 3472 Free Website Templates, CSS Templates, and Open Source Templates (no date). Available at: https://www.free-css.com/ (Accessed: 5 April 2023).