**PROJECT REPORT OF INDUSTRY ORIENTED HANDS-ON EXPERIENCE (IOHE)**

**ON**

**MediSphereX**

**submitted in partial fulfilment of the requirements for the award of degree of**

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

|  |  |
| --- | --- |
| **Submitted By:** | **Supervised By:** |
| **Avinash Mehta(2010990142)** | **Mr.Abhishek Bhardwaj** |
| **Harshdeep Singh(2010990280)** |  |
| **Lakshay Garg(2010991606)** |  |
|  |  |

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CHITKARA UNIVERSITY**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY, RAJPURA, PUNJAB, INDIA**

**CONTENTS**

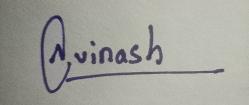
|  |  |  |
| --- | --- | --- |
| S.No. | Title | Page No. |
|  | Declaration | iii |
|  | Acknowledgement | iv |
|  | List of Figures and Tables | v |
|  | Abstract | vi |
| 1 | Introduction | 1-2 |
| 2 | Methodology | 3-7 |
| 3 | Tools and Technologies | 8 |
| 4 | Implementation | 9-29 |
| 5 | Major Findings/Outcomes/Output/Results | 30-37 |
| 6 | Conclusion and Future Scope | 37 |
|  | References | 38 |
|  |  |  |

**DECLARATION**

We hereby declare that the project work titled, MediSphereX submitted as part of Bachelor’s degree in CSE, at Chitkara University, Punjab, is an authentic record of our own work carried out under the supervision of Mr. Abhishek Bhardwaj .

**Signature(s):**

**Avinash Mehta**

****

**ACKNOWLEDGEMENT**

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing me an opportunity to undergo this Project as the part of the curriculum.

We are thankful to Mr. Abhishek Bhardwaj for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.He provided his valuable suggestions and precious time in accomplishing our Integrated project report.

Lastly, We would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improve our quality of work.

|  |  |  |
| --- | --- | --- |
| **Harshdeep Singh** | **Avinash Mehta** | **Lakshay Garg** |
| **ID No 2010990280** | **ID No 2010990142** | **ID No 2010991606** |

**LIST OF FIGURES AND TABLES**

|  |  |
| --- | --- |
| **Name** | **Page No.** |
| **Table 1.1 User Details** | **5** |
| **Table 1.2 Appointment** | **5-6** |
| **Table 1.3 Doctor** | **6-7** |
| **Table 1.4 Speciality** | **7** |

**ABSTRACT**

The purpose of the project entitled "MediSphereX" is to computerize the Front Office Management of a hospital by developing user-friendly, simple, fast, and cost-effective software. It deals with the collection of patient information, diagnosis details, and other relevant data, which was traditionally done manually.

The main functions of the system are to register and store patient details, doctor details, and retrieve this information as and when required. It also allows for the meaningful manipulation of these details.

The system input includes patient details and diagnosis information, while the system output displays these details on the screen. The Hospital Management System can be accessed using a username and password, either by an administrator or receptionist. Only they can add data into the database, which can be easily retrieved. The data is well-protected for personal use, making the data processing very fast.

**INTRODUCTION**

The **MediSphereX** project is a hospital management system that includes patient registration, storage of patient details, and computerized appointment booking. The software assigns a unique ID to every patient and automatically stores the details of patients and staff. It includes a search feature to check the current status of each patient appointment. Users can search patient details using the unique patient ID.

The Hospital Management System can be accessed using a username and password, either by an administrator or receptionist. Only they can add data into the database, which can be easily retrieved. The interface is user-friendly, and the data is well-protected for personal use, making the data processing very fast.

The Hospital Management System is a powerful, flexible, and user-friendly solution designed to provide tangible benefits to hospitals. It is an integrated end-to-end system that covers a wide range of hospital administration and management processes, providing relevant information across the hospital to support effective decision-making for patient care, hospital administration, and critical financial accounting.

The Hospital Management System is designed for multi-specialty hospitals to improve the quality and management of hospital operations, including clinical process analysis and activity-based costing. It enables organizations to develop and improve their effectiveness and efficiency.

**1.1 Problem Introduction:**

**Lack of immediate retrievals:** - The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient’s history, the user has to go through various registers. This results in in convenience and wastage of time.

**Lack of immediate information storage**: - The information generated by various transactions takes time and efforts to be stored at right place.

**Lack of prompt updating: -** Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

**Error prone manual calculation:** - Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient’s bill based on various treatments.

**Preparation of accurate and prompt reports:** - This becomes a difficult task as information is difficult to collect from various register.

**1.2 Goals:**

* User friendly
* Simple fast
* Low cost and effective
* It deals with the collection of patient’s information
* Diagnosis

**1.3 Objectives**

The primary objectives of the MediSphereX hospital management system are:

* To provide a comprehensive and integrated platform for managing various aspects of hospital operations, including patient records, appointment scheduling, inventory control, and billing.
* To enhance the overall patient experience by enabling efficient communication, reducing wait times, and improving the delivery of care.
* To empower healthcare professionals with intuitive and user-friendly tools that optimize operational efficiency and facilitate the provision of high-quality care.
* To leverage modern web technologies, database management, and the Spring MVC framework to create a scalable and adaptable solution that meets the evolving needs of healthcare institutions.
* To facilitate seamless integration with existing hospital information systems and enable efficient data management through the utilization of Hibernate.
* Ensure compatibility with mobile devices to allow healthcare providers and patients to access the system on-the-go, enhancing convenience and accessibility.
* Implement robust security measures to protect patient data against unauthorized access or breaches.

**METHODOLOGY**

**Requirement Analysis**: The first phase involves a comprehensive analysis of the requirements of healthcare facilities, including hospitals, clinics, and medical centers. This involves gathering input from healthcare professionals, administrators, and IT staff to understand the specific needs and challenges faced in managing hospital operations.

**System Design**: Based on the gathered requirements, the system design phase focuses on creating a detailed blueprint for the MediSphereX hospital management system. This includes designing the database schema using MySQL, creating wire frames for the user interface, and planning the architecture of the system using Java, Spring MVC, and Hibernate.

**Frontend Development**: The frontend development phase involves the implementation of the user interface using HTML, CSS, JavaScript, and Bootstrap. This includes creating intuitive and responsive interfaces for patient registration, appointment scheduling, billing, and other user interactions.

**Backend Development**: Simultaneously, the backend development phase focuses on implementing the server-side logic using Java and the Spring MVC framework. This includes developing the business logic, data processing, and integration with the MySQL database using Hibernate for efficient data management.

**Integration and Testing**: Once the frontend and backend components are developed, they are integrated to form a cohesive system. Rigorous testing is then conducted to ensure the functionality, performance, and security of the MediSphereX system. This includes unit testing, integration testing, and user acceptance testing to validate the system against the initial requirements.

**Deployment and Training**: After successful testing, the MediSphereX hospital management system will be deployed in the healthcare facilities. Training sessions will be conducted for the staff to familiarize them with the system's features and functionalities, ensuring a smooth transition to the new system.

**Maintenance and Support**: Post-deployment, ongoing maintenance and support are provided to address any issues, implement updates, and incorporate feedback from users. This ensures the continued reliability and effectiveness of the MediSphereX system in meeting the evolving needs of healthcare institutions.

The entire project mainly consists of:

1. **User Module(Patient)**
2. **Admin Module**
3. **Doctor Module(Staff)**

**User Module:**

The user module consists of various functionalities for the user (patient) such as navigating throughout website of the hospital where the user can find all the details related to the hospital.The user module consists of:

1. About section
2. Doctor’s Info section
3. Departments section
4. Pricing section
5. Blog section
6. User Registration/Login section
7. Book Appointment section

**Admin Module:**

The admin module allows the hospital staff to manage the information of patients,doctors,appointments,etc.It can be accessed using unique login id and password.

It consists of:

1. [ADD-DOCTOR](http://localhost:8090/Dcare/addDoctor.html)
2. VIEW-DOCTOR
3. VIEW PATIENT
4. UPDATE DOCTOR
5. DELETE DOCTOR
6. TOTAL NO. OF USERS
7. TOTAL APPOINTMENTS BOOKED
8. ADD SPECIALIST DETAILS

**Doctor Module:**

The doctor module consists of doctor login page where the doctor will be assigned a unique id and password by the admin.The doctor can then login using the id, password and view the doctor’s dashboard which consists of the appointments booked to that particular doctor and can comment on the patient details and treatment given.

**Database Analyzing, design and implementation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Column Name** | **Data Type** | **Description** |
| 1. | Id | int | Contains user id |
| 2 | full\_name | Varchar(45) | Contains full name of the user |
| 3. | email | Varchar(45) | Contains email id of the user |
|  | password | Varchar(45) | Contains password of the user |

**Table 1.1 User Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Column Name** | **Data Type** | **Description** |
| 1. | Id | int | Contains serial id |
| 2 | user\_id | int | Contains user id of the user |
| 3. | fullname | Varchar(45) | Contains name of the patient |
|  | gender | Varchar(45) | Contains gender of the patient |
|  | age | Varchar(45) | Contains age of the patient |
|  | appointDate | Varchar(45) | Contains the appointment date |
|  | email | Varchar(45) | Contains the email of the patient |
|  | phno | int | Contains the contact of the patient |
|  | diseases | Varchar(45) | Contains the disease acquired by the patient |
|  | doctor\_id | int | Contains the doctor id |
|  | status | Varchar(45) | Contains the status of the patient’s appointment |

**Table 1.2 Appointment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Column Name** | **Data Type** | **Description** |
| 1. | Id | int | Contains id |
| 2 | full\_name | Varchar(45) | Contains full name of the doctor |
| 3. | dob | Varchar(45) | Contains dob of the doctor. |
| 4. | qualification | Varchar(45) | Contains qualification of the doctor |
| 5 | specialist | Varchar(45) | contains speciality of the doctor |
| 6 | email | Varchar(45) | contains email of the doctor |
| 7 | mobno | varchar(10) | contains mobile no. of the doctor |
| 8 | password | Varchar(45) | contains password of thr doctor login id |

**Table 1.3 Doctor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Column Name** | **Data Type** | **Description** |
| 1. | Id | int | Contains id |
| 2 | spec\_name | Varchar(45) | Contains speciality of the doctor. |

**Table 1.4 Speciality**

**TOOLS AND TECHNOLOGIES**

**HTML**: HTML is used to create the structure and content of the web pages within the MediSphereX system. It is responsible for defining the various elements of the user interface, such as forms for patient registration, input fields for medical records, and layout structures for displaying information.

**CSS**: CSS is utilized to style and format the HTML elements, ensuring a consistent and visually appealing presentation of the MediSphereX interface. It is used to define the colors, fonts, spacing, and overall layout of the web pages, enhancing the user experience.

**JavaScript**: JavaScript is employed to add interactivity and dynamic functionality to the MediSphereX system. It facilitates tasks such as form validation, real-time data updates, and interactive user interfaces, enhancing the responsiveness and usability of the application.

**Bootstrap**: Bootstrap is used to leverage its pre-built design templates and components to create a responsive and mobile-friendly user interface for MediSphereX. It provides a framework for building consistent and visually appealing web pages across different devices and screen sizes.

**MySQL**: MySQL serves as the database management system for the MediSphereX system, storing and organizing the structured data related to patient records, appointments, medical history, and other essential information. It enables efficient data storage, retrieval, and management for the hospital management system.

**Java Server Pages (JSP)**: JSP is utilized to create dynamic web pages within the MediSphereX system, allowing for the integration of Java code with HTML to generate dynamic content. It enables the presentation of data-driven information and the execution of server-side logic within the web application.

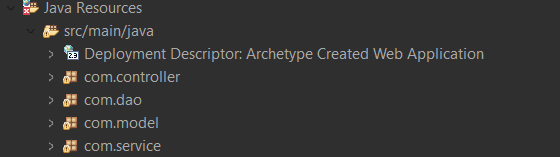
**Java**: Java is used for the server-side development of the MediSphereX system, providing the foundation for implementing business logic, data processing, and system integration. It enables the creation of robust and scalable server-side components for the hospital management system.

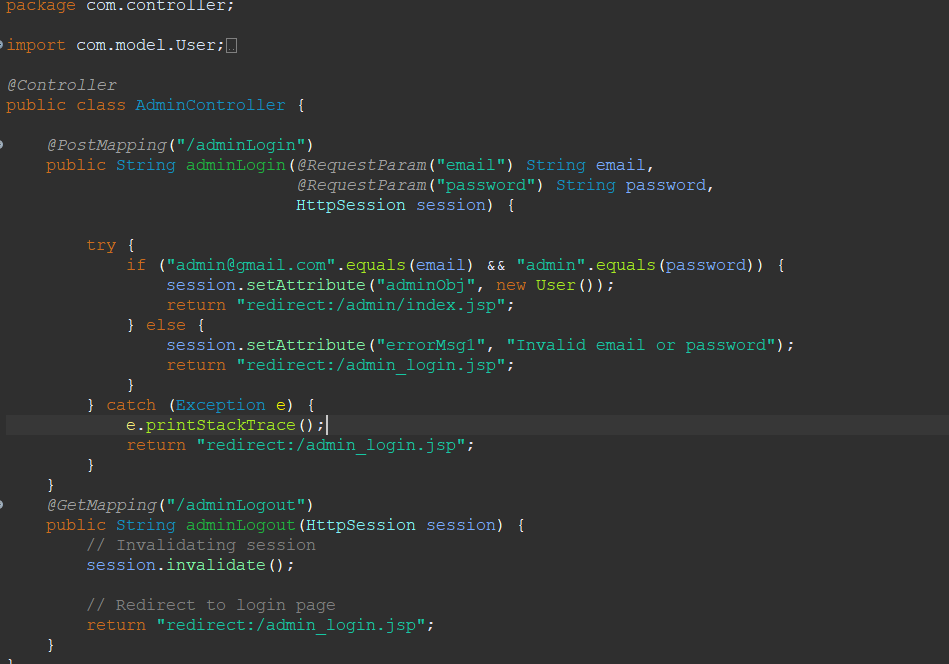
**Spring MVC**: The Spring MVC framework is employed to simplify the development of the server-side components of the MediSphereX system. It provides a robust and scalable architecture for handling web requests, managing the application's business logic, and facilitating the integration with the MySQL database using Hibernate.

