Online Health Care Management System

TEAM NAME : SLAYSTACK

TEAM MEMBER :

AAYUSHI KASHYAP
DISHA SUGRA
SNEHA JADAUN

Project Aim and Capabilities

The **Online Healthcare Management System** aims to streamline healthcare operations by enabling patients to book appointments, while doctors manage schedules and admin handles user settings.

User Types

ADMIN

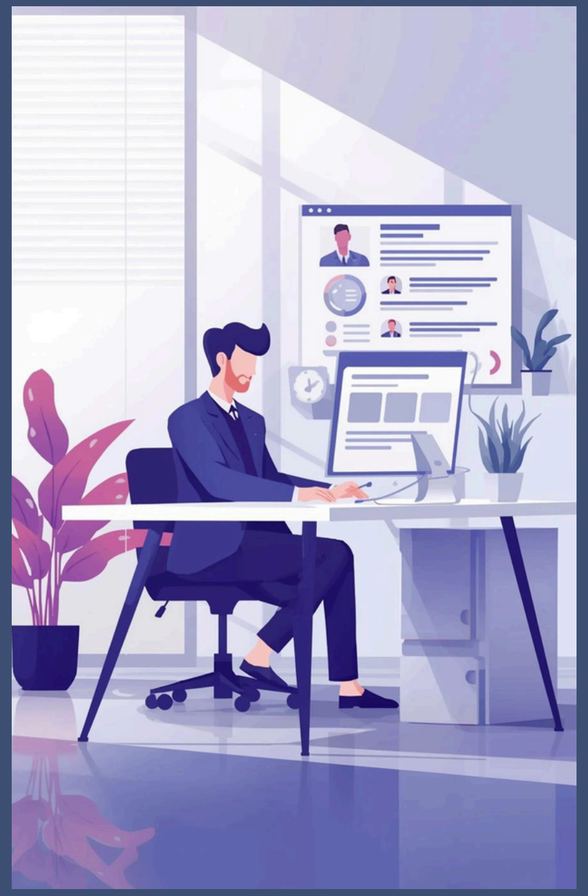
System overseer

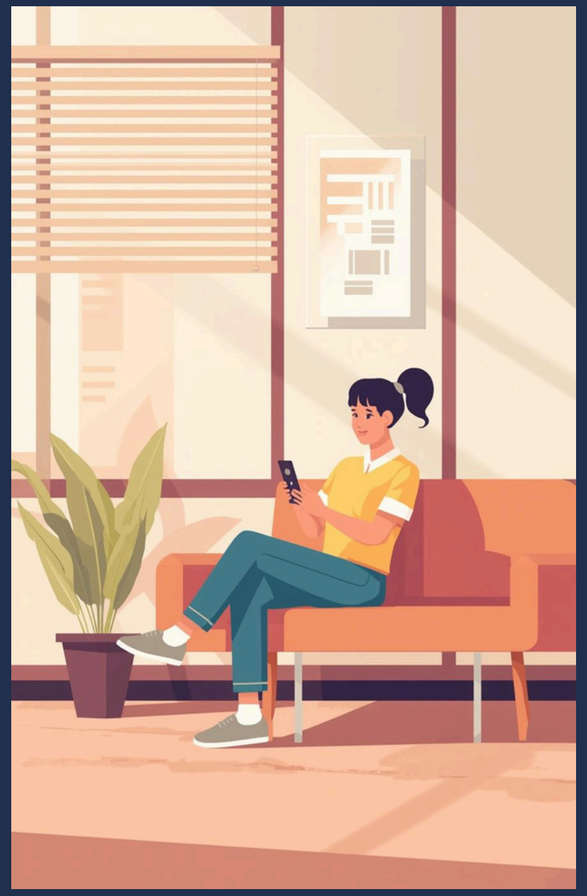
DOCTOR

Healthcare provider

PATIENT

Service user





Admin Functionalities Overview

USER MANAGEMENT

The system allows admins to easily input user details, manage user roles, and ensure efficient updates, providing confirmation for each create, update, or delete operation.

APPOINTMENT MANAGEMENT

Admins can manage all appointment details, ensuring the correct scheduling of patient visits and doctor availability, with intuitive interfaces for easy access and modifications.

