



DR. REDDY’S DENTAL HEALTHCARE

CS 505, Introduction to Computer Science w/ Java Project Professor. Saha Sukla



DECEMBER 9, 2022

GROUP MEMBER’S

Surekha Salunkhe [ss87010n@pace.edu](mailto:ss87010n@pace.edu)

Rohit Ujgare [ru71276n@pace.edu](mailto:ru71276n@pace.edu)

Ankith Reddy [vu49059n@pace.edu](mailto:vu49059n@pace.edu)

Soma Sekhar Alapati [sa81686n@pace.edu](mailto:sa81686n@pace.edu)

Noma Zareen [nz79920n@pace.edu](mailto:nz79920n@pace.edu)

# Table of contents

[ABSTRACT 2](#_bookmark0)

1. [CLINIC MANAGEMENT SYSTEM 3](#_bookmark1)
2. [REQUIREMENT ANALYSIS 4](#_bookmark2)
   1. [NEED FOR A SOFTWARE SYSTEM AT THE CLINIC 4](#_bookmark3)
   2. [MODULES 5](#_bookmark4)
      1. [Patient Registration Module 5](#_bookmark5)
      2. [Appointment and Scheduling 5](#_bookmark6)
      3. [Reception Management 5](#_bookmark7)
      4. [Billing and Payment 5](#_bookmark8)
   3. [REQUIREMENTS OF THE PROJECT 6](#_bookmark9)
      1. [Functional Requirements: 6](#_bookmark10)
      2. [Non-Functional Requirements: 6](#_bookmark11)
3. [SPECIFICATIONS OF THE PROJECT 7](#_bookmark12)
   1. [Backend structure: 7](#_bookmark13)
   2. [Front-end structure 7](#_bookmark14)
4. [USE CASE DIAGRAM 8](#_bookmark15)
5. [CLASS DIAGRAM 9](#_bookmark16)
6. [PROJECT IMPLEMENTATION 11](#_bookmark17)
   1. [Admin 12](#_bookmark18)
   2. [Doctor 12](#_bookmark19)
   3. [Main 12](#_bookmark20)
   4. [Patient 13](#_bookmark21)
   5. [Receptionist 13](#_bookmark22)
7. [Flow Diagram 14](#_bookmark23)
8. [SNAPSHOTS OF PROJECT EXECUTION 15](#_bookmark24)
9. [SNAPSHOTS OF PHP Database STRUCTURE 23](#_bookmark25)
10. [CONCLUSION 25](#_bookmark26)

Disclaimer:

# ABSTRACT

For this project, we have assumed a fictious dental clinic by the name of **Dr. Reddy’s Dental Healthcare Center** to make the planning and design process easier.

Dr. Reddy’s Dental Healthcare Center is a busy dental clinic run by a group of dentists that have an existing patient database. However, all of its current records are stored manually. We, in this project, aim to digitalize the day-to-day operations of the clinic and move all existing patient records to a Clinic Management Software.

By the end of this project, we will be able to build a working prototype of a dental clinic management system that has 4 different modules-

1. Receptionist
2. Appointment
3. Patient
4. Doctor

We will further define the functionality of each module both at the frontend and the backend and discuss in detail about the best model to implement the software development lifecycle for this project.

As for the storage component, we will determine the parameters needed for storage and design a database to store all relevant records. Using use-case and ER diagrams, we will design an efficient a dynamic database that can update all operations in real time.

# CLINIC MANAGEMENT SYSTEM

A Clinic Management System is an integrated information system for managing all aspects of a medical clinic's operations such as medical, financial, administrative, legal, and compliance. It includes electronic health records, business intelligence, and revenue cycle management. Medical health facilities improve the quality of healthcare services, reduce operating costs, and improve the revenue cycle by using such clinic management systems.

A Clinic Management System allows users to enter and track patients, schedule, and track patient appointments, send out insurance claims and patient statements as part of the collection process, process insurance, patient, and third-party payments, and generate reports for the administrative and clinical staff of the practice. eClinic systems also involves keeping up to date large sets of data including lists of diagnosis and procedures, lists of insurance companies, referring physicians, providers, facilities, and much more.

# REQUIREMENT ANALYSIS

## NEED FOR A SOFTWARE SYSTEM AT THE CLINIC

* + - The Clinic Management System allows easy access to patient data to generate various records, including classification based on demographic, gender, age, and so on. It is especially beneficial at the ambulatory point, hence enhancing continuity of care. Internet-based access improves the ability to access such data remotely.
    - It helps as a decision support system for the medical clinic authorities for developing comprehensive health care policies.
    - It efficiently engenders the running of finance, the diet of patients, and the distribution of medical aid. It gives a vivid picture of future clinic growth.
    - It enhances information integrity by a reduction in transcription errors and duplication of information entries.
    - Clinic Management System is easy to use and eliminates errors caused by handwriting.
    - It enhances the overall health care experience in a healthcare facility.
    - It reduces expenses of an organization because of less paperwork, improved safety, and reduced duplication of testing.

To sum up, the Dental Clinic Software System that we’re developing handles the following four functions on a day-to-day basis-

1. Registration
2. Scheduling
3. Consultation
4. Billing and Payment

## MODULES

* + 1. Patient Registration Module
       - Patient Registration module of our Practice Management System is designed to manage vital information for the patient information chart, which marks the outset of demographics capture. It encompasses the patient’s name, address and contact information, birth date, employer, and insurance information.
       - Unique patient ID for tracking visits
    2. Appointment and Scheduling
       - This module of Clinic Management System facilitates effective scheduling of appointments of patients for the doctors, laboratory, and radiology services.
       - Quick and effective patient scheduling
       - Online and offline appointment availability
    3. Reception Management
       - Management module manages front desk reception activities. Status of any patient/doctor can be queried from this module, e.g., timing of consultant, residential address/patient room search.
    4. Billing and Payment
* The billing and payment module handles all money-related transactions of the clinic. May it be just consultation fee, or any procedure performed or any kind of fee for any treatment done, the billing and payment module handles it.

## REQUIREMENTS OF THE PROJECT

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule.

Based on the above premise, there are

1. FUNCTIONAL REQUIREMENTS OF and
2. NON-FUNCTIONAL REQUIREMENTS
   * 1. Functional Requirements:

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it’s important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions. Coming to Dr. Reddy’s Dental Healthcare Management System, the following are the functional requirements:

* + - * Receptionists’ schedules (create/update) appointment for patients
      * Patient can select mode of payment (Cash/Card)
      * Patient will inform their details to book appointment
      * Receptionists retrieves previous patient records
      * Dentist can update / create patients problem record
      * CRUD operations can be performed by Receptionist as well as Dentists
      * Receptionists sends confirmation reminder to patient for appointment
    1. Non-Functional Requirements:

Non- functional requirements are system defined and necessary even if not specified in the logic or code of the program. This requirement specifies the services offered with the software application.

* + - * Appointments cannot be scheduled without Patient ID (Patient ID != Null)
      * No two patients can be appointed at same slots (based on Patient ID)
      * Reschedule/Cancel appointment should be done 24 hr prior to appointment
      * Receptionists should maintain correct record as per patients' information
      * Dentist must maintain his ethical standards to not affect patients’
      * Patients should explain accurate problem to Dentist
      * Dentist should be available at the time of appointment

# SPECIFICATIONS OF THE PROJECT

There are two key components to building any software project, namely,

1. Backend structure
2. Frontend

## Backend structure:

Back-end development comprises a site's structure, system, data, and logic. For this project, we have used the following to build the backend-

* + 1. IntelliJ Net Beans Version 16
    2. MySQL Version 8.0.31
    3. PHP
    4. XAMPP control panel 3.3.0
  1. Front-end structure:

Front-end development focuses on the visual aspects of a website – the part that users see and interact with. For this project, we have used the following frontend resources-

* + 1. HTML
    2. CSS

# USE CASE DIAGRAM

Use case diagrams are typically developed in the early stage of development and people often apply use case modeling for the following purposes:

* Specify the context of a system
* Capture the requirements of a system
* Validate a systems architecture
* Drive implementation and generate test cases
* Developed by analysts together with domain experts For this project, the following is the use-case diagram-