**Hibernate**: Hibernate is used as the Object-Relational Mapping (ORM) framework to simplify the interaction between the Java objects and the MySQL database. It enables efficient data management, mapping database tables to Java classes, and providing a seamless way to store, retrieve, and update data.

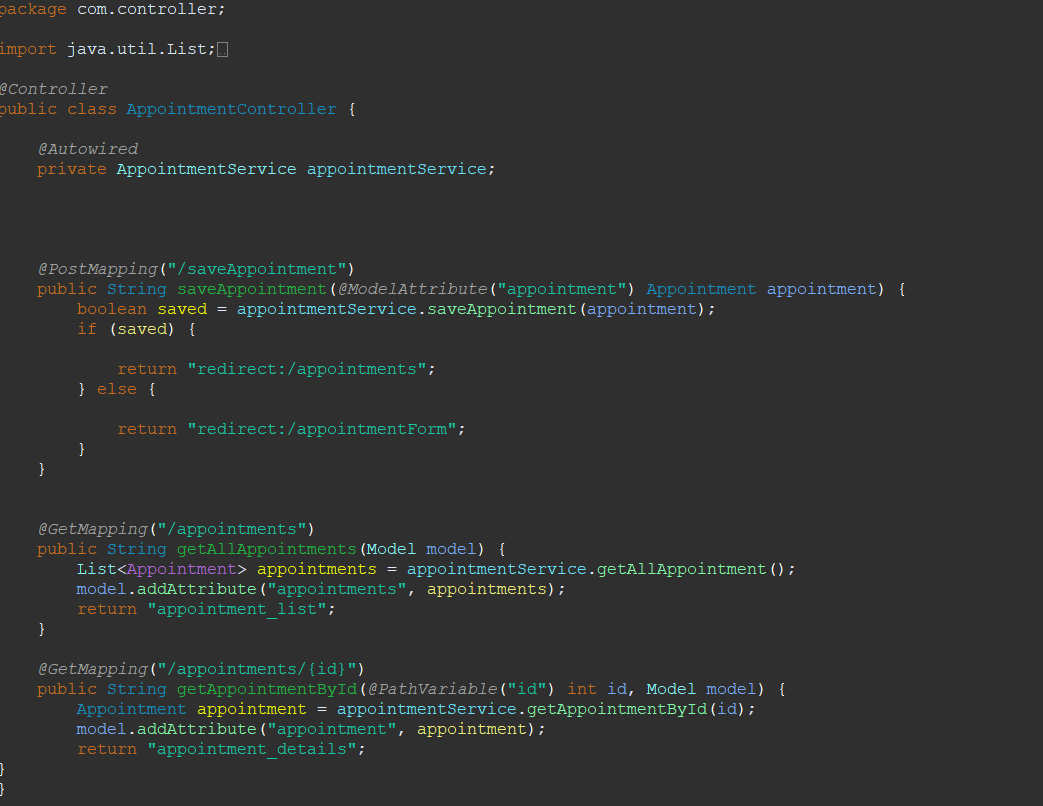
**IMPLEMENTATION**

**Java Resources implemented in the project:**

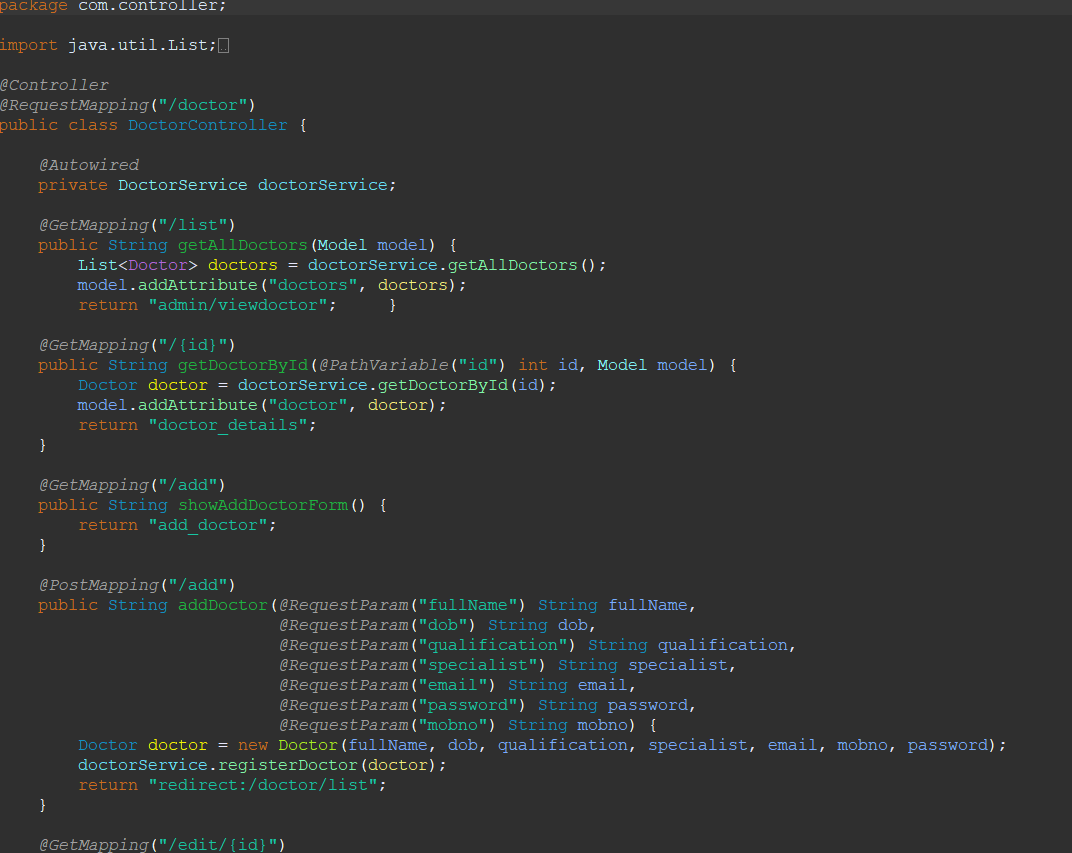


1. **com.Controller**