SYSTEM SETTINGS

Administrators can configure system settings efficiently, allowing for seamless updates and management of the healthcare platform, which ensures optimal performance and security for all users.

Doctor Functionalities in the System

SCHEDULE MANAGEMENT

Doctors can efficiently manage their **daily schedules**, allowing them to set, modify, and view appointments, ensuring they remain organized and accessible to both patients and staff.

PATIENT RECORDS

The system enables doctors to maintain **comprehensive patient records**, offering quick access to medical histories, treatment plans, and test results, which assists in providing personalized care to each patient.

APPOINTMENT MANAGEMENT

Doctors can handle all aspects of **appointment management**, from confirming bookings to rescheduling, ensuring optimal patient flow and enhancing overall service quality within the healthcare facility.

Patient Functionalities in the System

APPOINTMENT BOOKING

Patients can easily schedule their appointments online, selecting preferred times and doctors. This functionality streamlines the booking process and reduces waiting times for both patients and clinics.

MEDICAL HISTORY

The system allows patients to view and update their medical history, ensuring that doctors have access to accurate and complete information for informed decision-making during consultations.

PROFILE MANAGEMENT

Users can manage their profiles efficiently, updating personal information and preferences. This feature promotes user engagement and ensures that patient data remains current and secure.

Admin Dashboard Features

USER MANAGEMENT TABLE

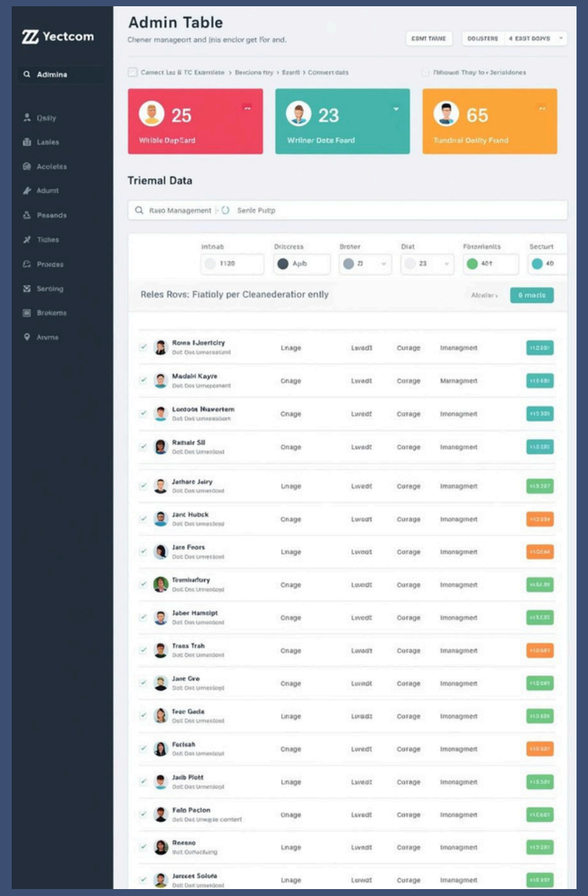
Organized data

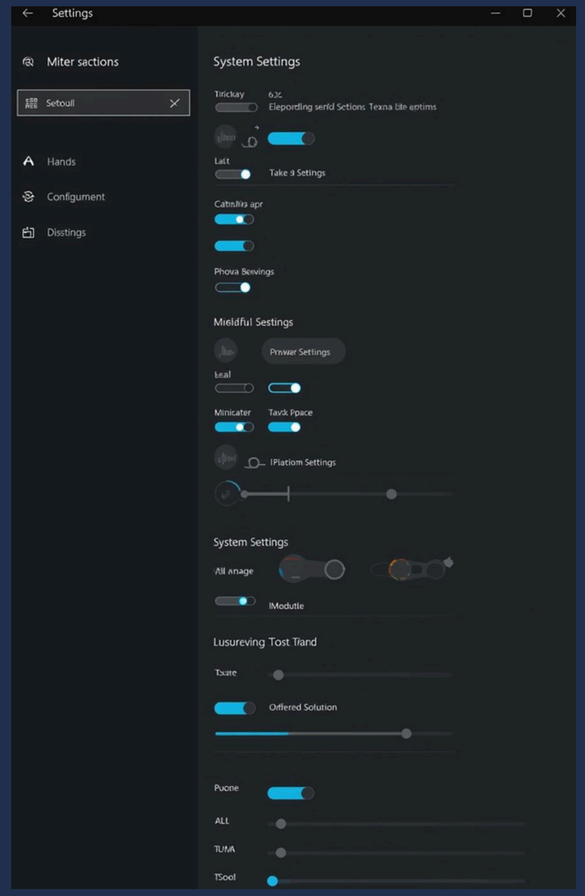
APPOINTMENT MANAGEMENT

Streamlined interface

SYSTEM SETTINGS PANEL

Configuration options





Doctor Dashboard Features

APPOINTMENT SCHEDULE

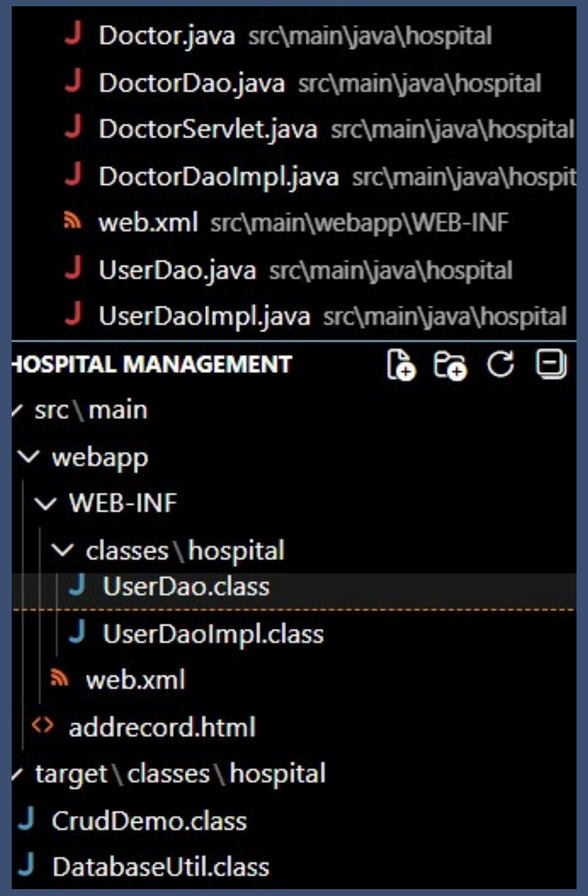
Calendar view

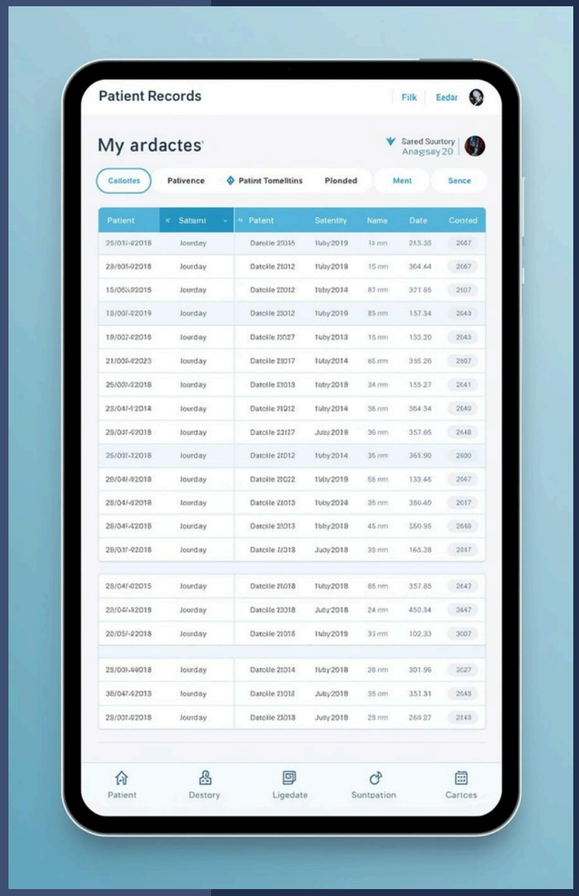
PATIENT RECORDS

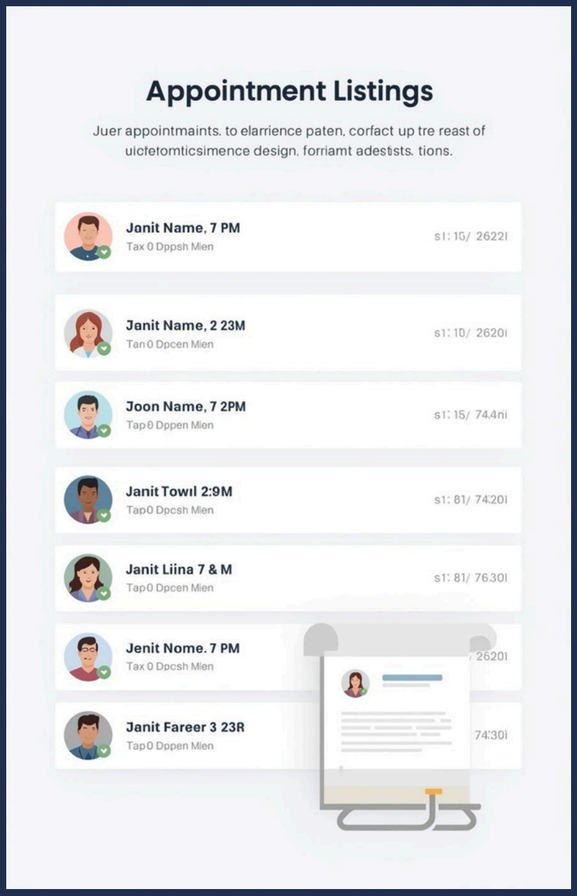
Data overview

APPOINTMENT LISTINGS

Overview panel







Patient Dashboard Features

APPOINTMENT HISTORY

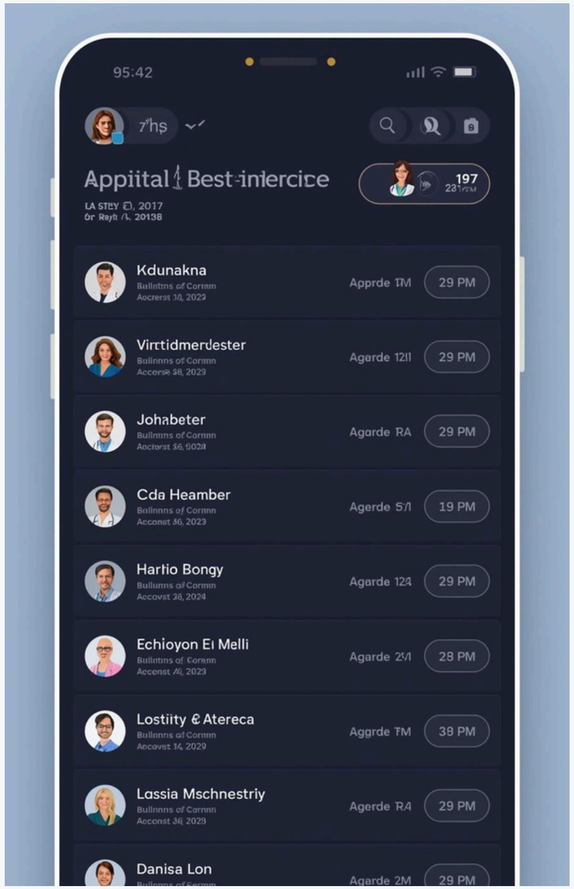
Track all past visits

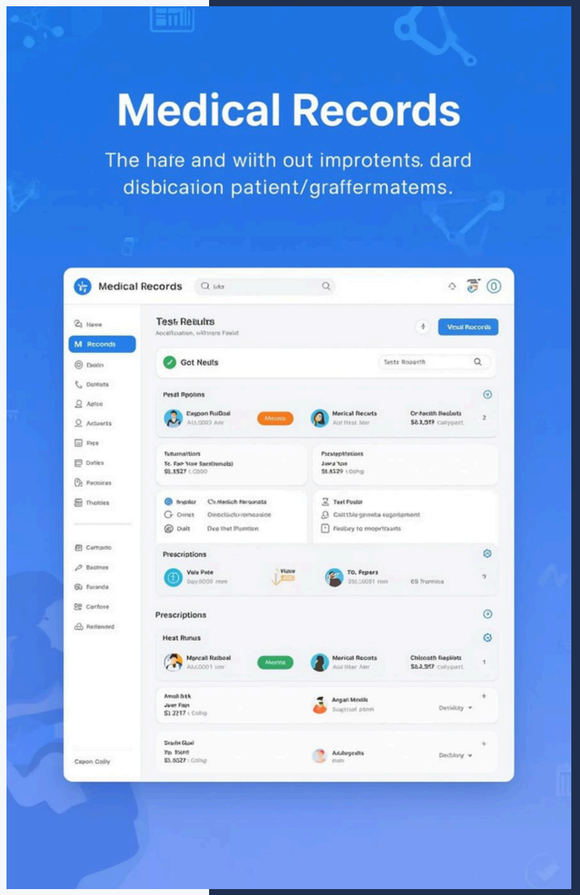
MEDICAL RECORDS VIEWER

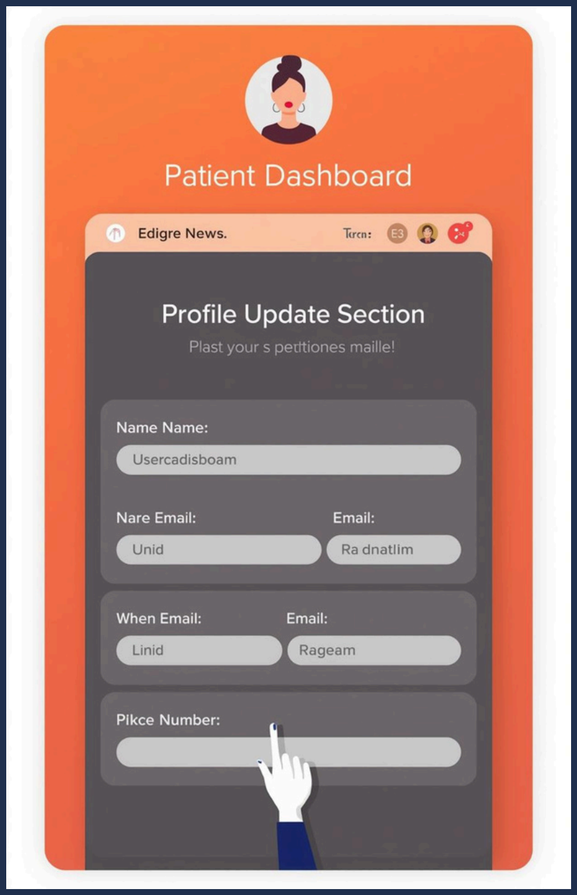
Access health history

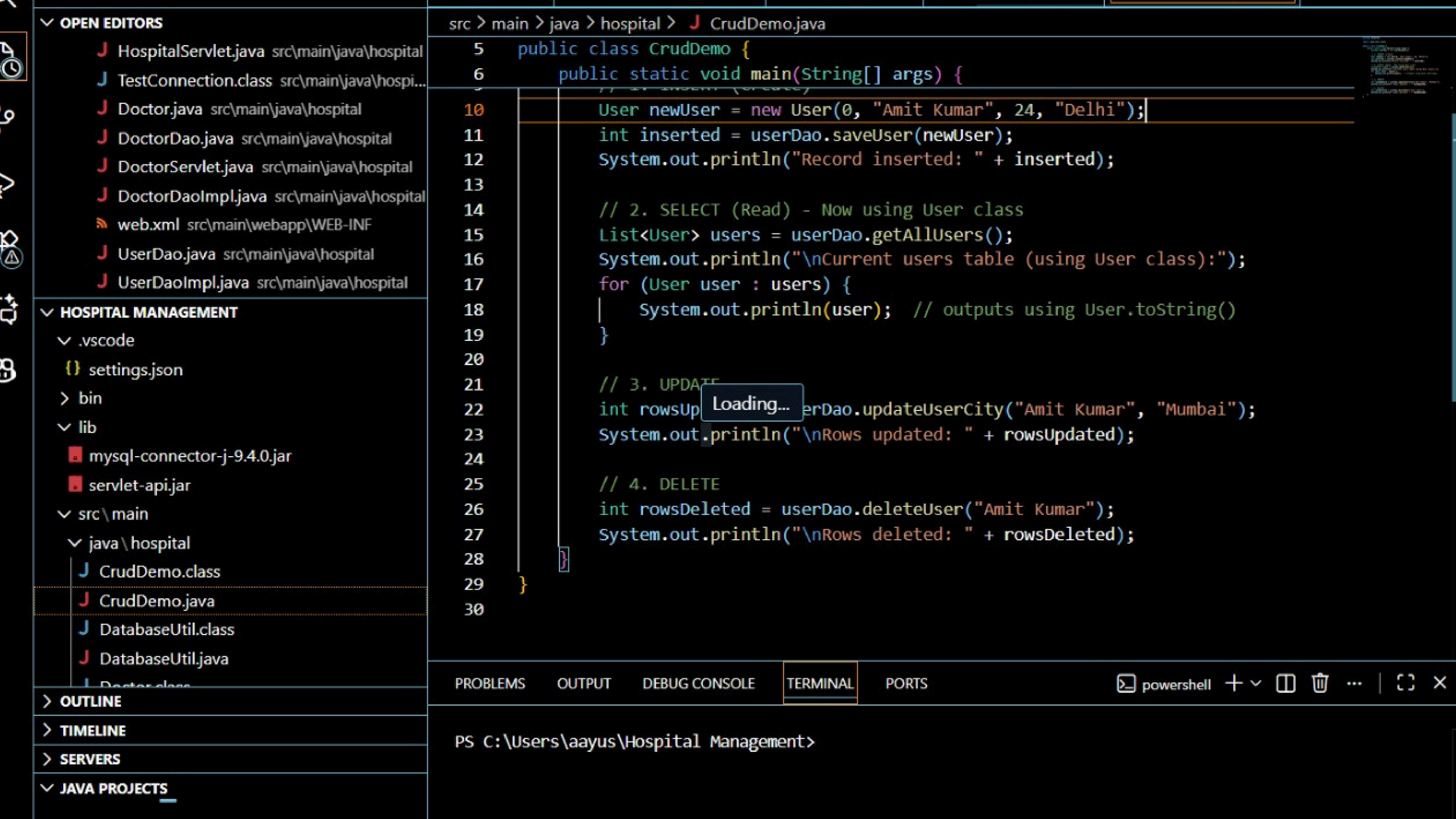