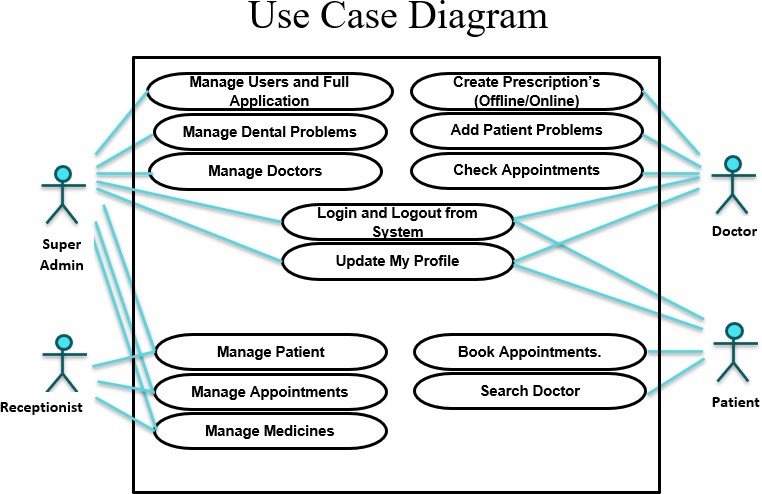
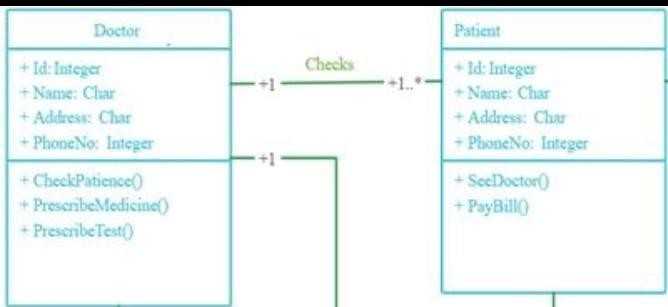
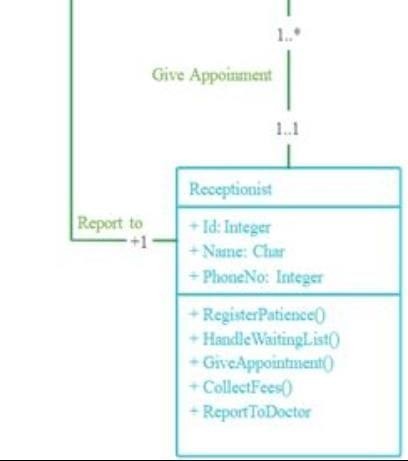


Figure1: Use Case Diagram

# CLASS DIAGRAM





Class diagrams are blueprints that are used to visualize, specify, and document structural features in our models. For example, during the analysis and design phases of the development cycle, we can create class diagrams to perform the following functions:

* Capture and define the structure of classes and other classifiers
* Define relationships between classes and classifiers
* Illustrate the structure of a model by using attributes, operations, and signals
* Show the common classifier roles and responsibilities that define the behavior of the system
* Show the implementation classes in a package
* Show the structure and behavior of one or more classes
* Show an inheritance hierarchy among classes and classifiers
* Show the workers and entities as business object models

# PROJECT IMPLEMENTATION

We have implemented this project using-

1. Java 8 for the programming and logic component
2. IntelliJ Netbeans 16 IDE for integration of frontend and backend
3. HTML and CSS for designing the frontend

The following is an overview of the classes that we implemented in one of the project modules-

|  |  |  |  |
| --- | --- | --- | --- |
| Project: DentalClinic | | | |
| Packages: | Main | Admin | Doctors |
| Classes: |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project: DentalClinic | | | | |
| Packages: | Patient | Receptionists | Images | Icon |
| Classes: |  |  |  |  |

Figure 2: Classes and Packages used in Project

## Admin

The admin package contains three main classes:

* + - AdminActivity- this class contains the logic and functionality behind the admin’s rights. It helps to track the activity of the admin’s transactions
    - ReceptionistManagement- this class defines the scope of the receptionist’s management authority
    - doctorManagement- this class defines the scope of the doctor’s management authority.

## Doctor

The doctor package contains the following classes:

* + - Appointment- used to schedule patient appointments
    - addDoctor- used to add a new doctor to the database
    - deleteDoctor- used to delete dentists from the database
    - doctorActivity- used to manage and track doctor activity on the portal
    - searchDoctor- helps to search doctors using their names
    - PatientDetails- used to view the details of the patient assigned to that doctor
    - ViewDoctor- used to view the list of doctors added to the database
    - doctorActivity-page to view the appointment list

## Main

This is the main package that has three main classes-

* + - Admin.java- This is the main admin class that contains the entire functionality of the admin
    - Hospital.java - contains various subclasses like doctor, patient, receptionist, and admin and certain functionalities.
    - Connector.java- This class is used to connect the java program to php mysql database
    - Doctor.java- This is the main login page for doctor
    - Receptionist.java- This is the main login page for receptionist

## Patient

* + - AddPatient.java – this class contains the logic and functions required to add a patient to the clinic database- for example, when a new patient calls up the reception, the receptionist takes the patient details and add them to the backend.
    - deletePatient.java- this class contains the logic and functions required to delete a patient from the database. For instance, if a patient requests for their information to be erased, then this function comes into play.
    - patientActivity.java- a class used to track and manage all patient activity.
    - searchPatient.java- this class helps to search for a patient in the clinic database by using their information such as, first name, last name, patient Id, etc.
    - updatePatient.java- this class enables the admin to update the patient information such as, add new appointments or reschedule existing ones, or update the case, etc.
    - viewPatient.java- by running this class, the user can view any of the patient’s info using their patientID.

## Receptionist

* + - AddReceptionist.java- this class contains the logic and functionality required to add a receptionist when they’re hired by the hospital.
    - DeleteReceptionist.java- this class consists of the code to delete a receptionist from the database.
    - searchReceptionist.java- this class helps to search for receptionist data in the clinic database by using their information such as, first name, last name, patient Id, etc.
    - receptionistActivity.java- helps to track and manage the transactions performed by a receptionist in the portal.

# Flow Diagram

Below is the flow diagram of the project where the Home Page is the start point of execution of the program and mainly the logins are as follow and their functionality is arranged in the form of flow diagram:

* + - Admin
    - Doctor
    - Receptionist

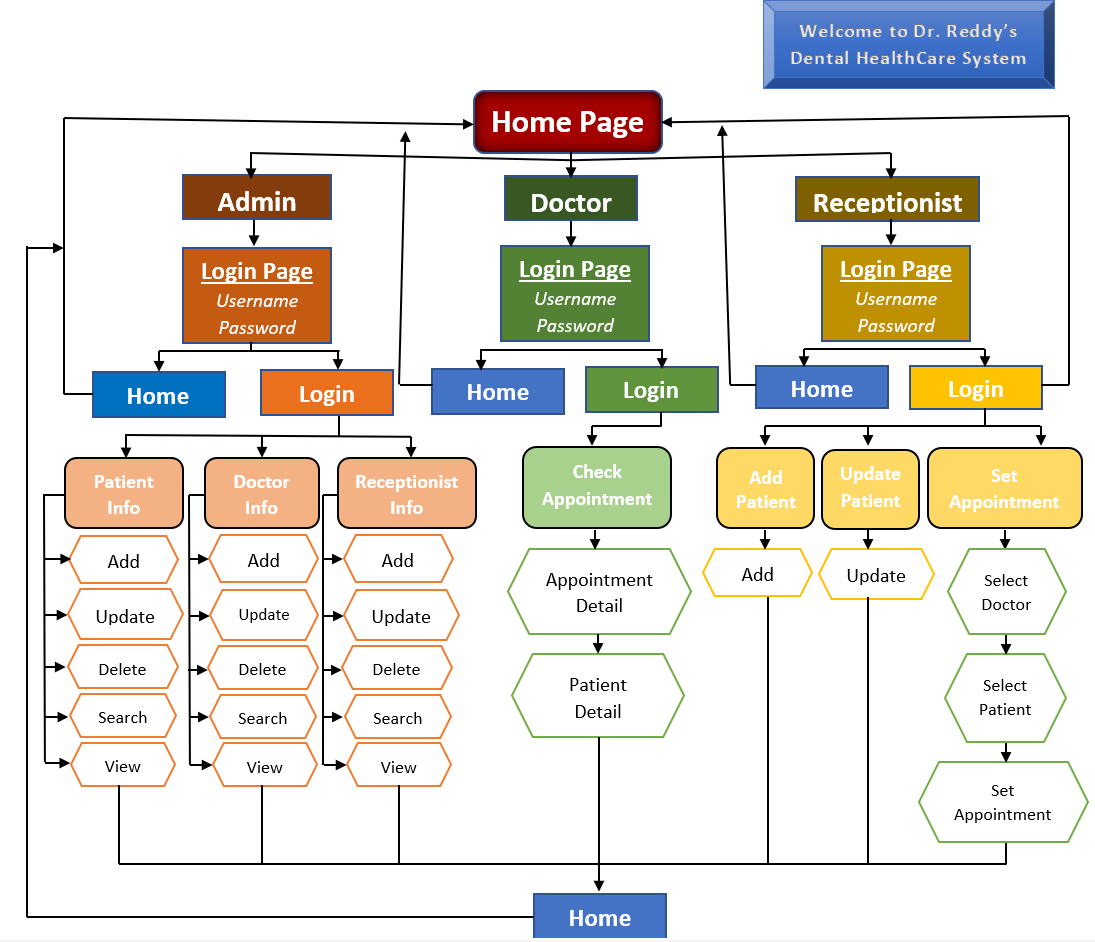
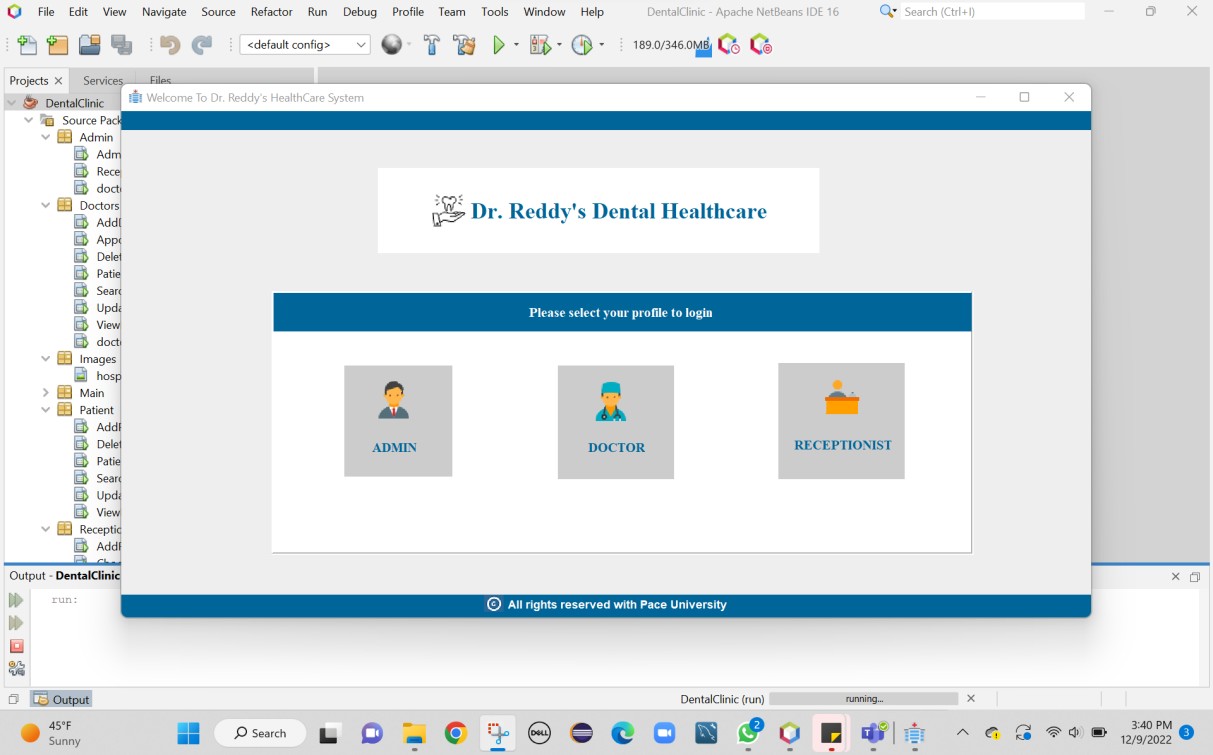


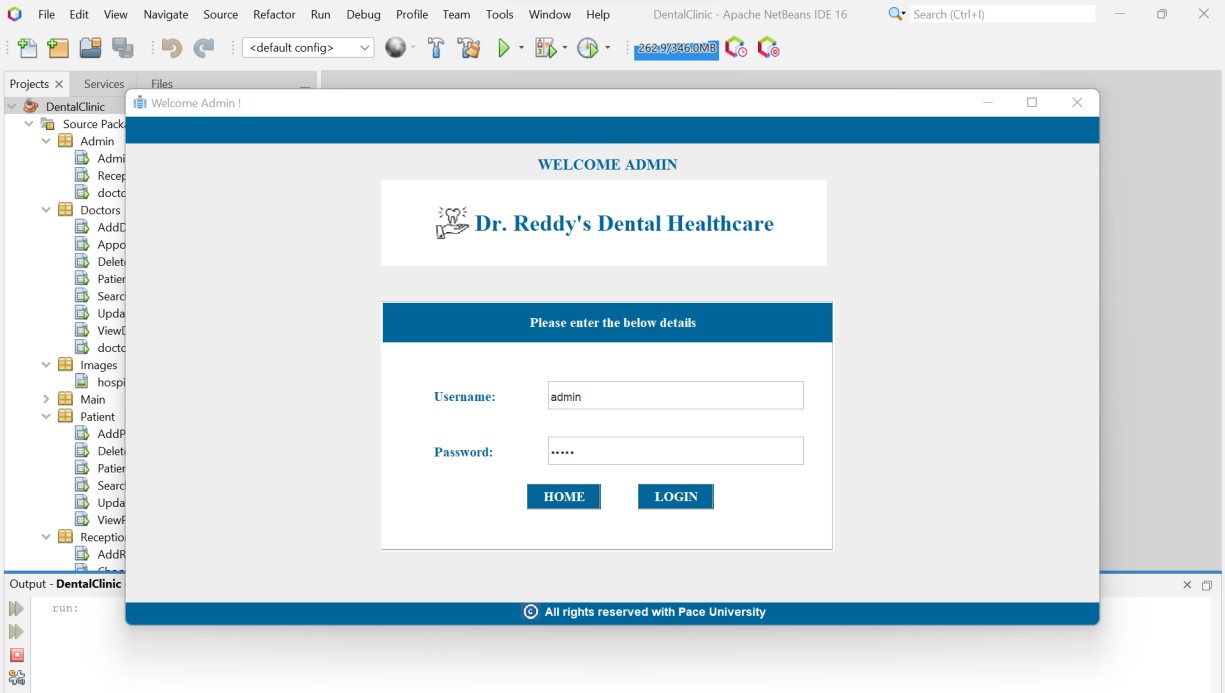
Figure 3: Flow diagram of the DentalClinic Project