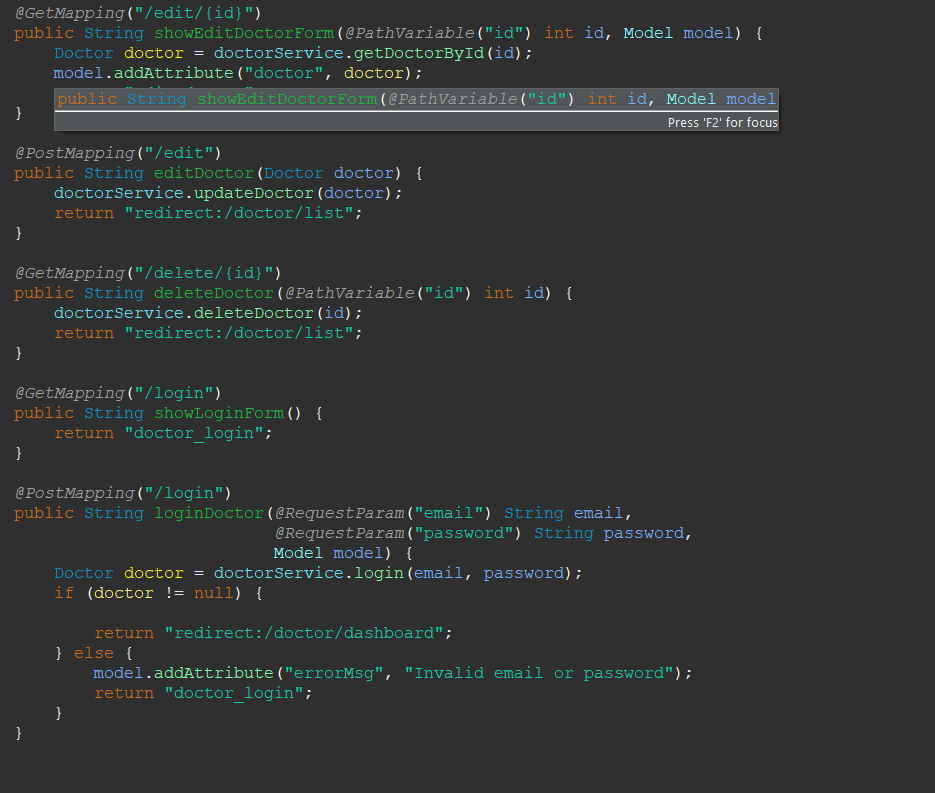
AdminController.java



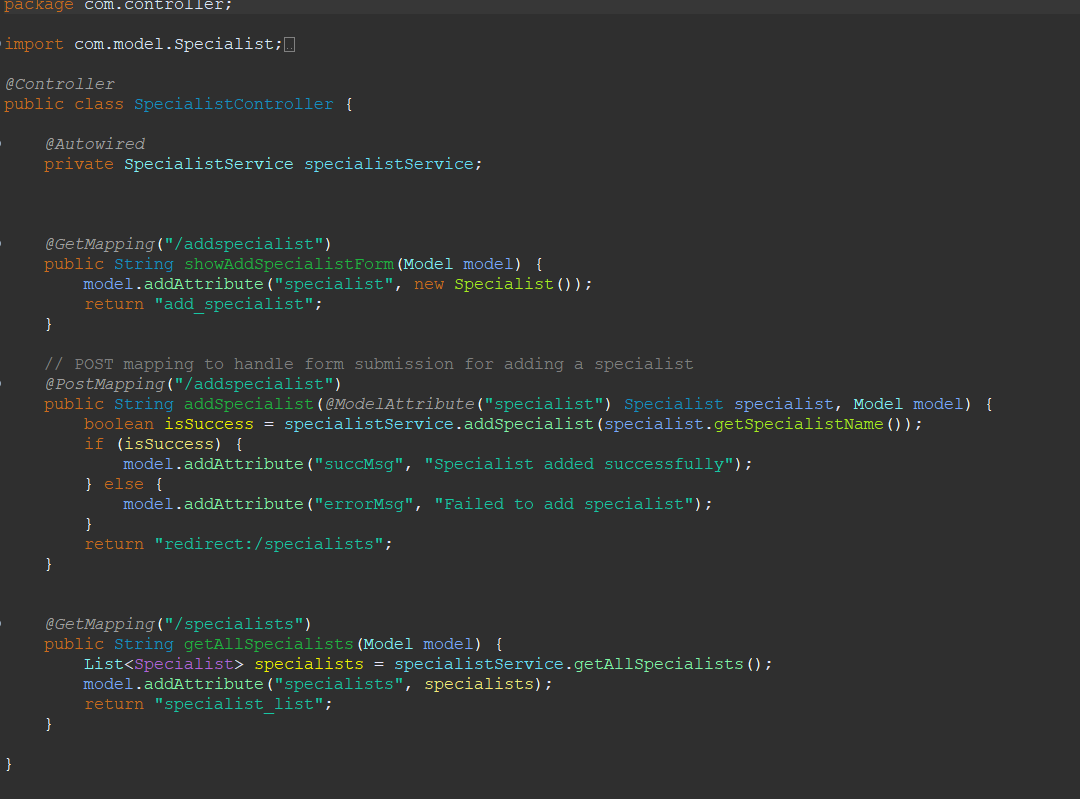
AppointmentController.java



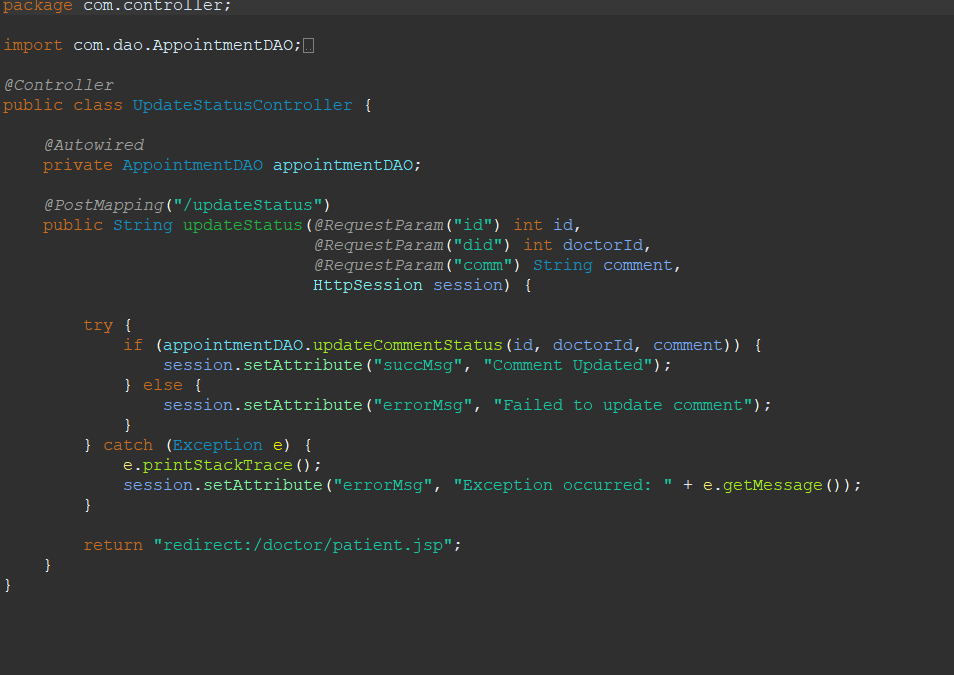
DoctorController.java



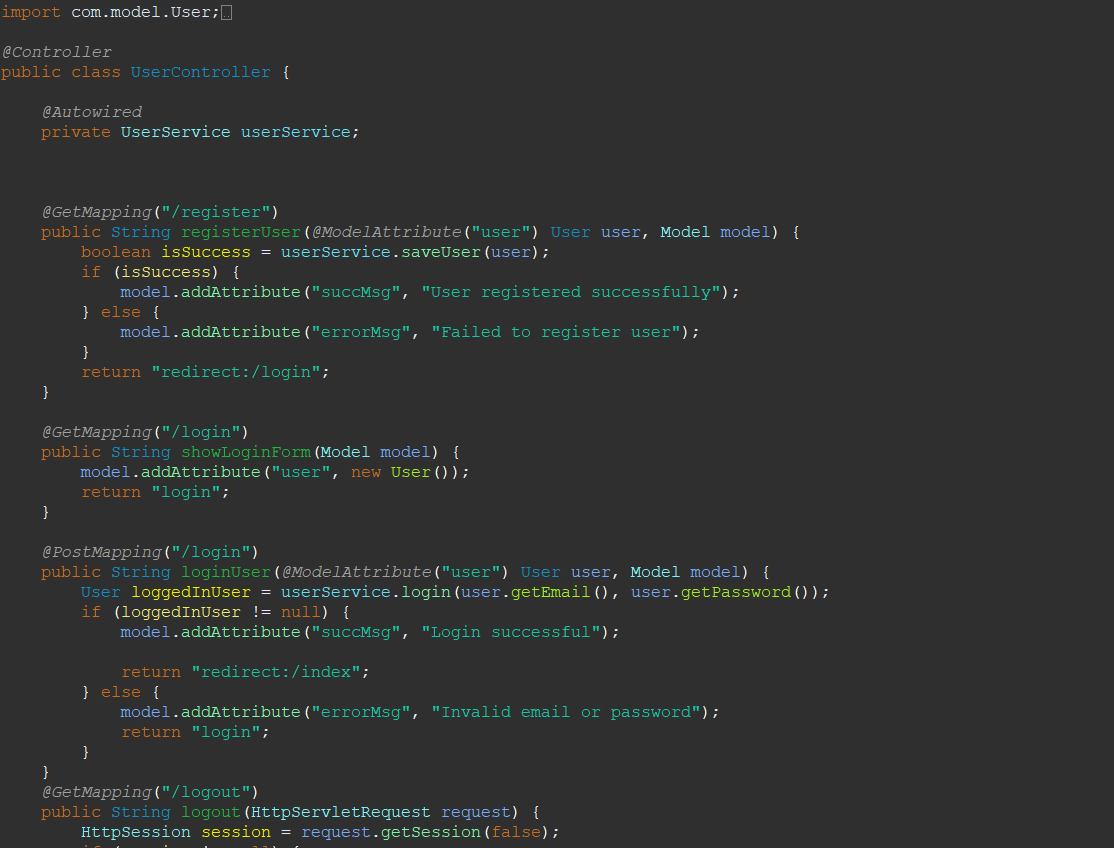
DoctorController.java



SpecialistController.java

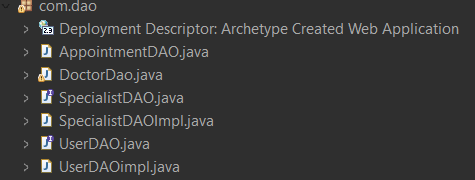


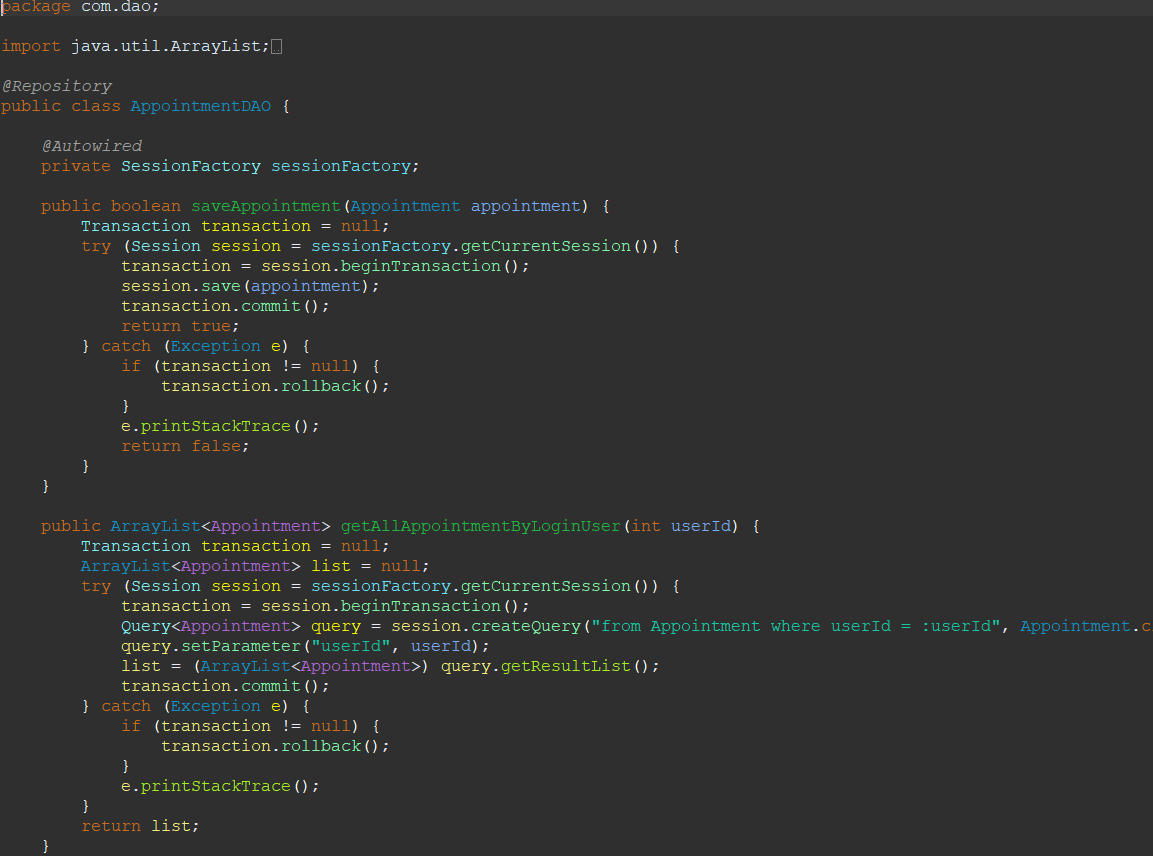
UpdateStatusController.java



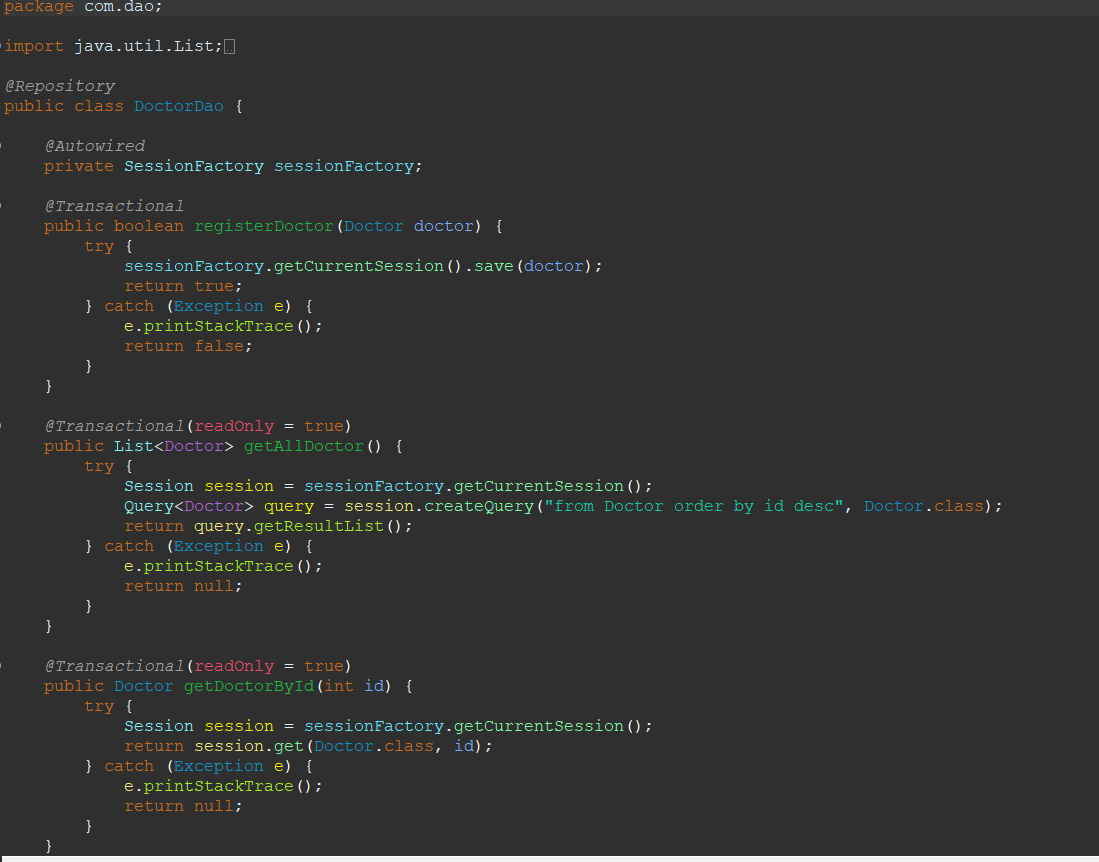
UserController.java

1. **com.dao**