PROFILE UPDATE SECTION

Manage personal details








OPEN EDITORS
HospitalServlet.java src\main\java\hospital
TestConnection.class src\main\java\hospi...
Doctor.java src\main\java\hospital
DoctorDao.java src\main\java\hospital
DoctorServlet.java src\main\java\hospital
DoctorDaoImpl.java src\main\java\hospital
web.xml src\main\webapp\WEB-INF
UserDao.java src\main\java\hospital
UserDaoImpl.java src\main\java\hospital
HOSPITAL MANAGEMENT
.vscode
settings.json
bin
lib
mysql-connector-j-9.4.0.jar
servlet-api.jar
src\main
java\hospital
CrudDemo.class
CrudDemo.java
DatabaseUtil.class
DatabaseUtil.java
OUTLINE
TIMELINE
SERVERS
JAVA PROJECTS
src > main > java > hospital > CrudDemo.java
5 public class CrudDemo {
6 public static void main(String[] args) {
10 User newUser = new User(0, "Amit Kumar", 24, "Delhi");
11 int inserted = userDao.saveUser(newUser);
12 System.out.println("Record inserted: " + inserted);
13
14 // 2. SELECT (Read) - Now using User class
15 List<User> users = userDao.getAllUsers();
16 System.out.println("\nCurrent users table (using User class):");
17 for (User user : users) {
18 System.out.println(user); // outputs using User.toString()
19 }
20
21 // 3. UPDATE
22 int rowsUp Loading... erDao.updateUserCity("Amit Kumar", "Mumbai");
23 System.out.println("\nRows updated: " + rowsUpdated);
24
25 // 4. DELETE
26 int rowsDeleted = userDao.deleteUser("Amit Kumar");
27 System.out.println("\nRows deleted: " + rowsDeleted);
28 }
29 }
30
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS C:\Users\aayus\Hospital Management>

THANK YOU