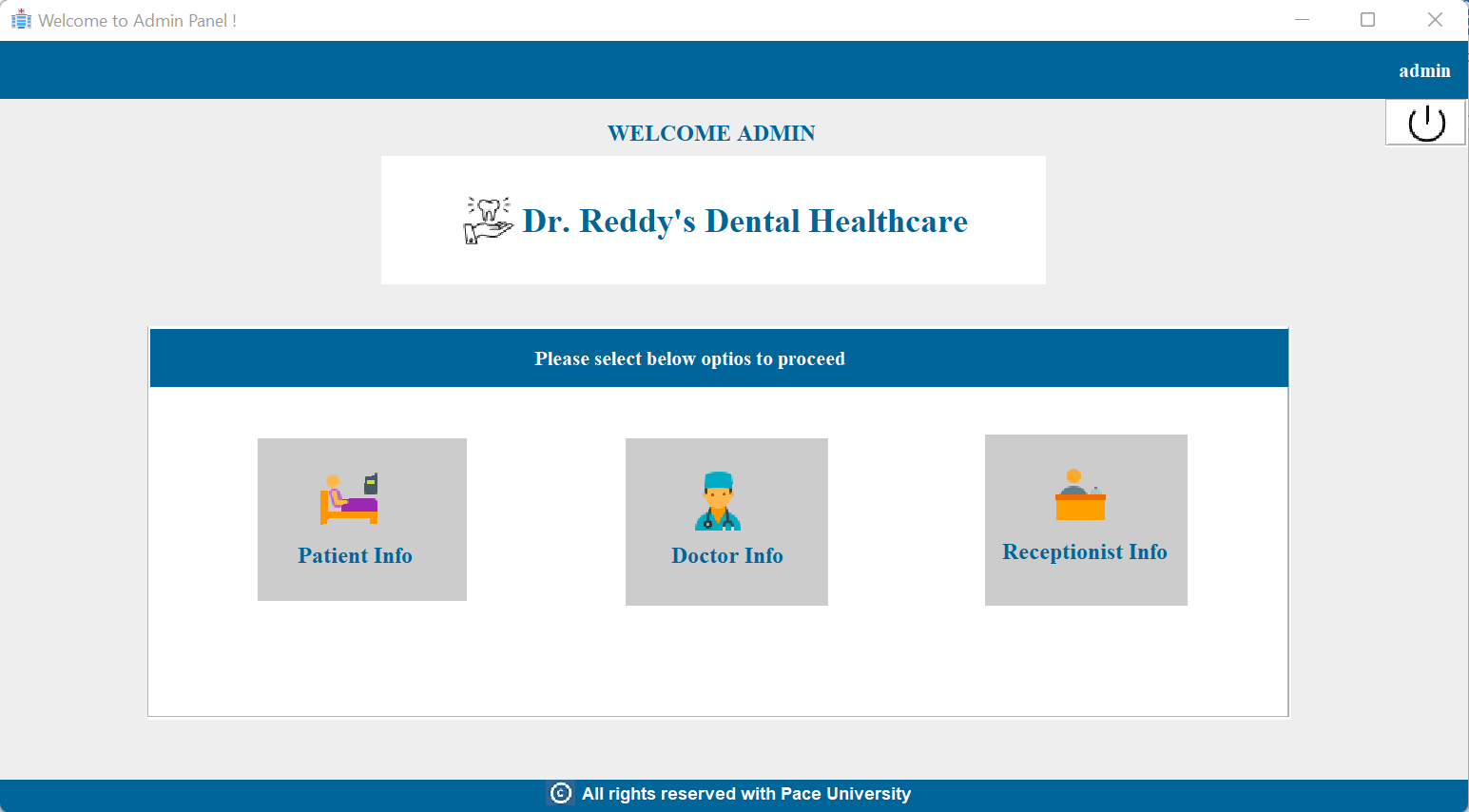
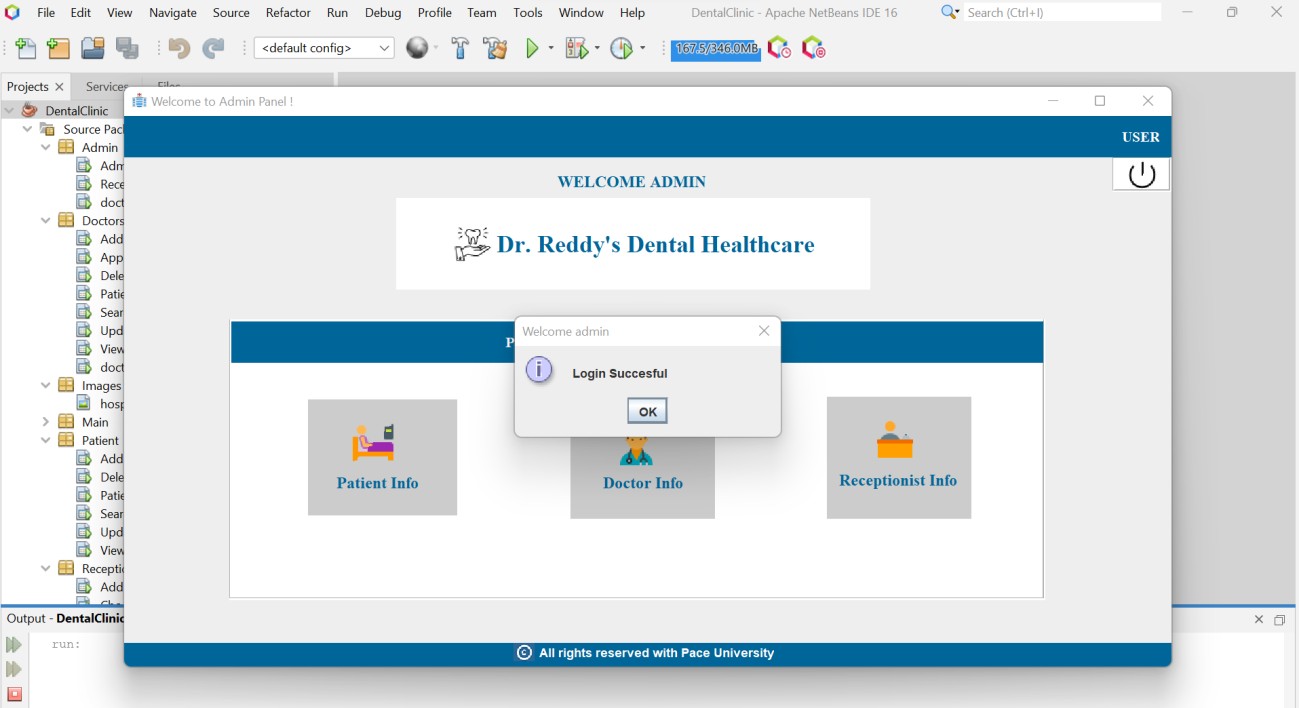
# SNAPSHOTS OF PROJECT EXECUTION

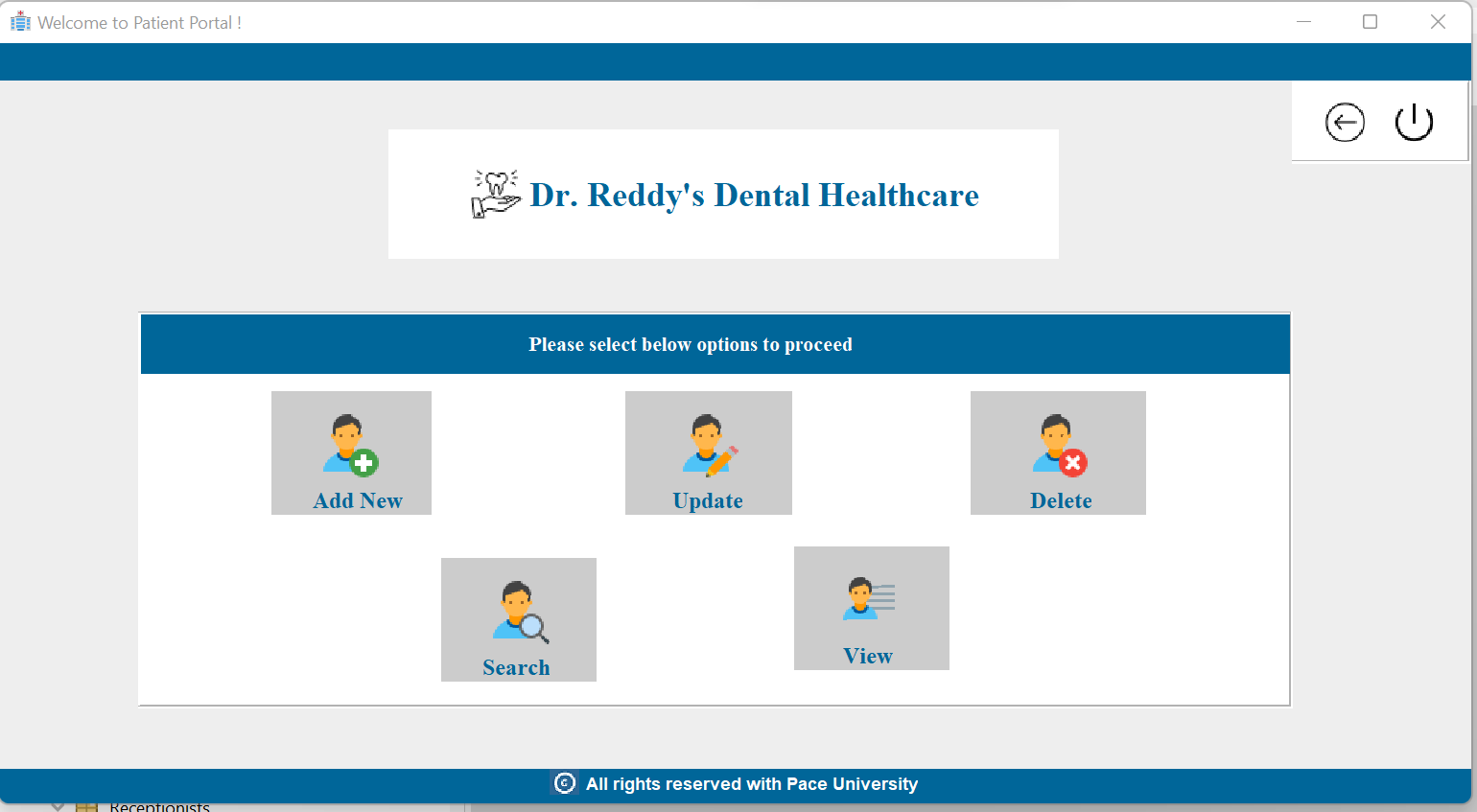
Steps to follow to run the program:

1. Run the project: User can view the home page of application
2. Click on ADMIN Panel: This will open the Login Page for ADMIN

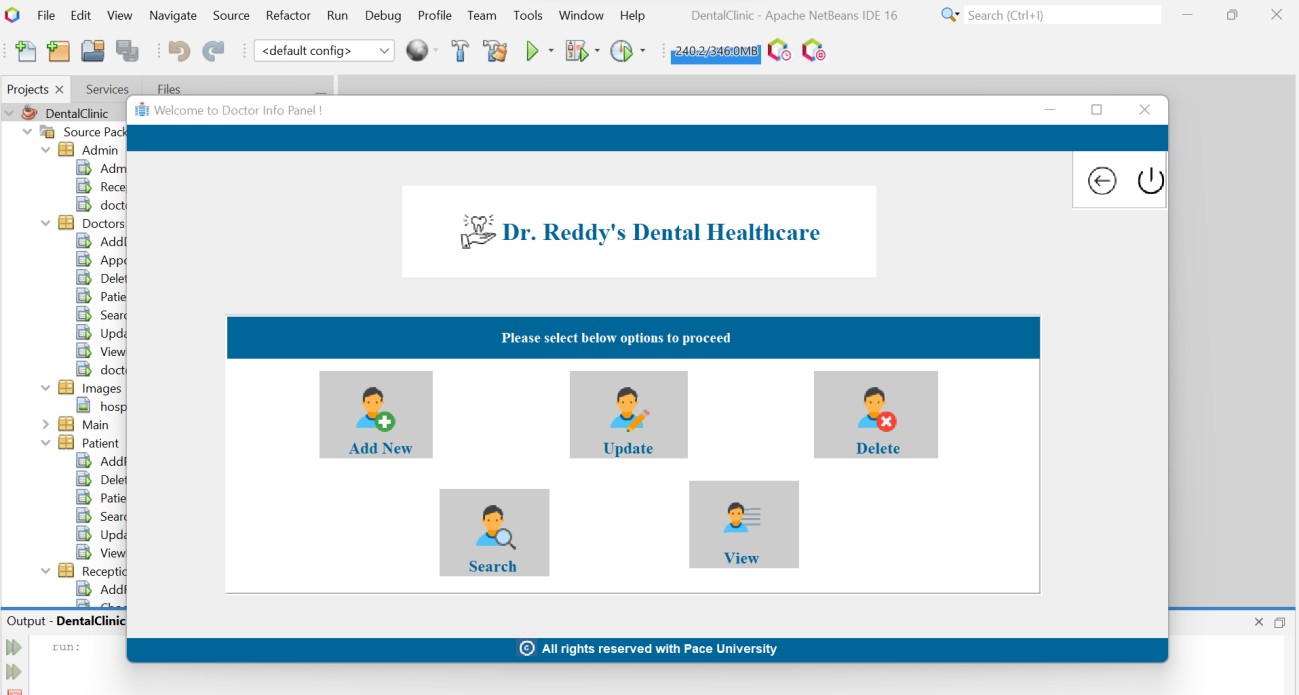
Enter Credentials as “Username: admin” and “Password: admin” specified in the database



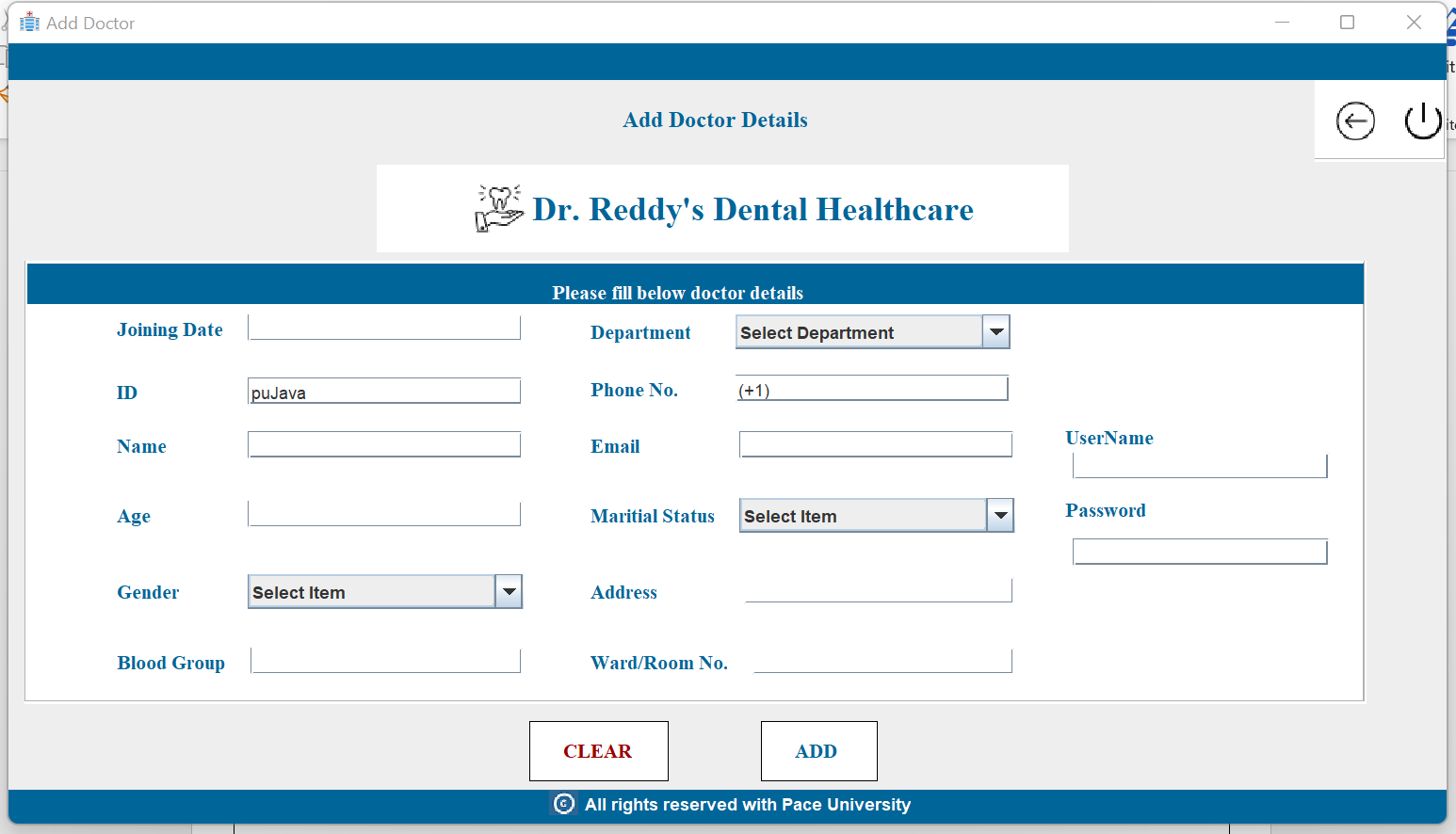
1. Once logged in successfully, admin can see below options
2. Click on Patient Info to perform actions related to patient