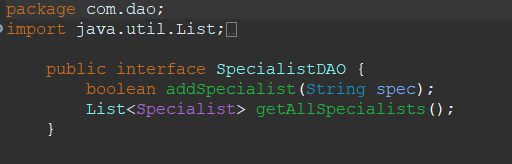




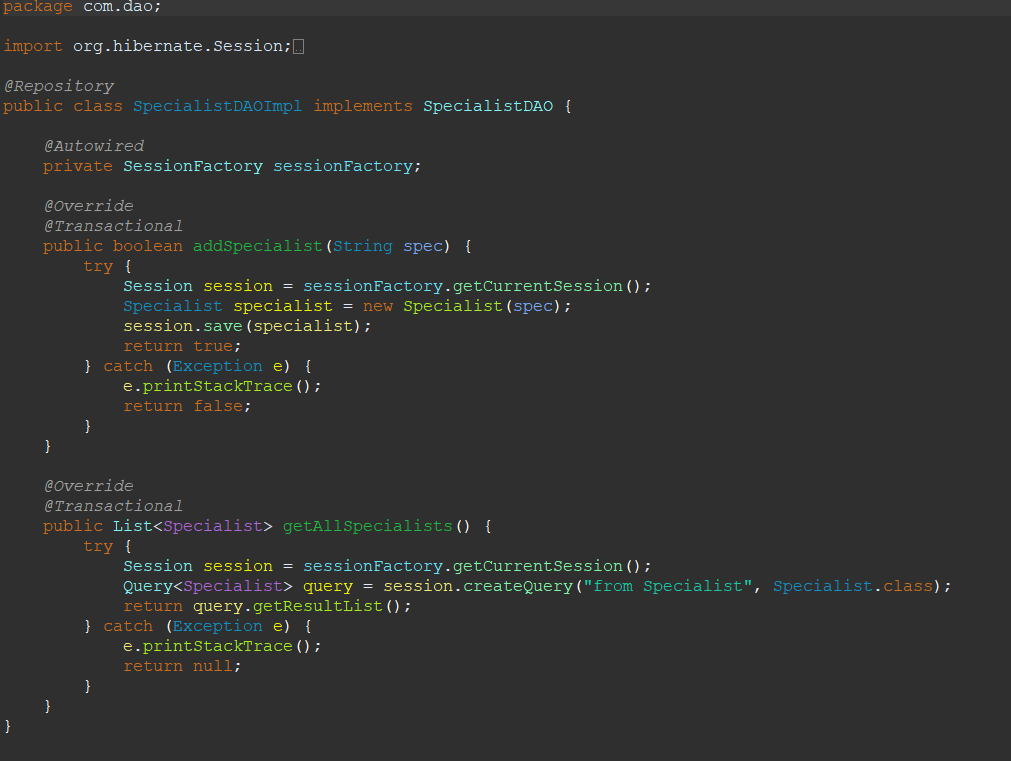
AppointmentDAO.java



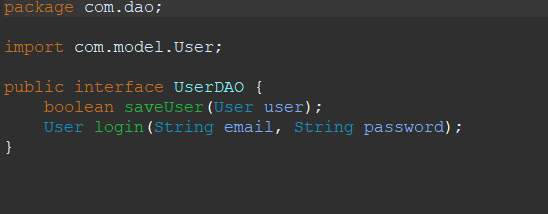
DoctorDAO.java



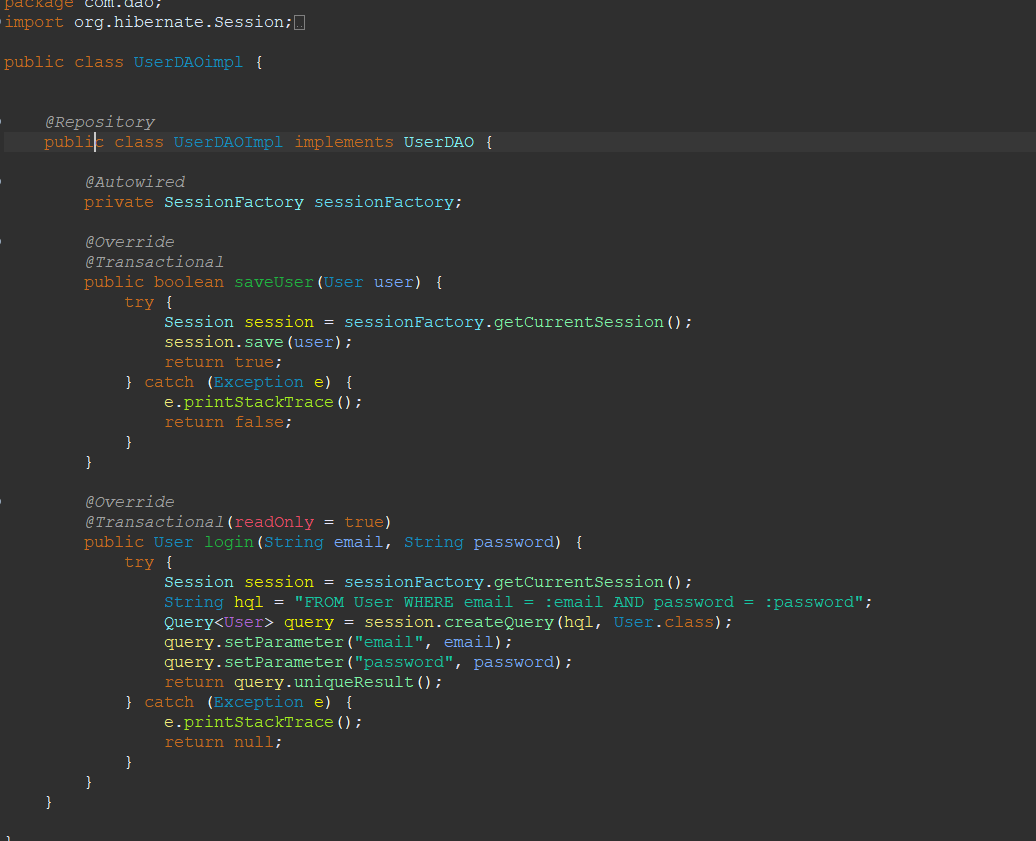
SpecialistDAO.java



SpecialistDAOImpl.java

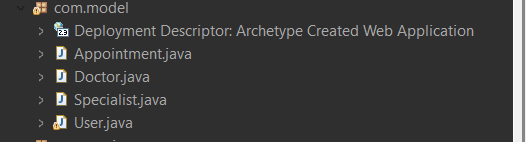


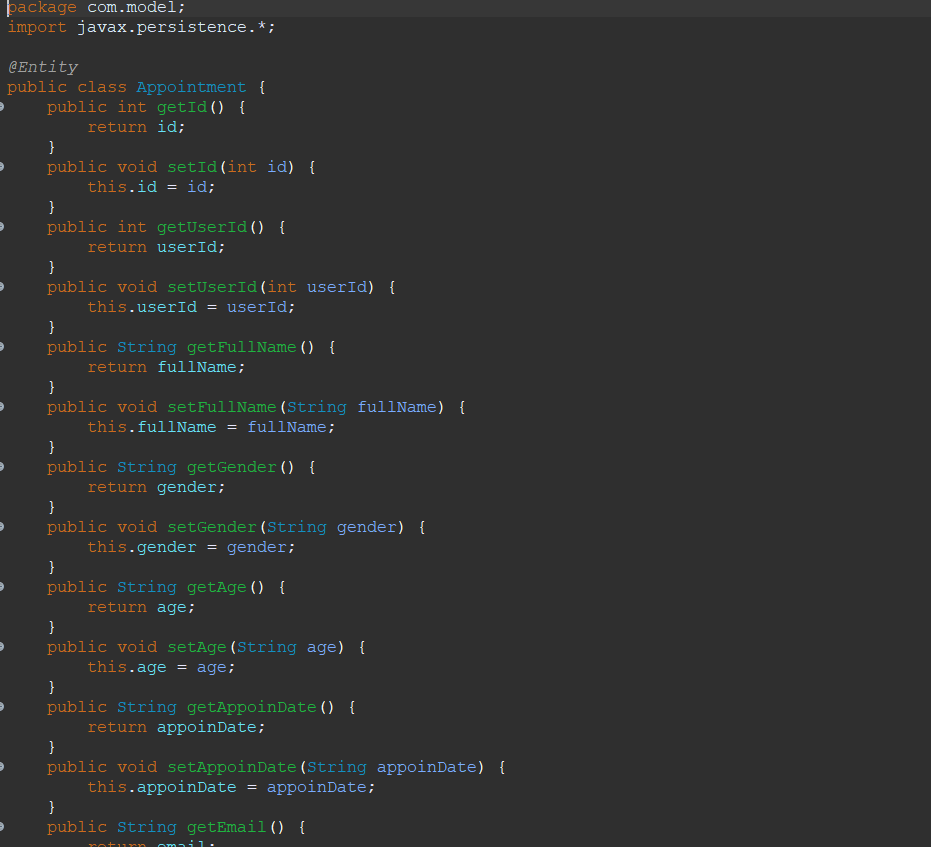
UserDAO.java



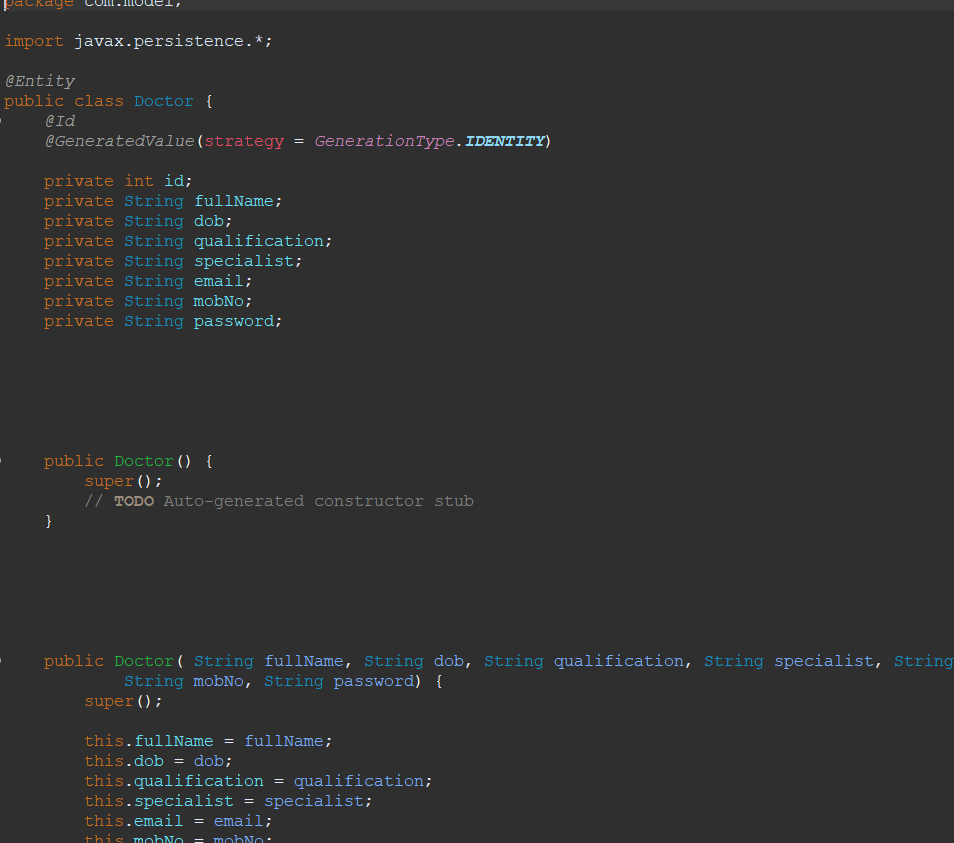
UserDAOImpl.java

1. **com.model**

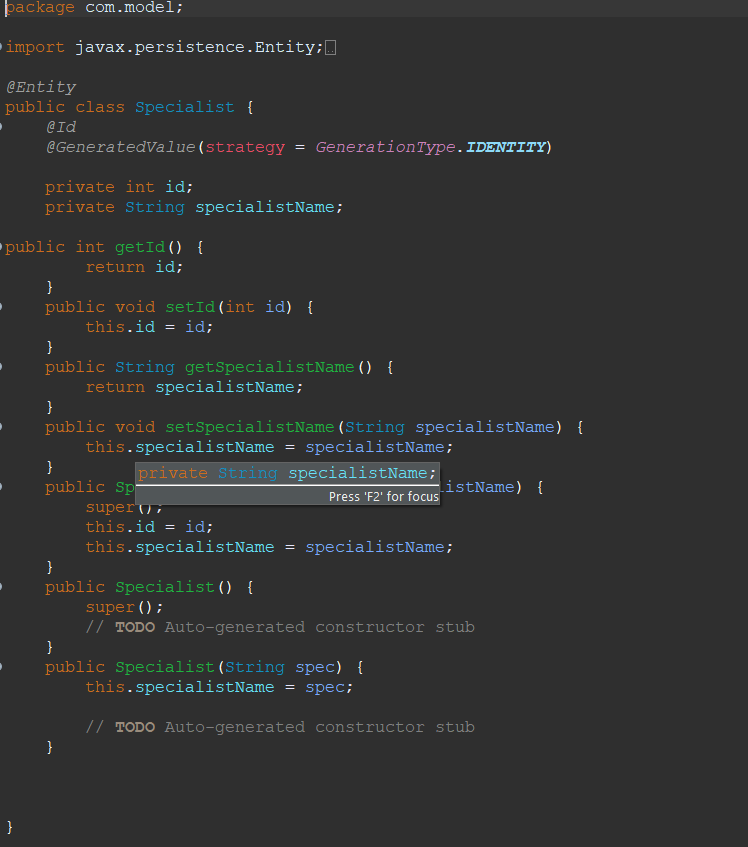




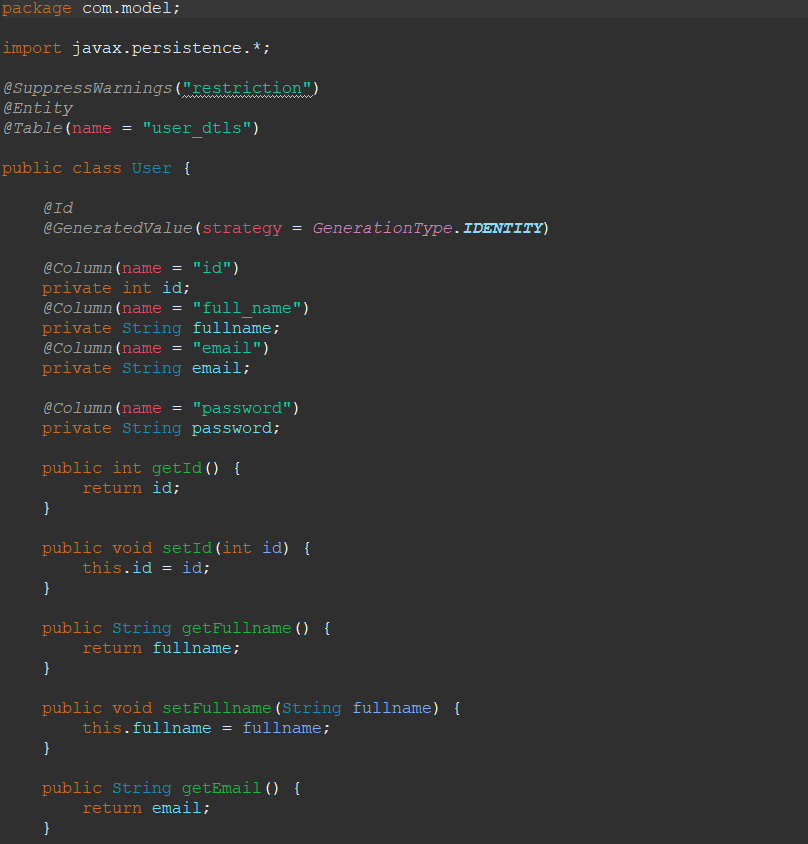