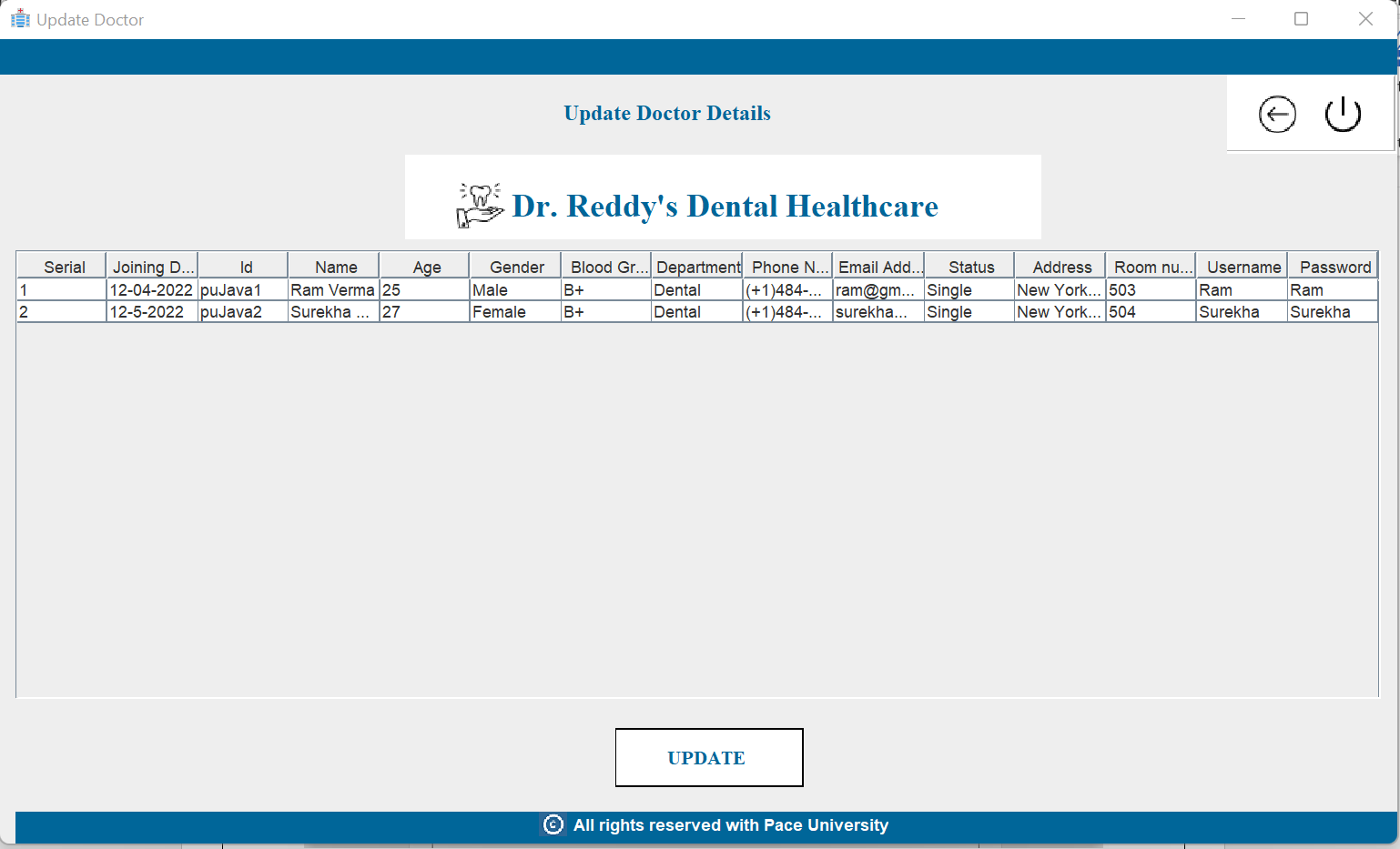
1. Similarly for Doctor Information admin can click on Doctor Info



1. Admin can click on Add New and add details of the doctor



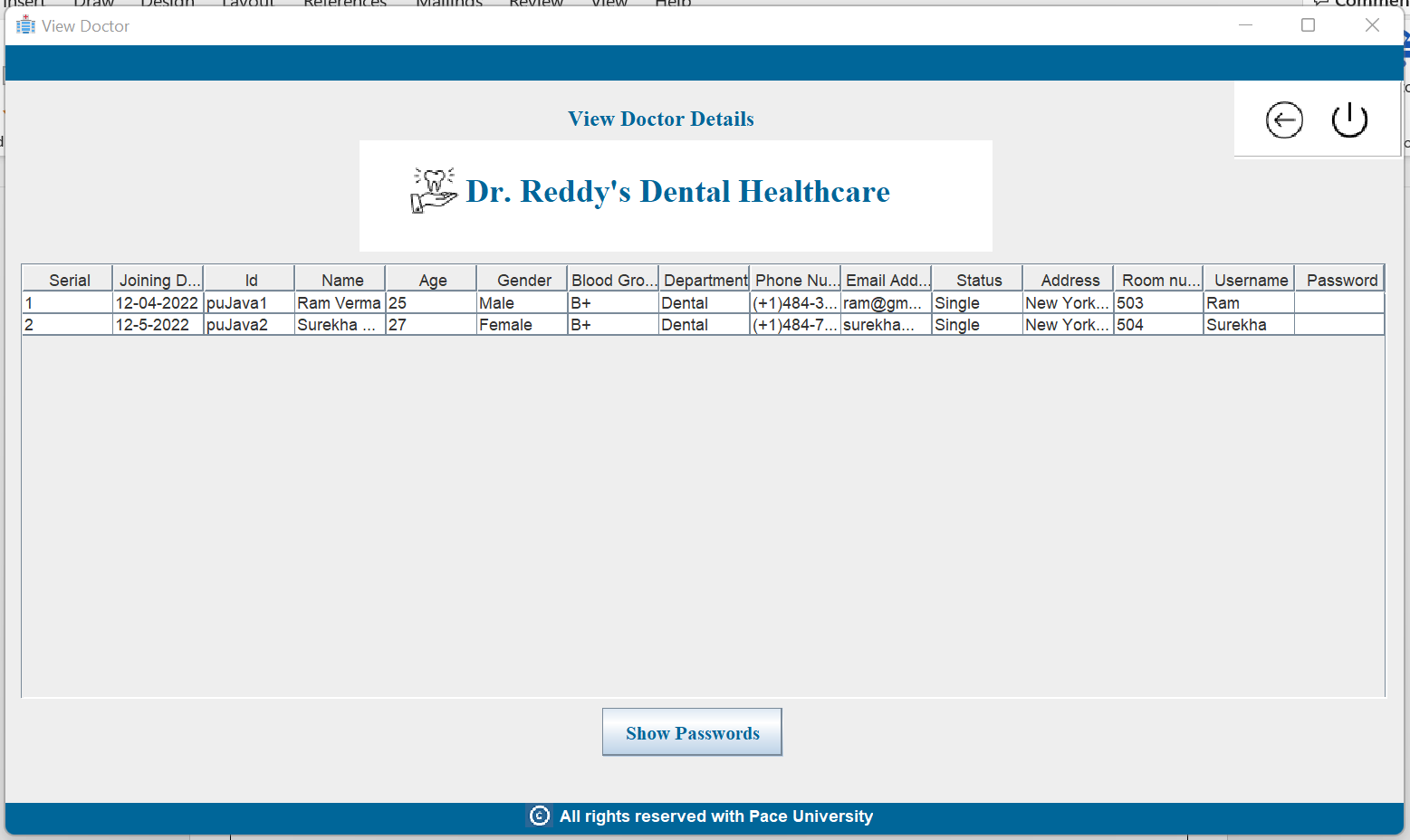
1. Admin can update the information of the doctor



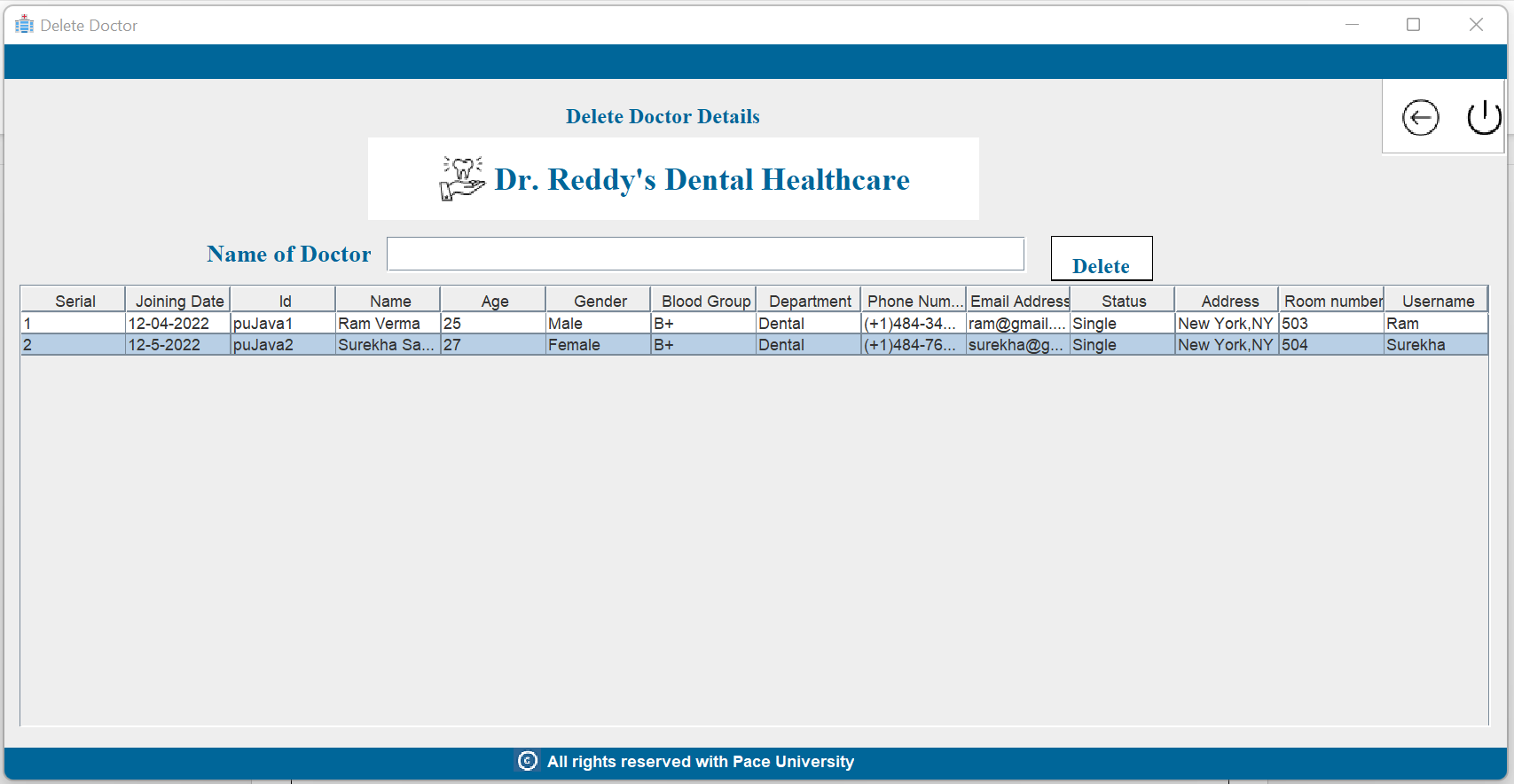
1. Admin can search doctors list



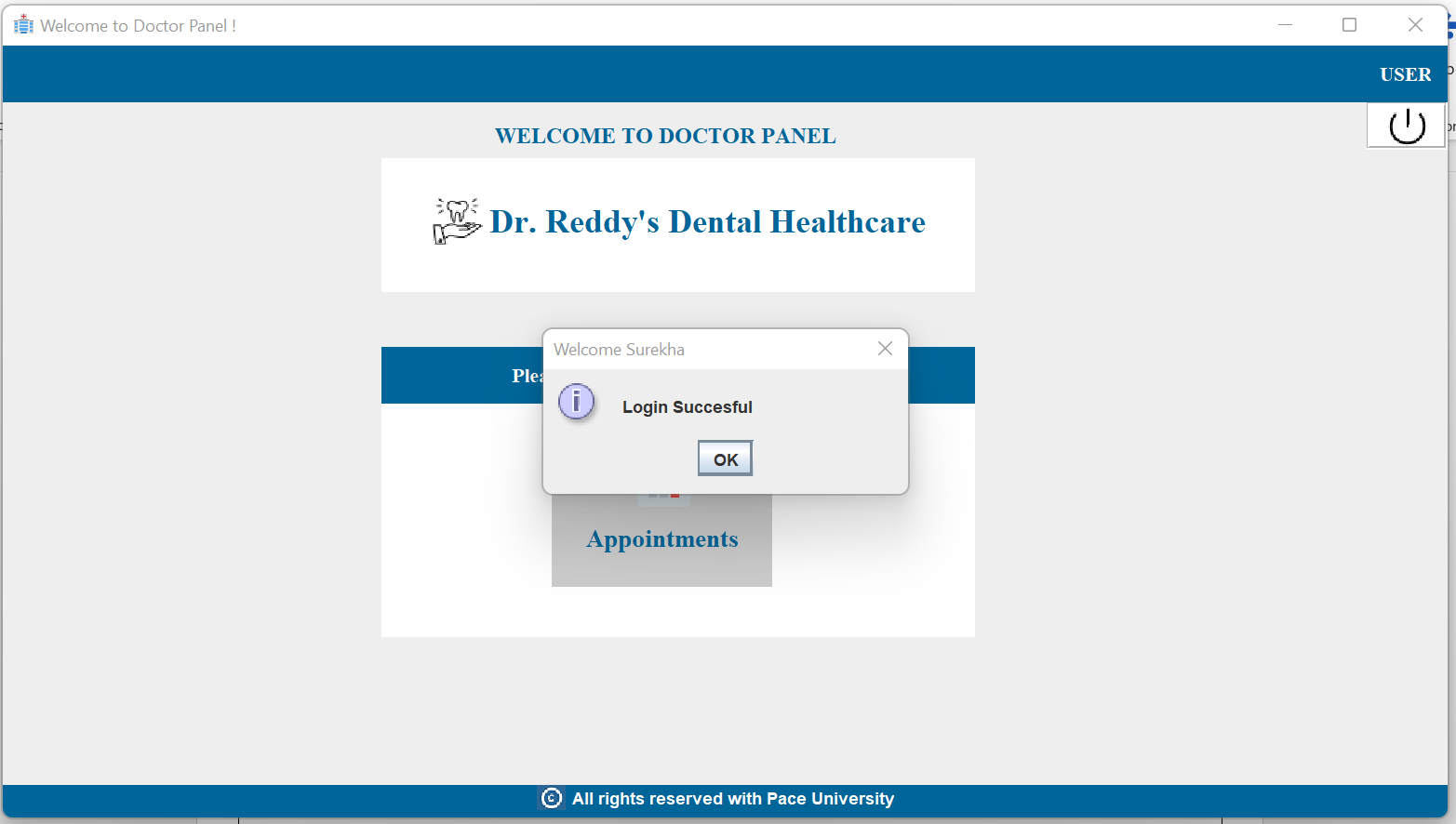
1. Admin can view the list of doctors



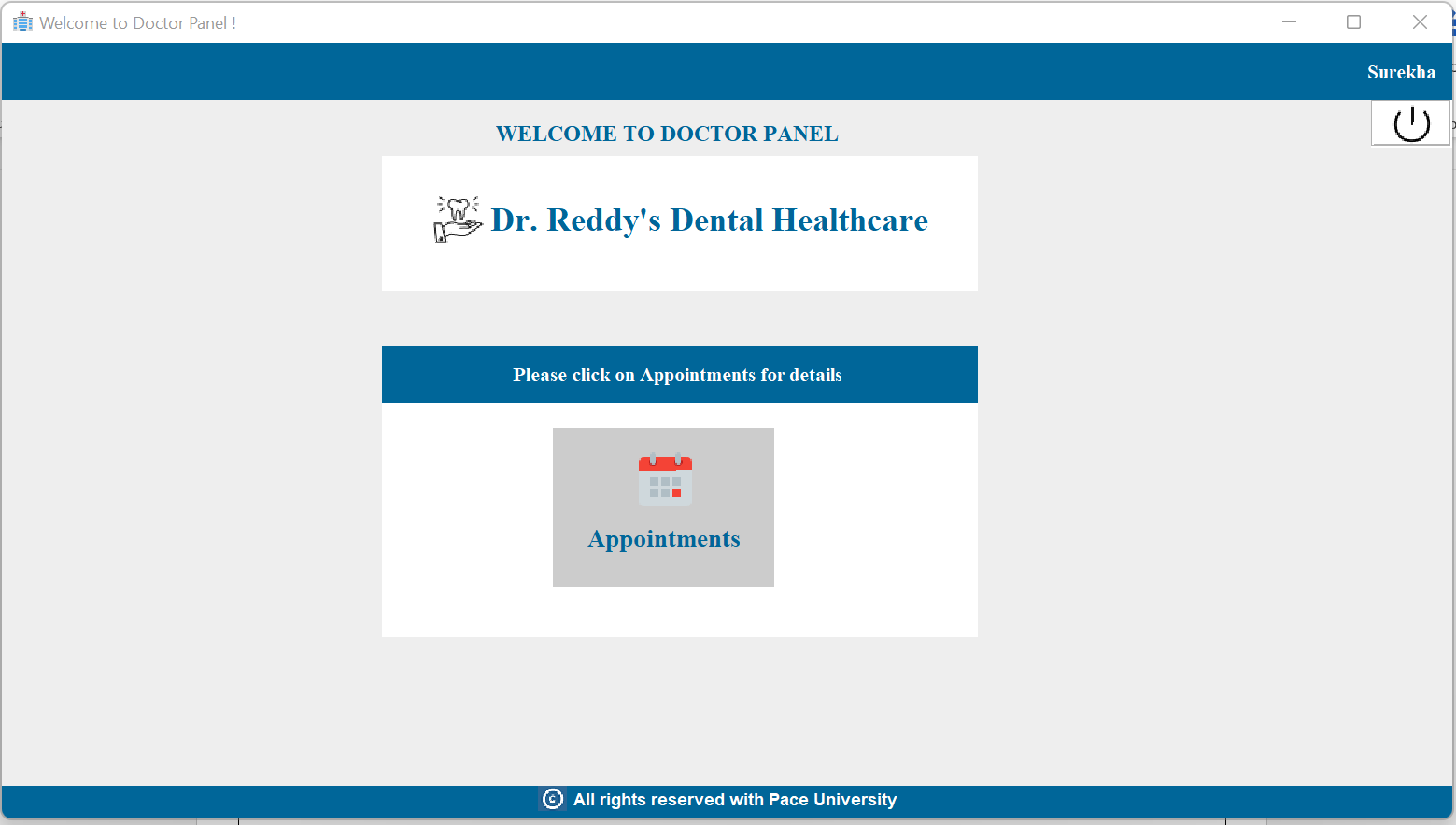
1. Admin can delete the list of doctors