Appointment.java



Doctor.java

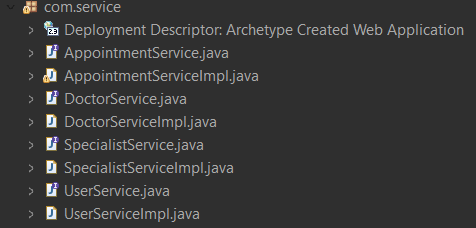


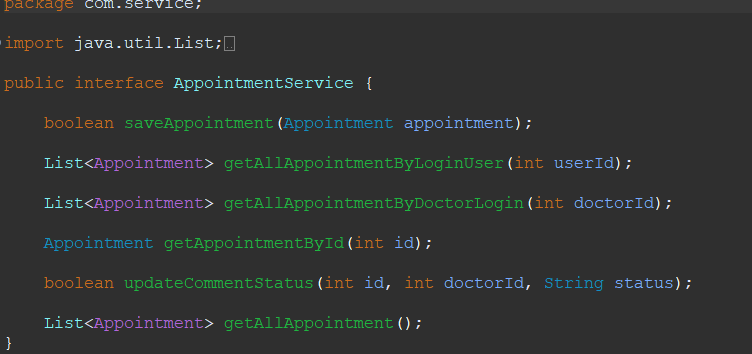
Specialist.java



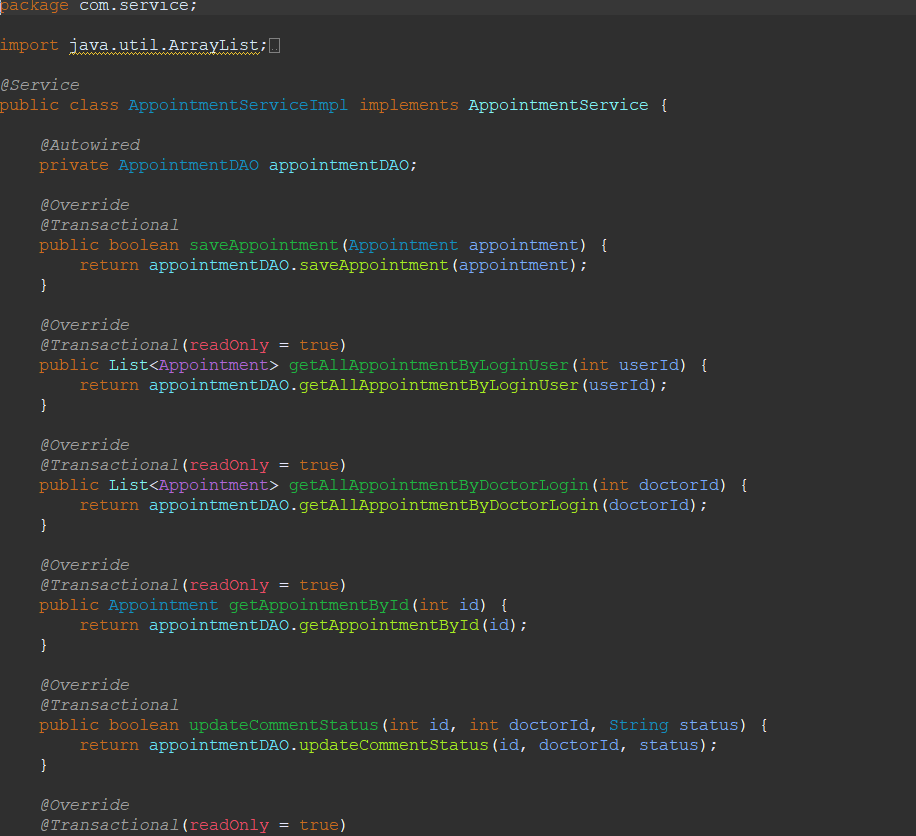
User.java

1. **com.service**

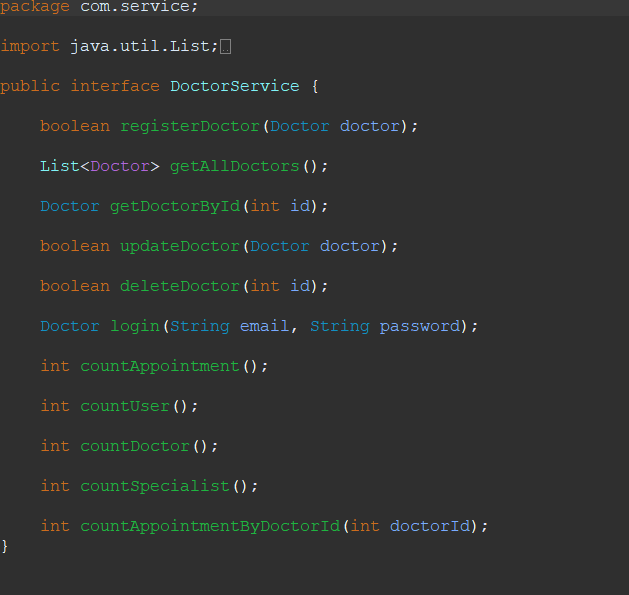




AppointmentService.java



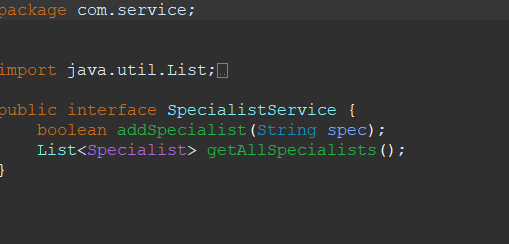
AppointmentServiceImpl.java



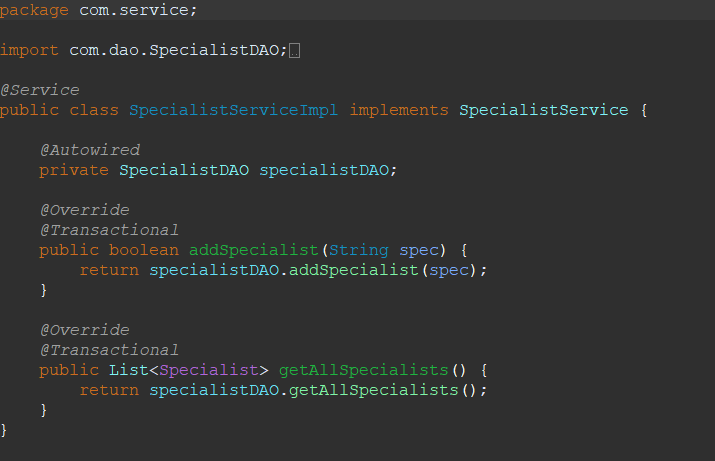
DoctorService.java



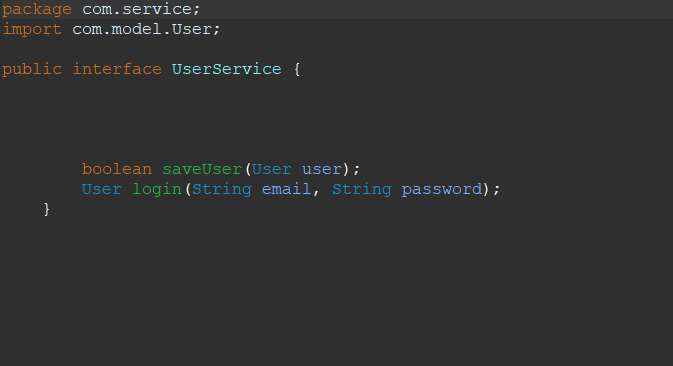
DoctorServiceImpl.java



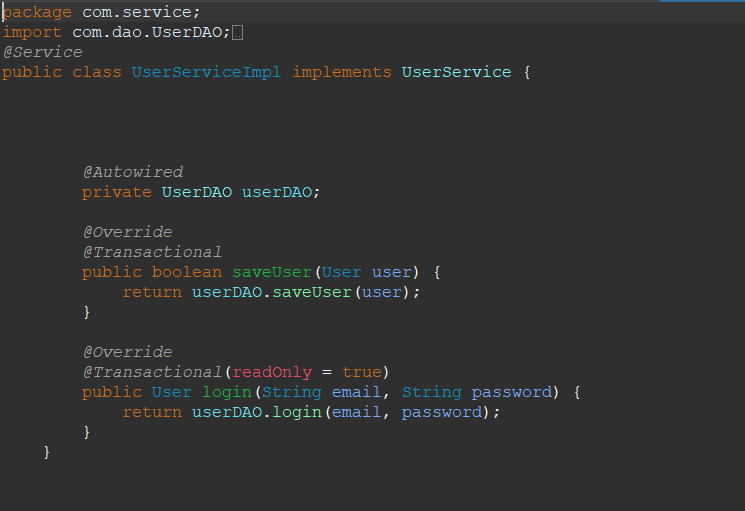
SpecialistService.java



SpecialistServiceImpl.java

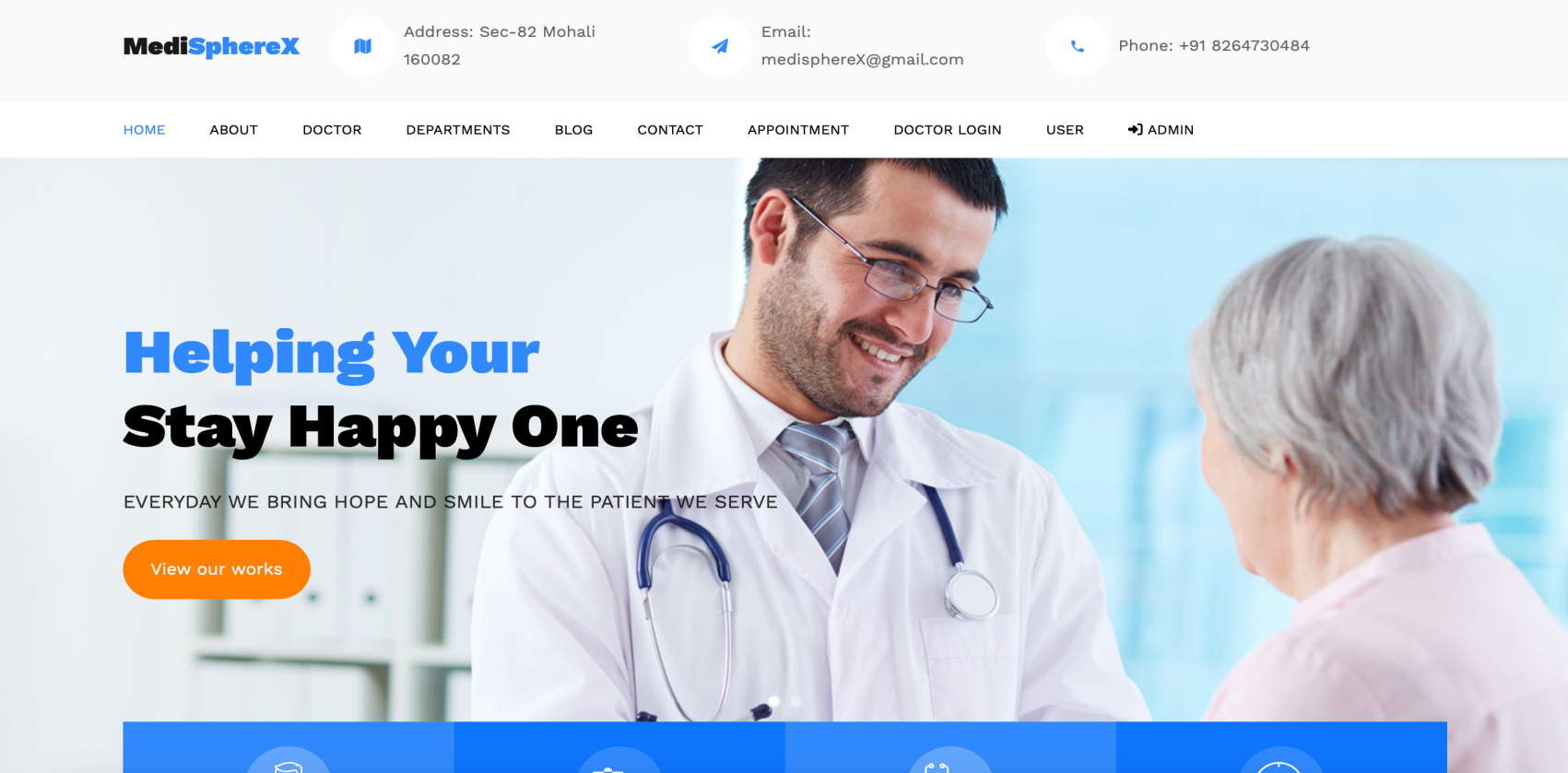


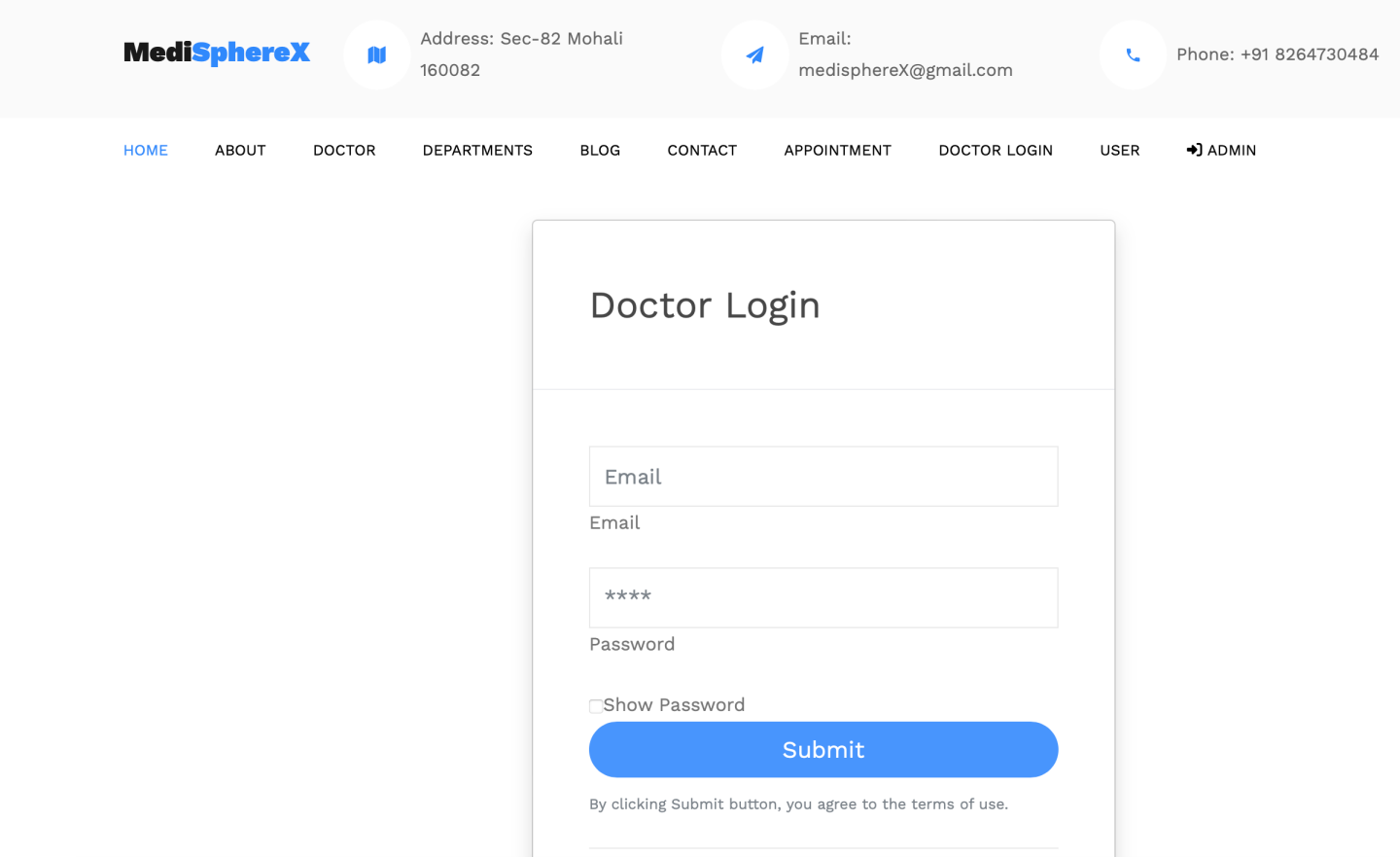
UserService.java

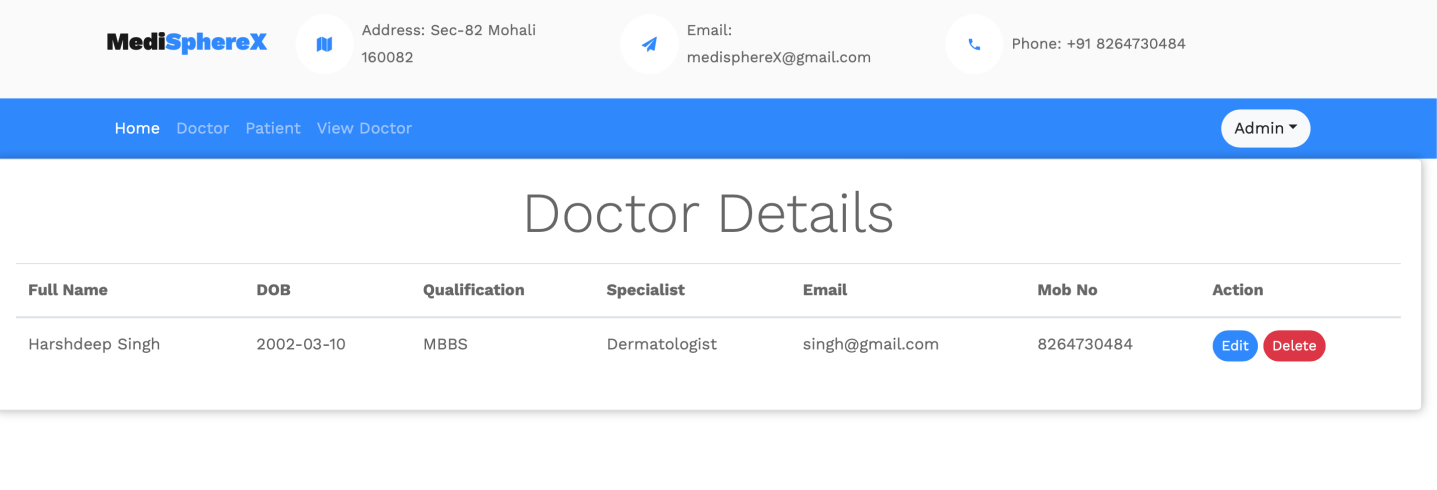


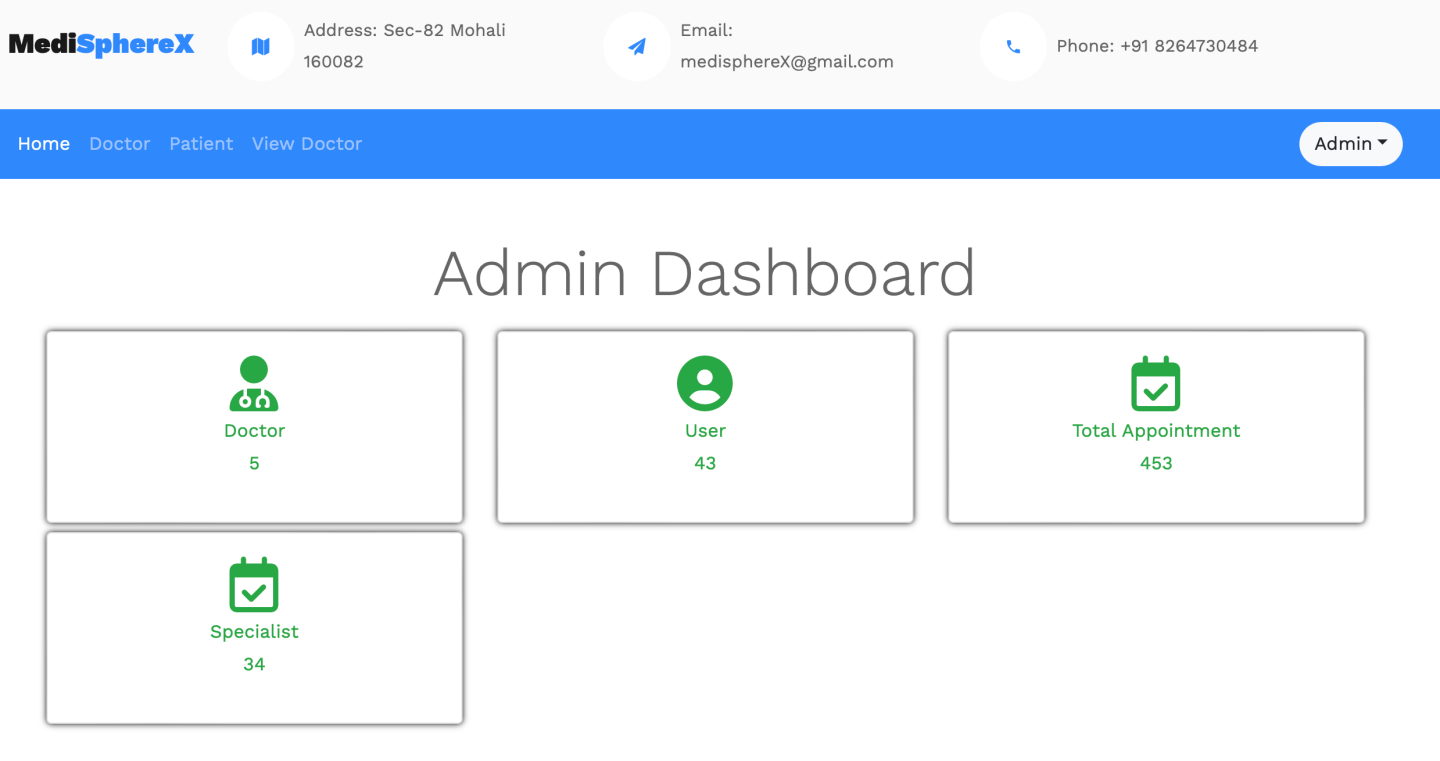
UserServiceImpl.java

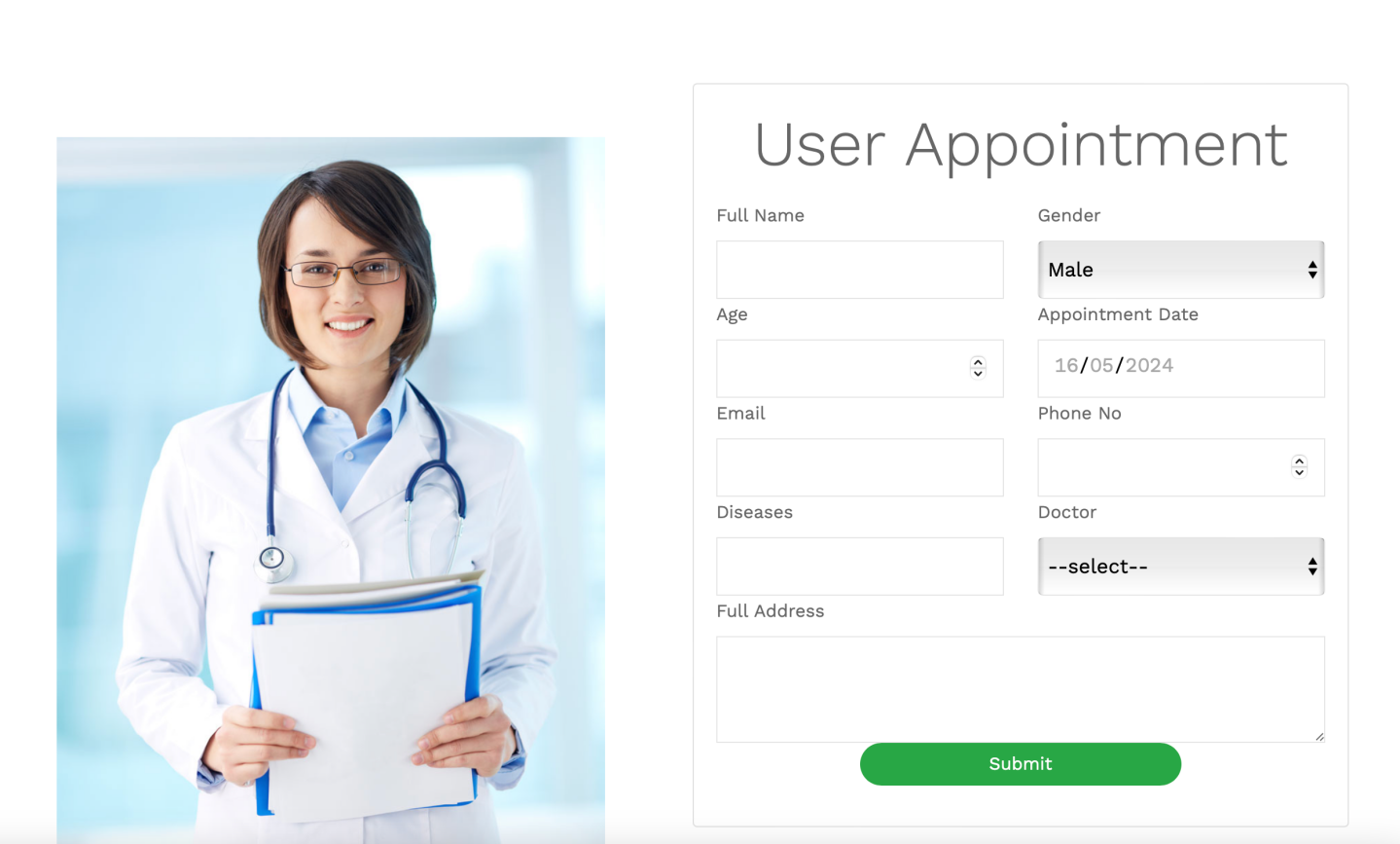
**RESULTS**

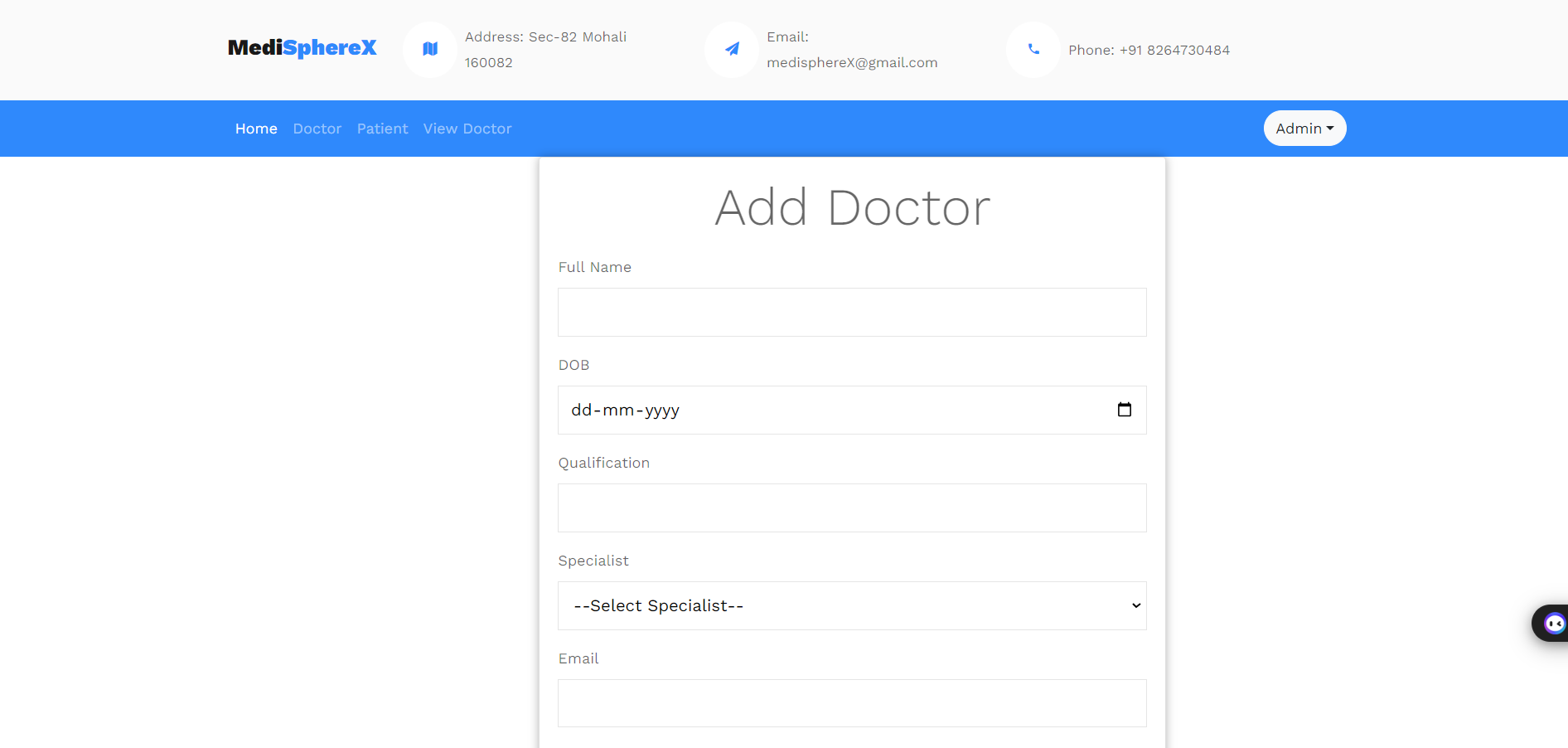


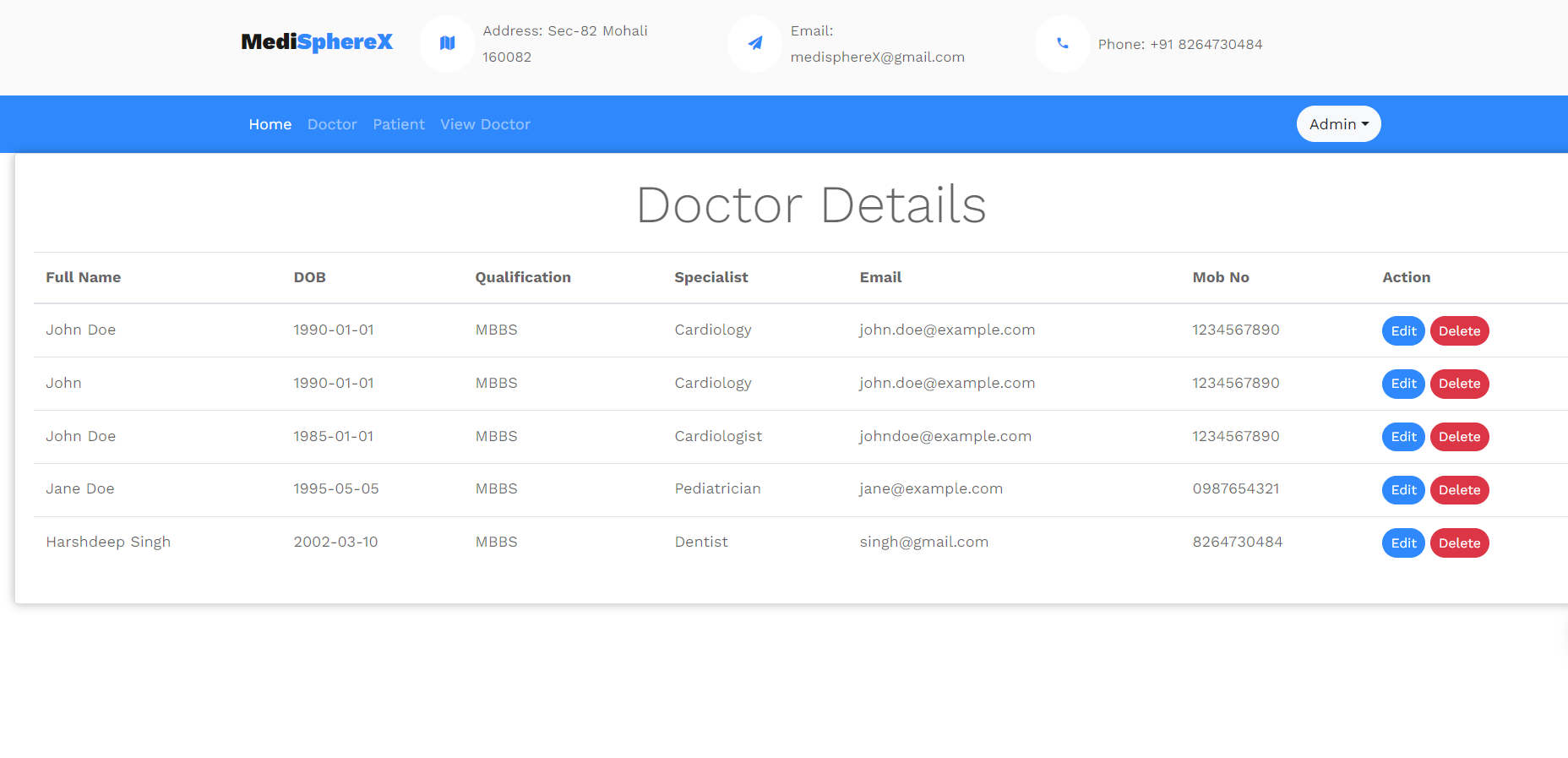
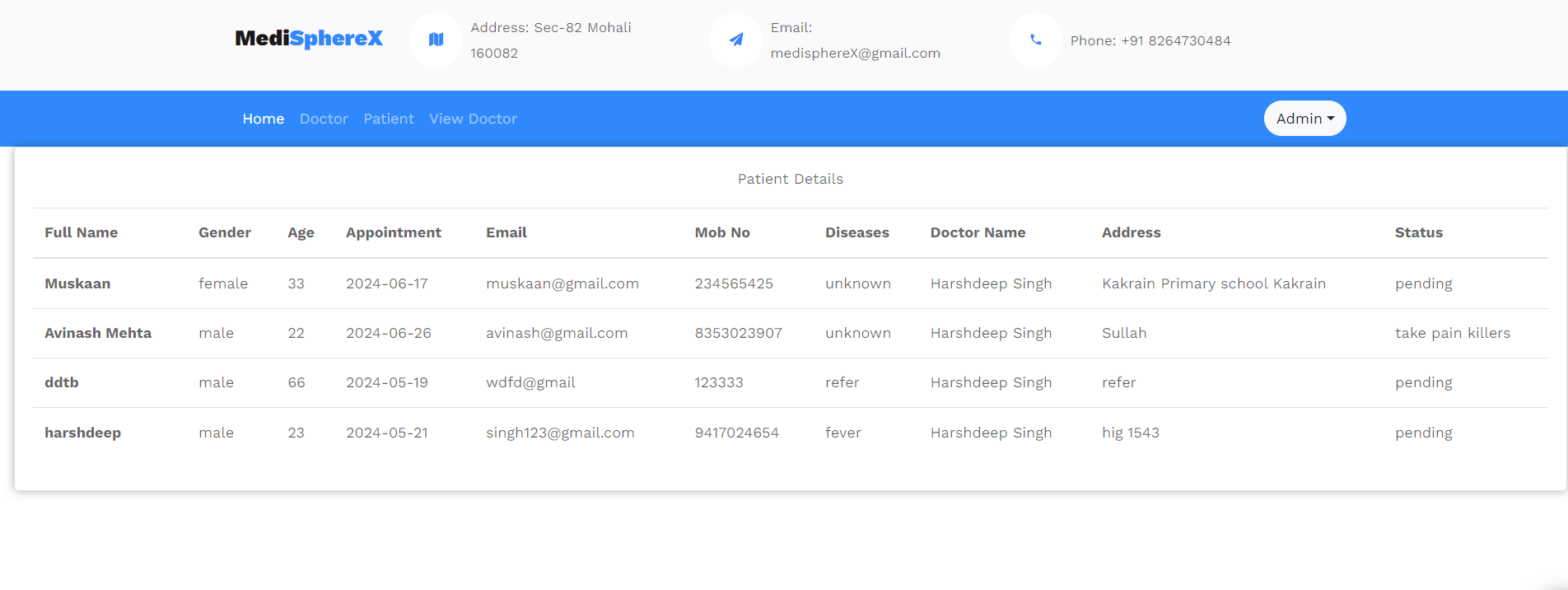


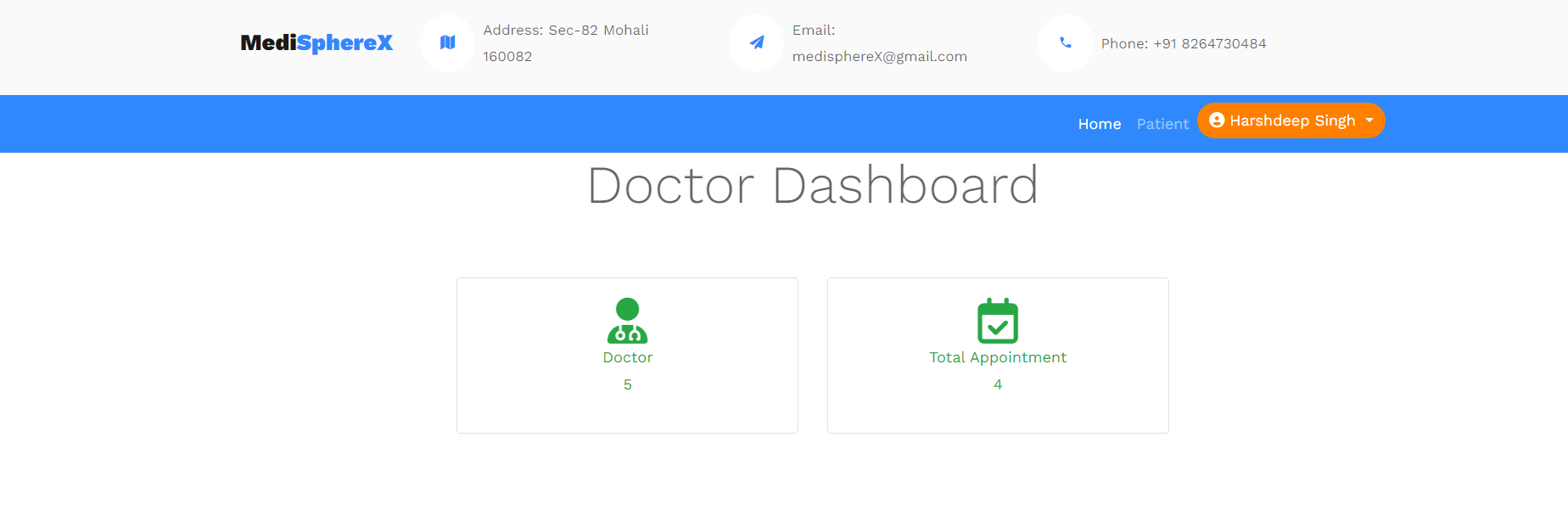


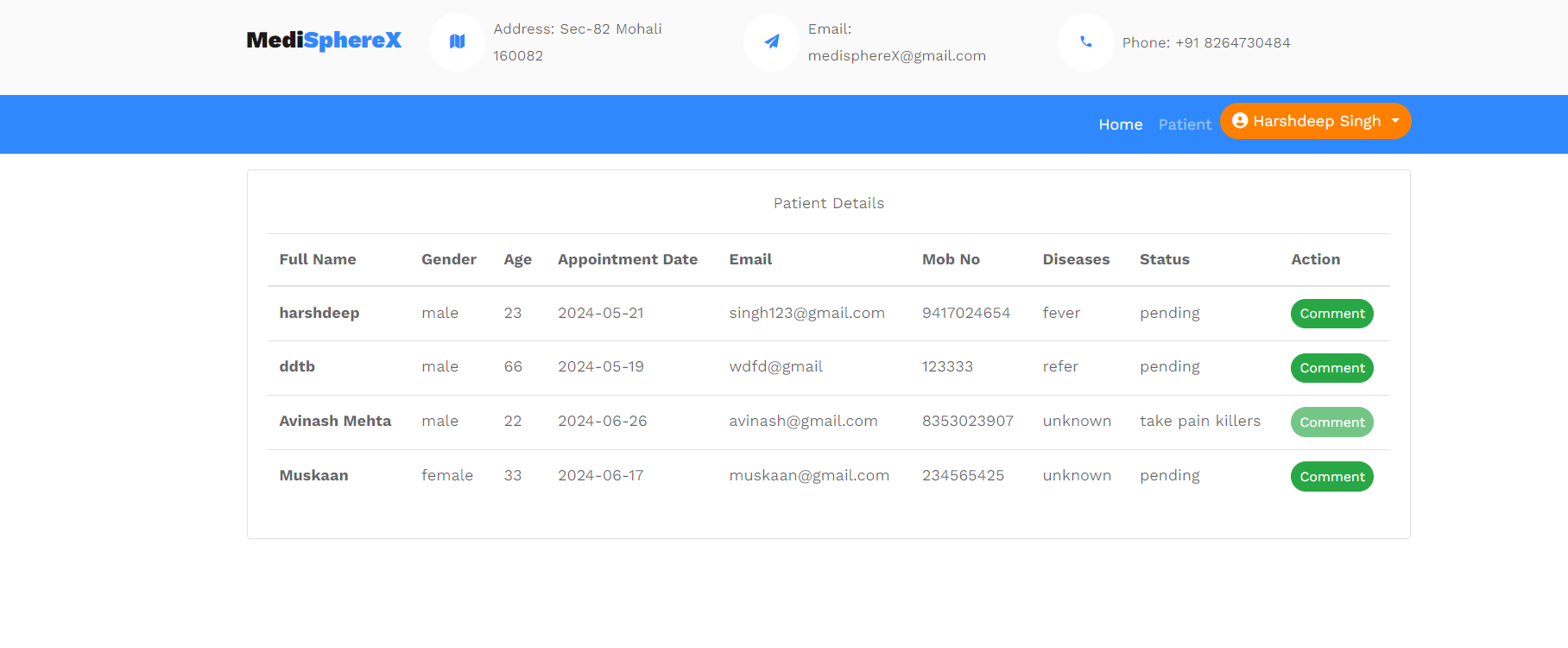


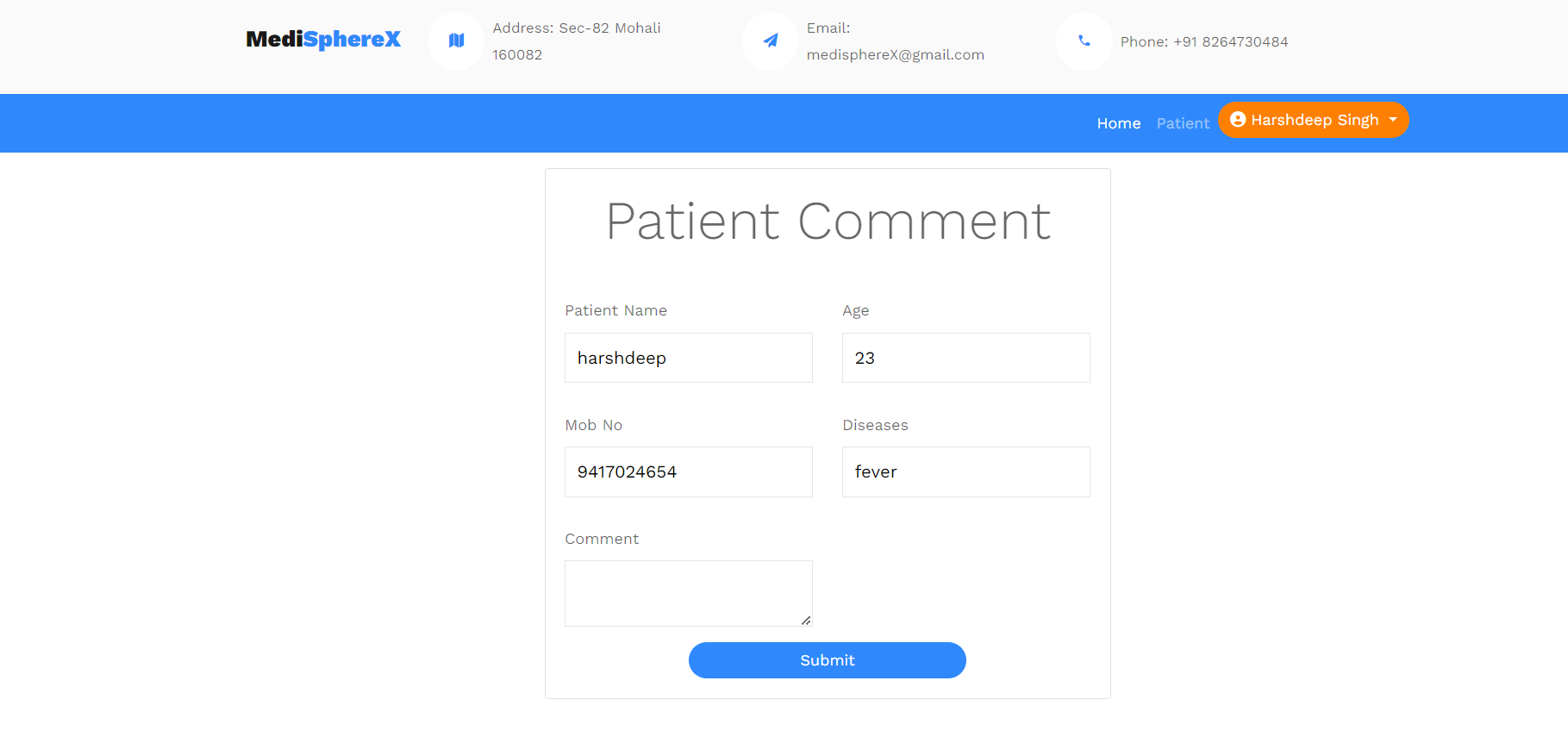




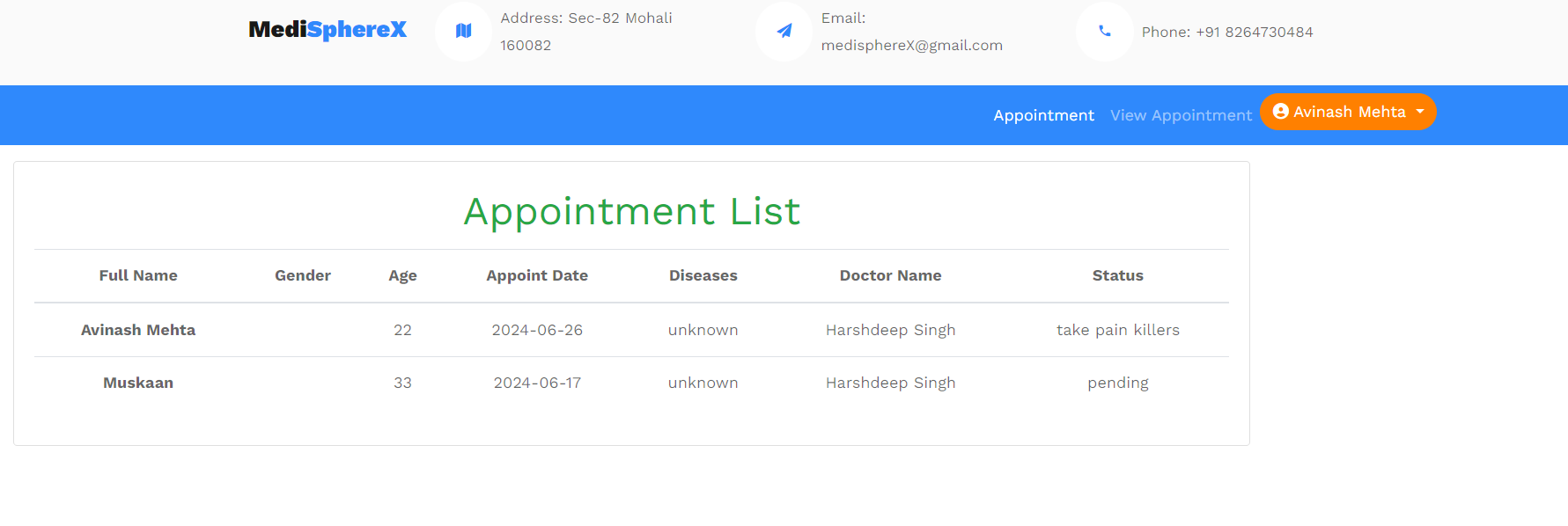


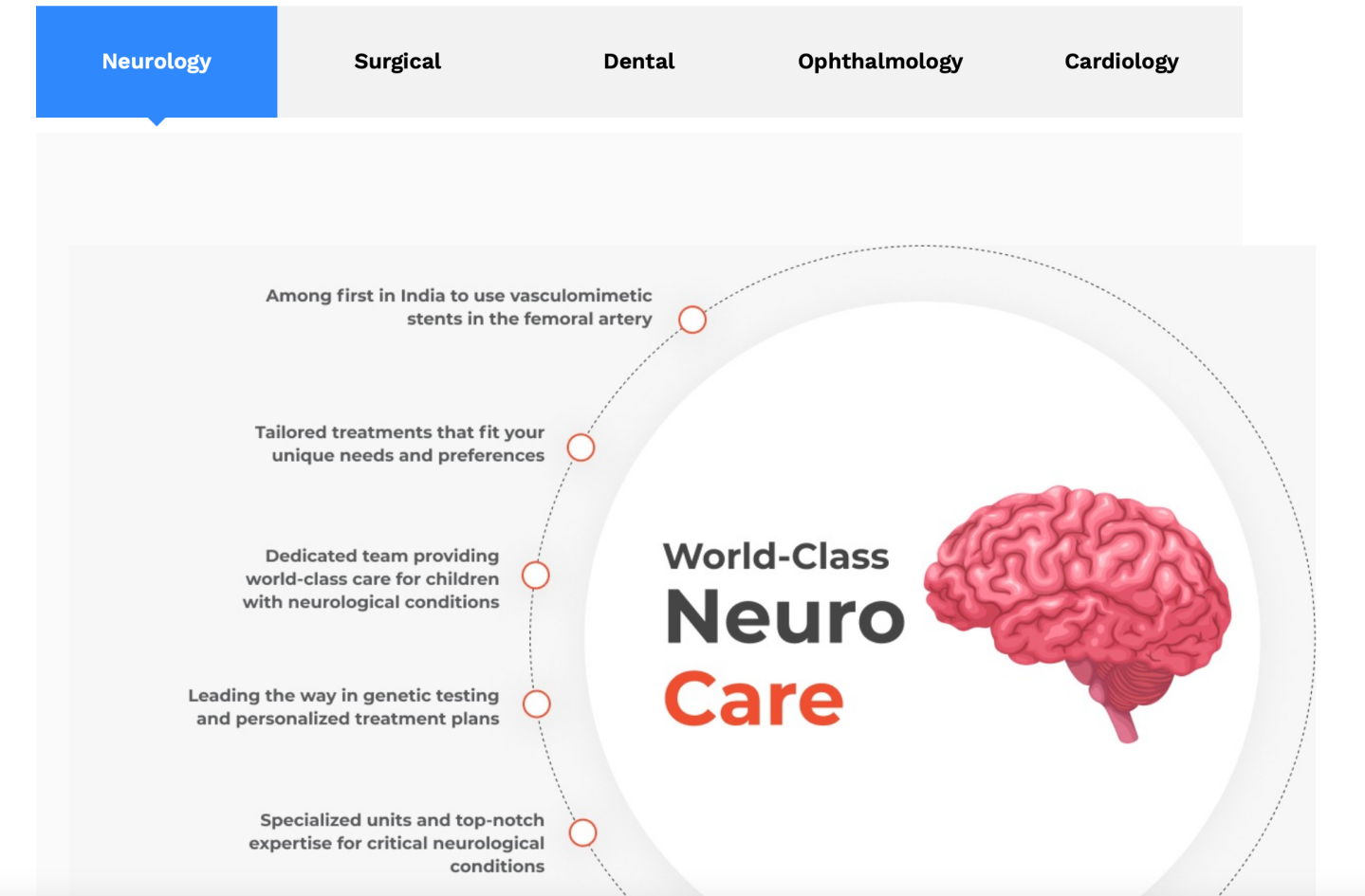




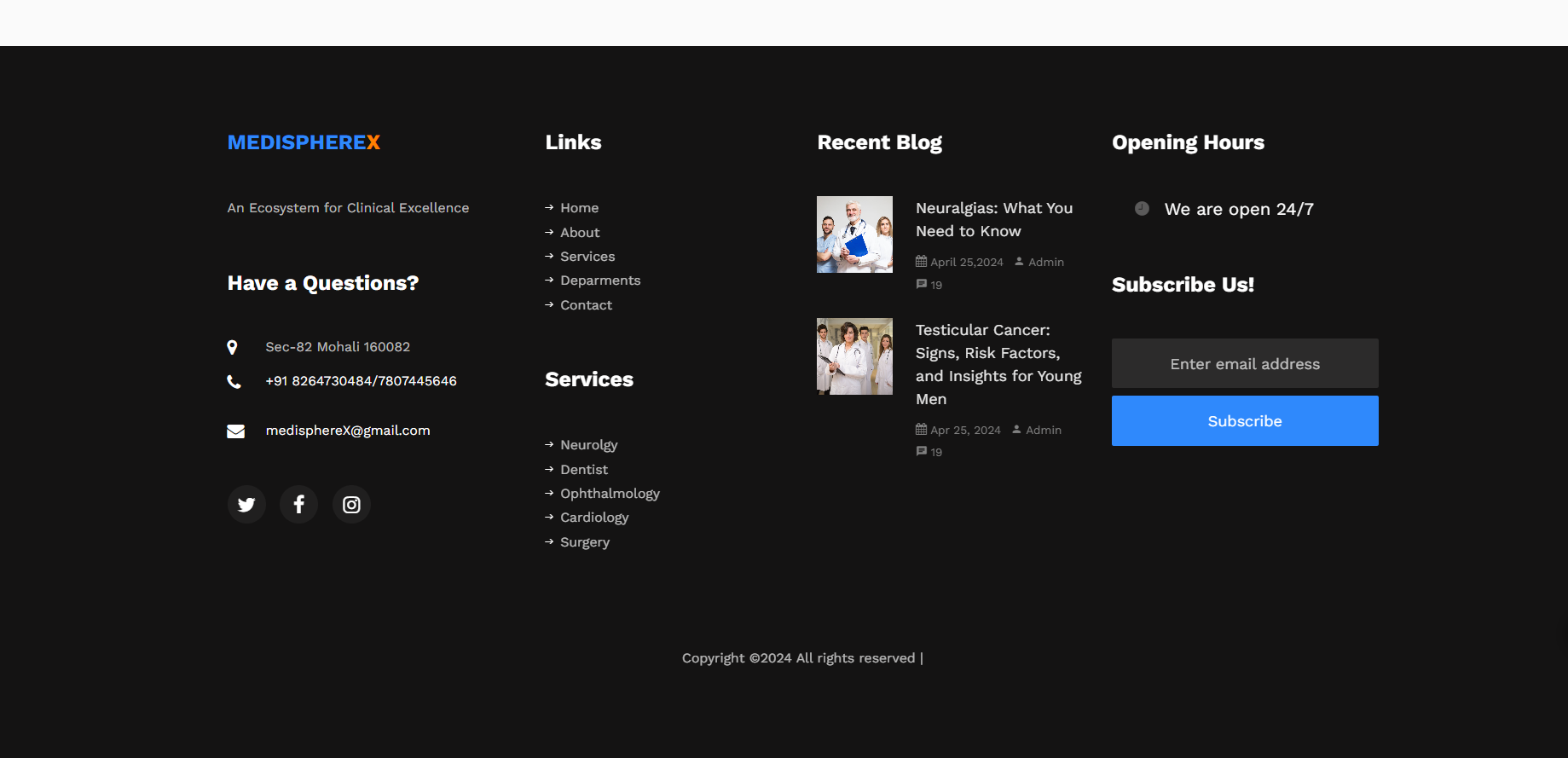












**CONCLUSION**

Working on the project was an excellent experience. It helped us to understand the importance of planning, designing and implementation so far we have learned in our theory books. It helped us unleashing our creativity while working in a team. It also realized the importance of team working, communication as a part of this project.

The project was successfully completed after a lot of efforts and work hours. This project underwent number of compiling, debugging, removing errors, making it bug free, adding more facilities in Hospital Management System and interactivity making it more reliable and useful.

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the hospital. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points...

* A description of the background and context of the project and its relation to work already done in the area.
* Made statement of the aims and objectives of the project.
* The description of Purpose, Scope, and applicability.
* We have defined the problem on which we are working in the project.
* We have described the requirement Specifications of the system and the actions that can be done on these things.

**REFERENCES**

* Google for problem solving
* <https://www.tutorialspoint.com/jdbc/index.htm>
* <https://www.tutorialspoint.com/java/>
* [https://www.javatpoint.com/java-tutorial]( https:/www.javatpoint.com/java-tutorial)
* [https://docs.oracle.com/javase/tutorial/]( https:/docs.oracle.com/javase/tutorial/)
* [https://getbootstrap.com/]( https:/getbootstrap.com/)
* [https://www.JSP.net/]( https:/www.JSP.net/)
* [https://www.tutorialspoint.com/mysql/](https:/www.tutorialspoint.com/mysql/)
* [https.apache.org/docs/2.0/misc/tutorials.html](https.apache.org/docs/2.0/misc/tutorials.html )