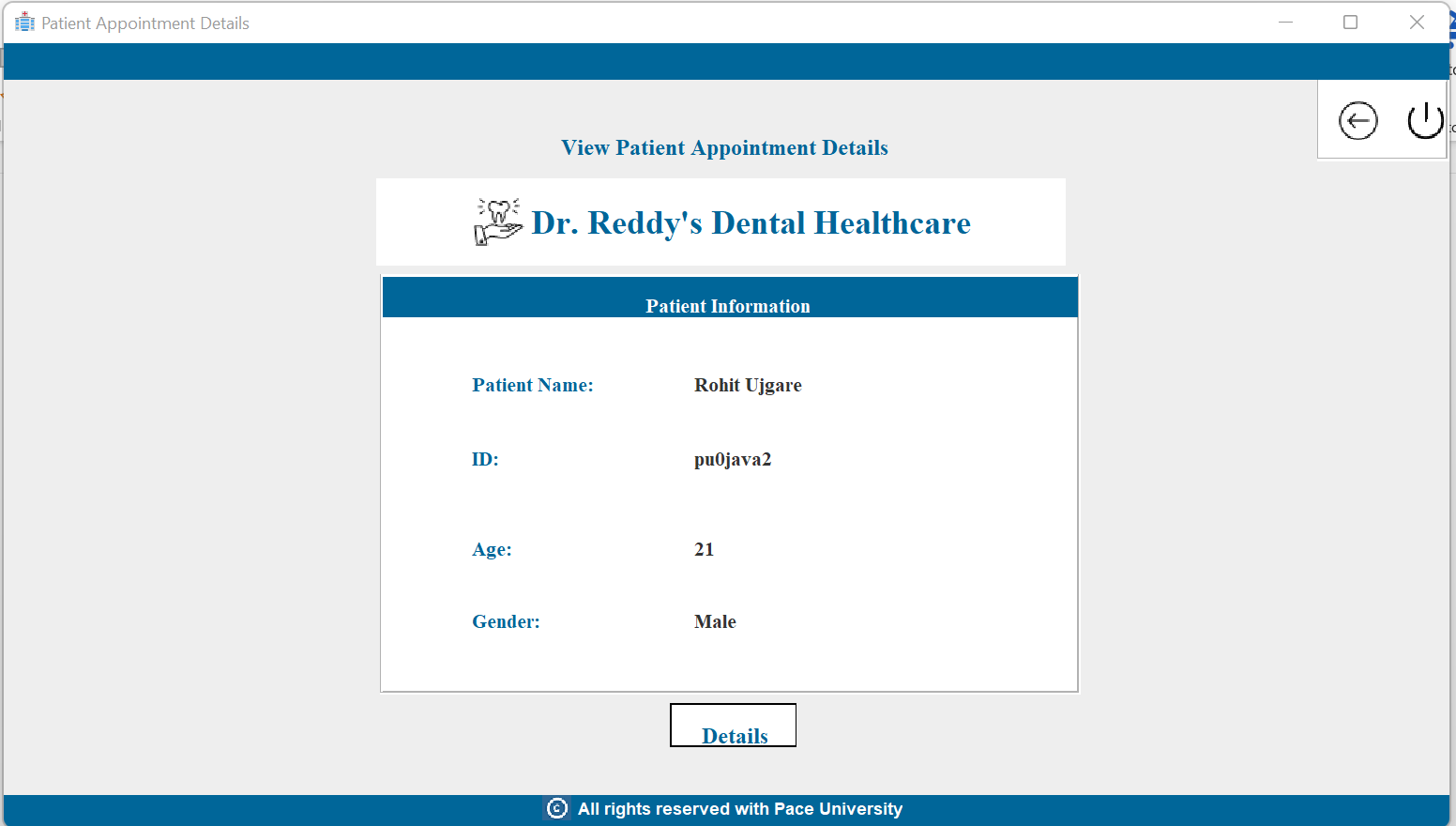
1. Go to home page and Login with Doctor Credentials: “Username: Surekha, Password: Surekha”



1. Doctor can view the appointment tab



1. Doctor can view the patient detail appointed to himself

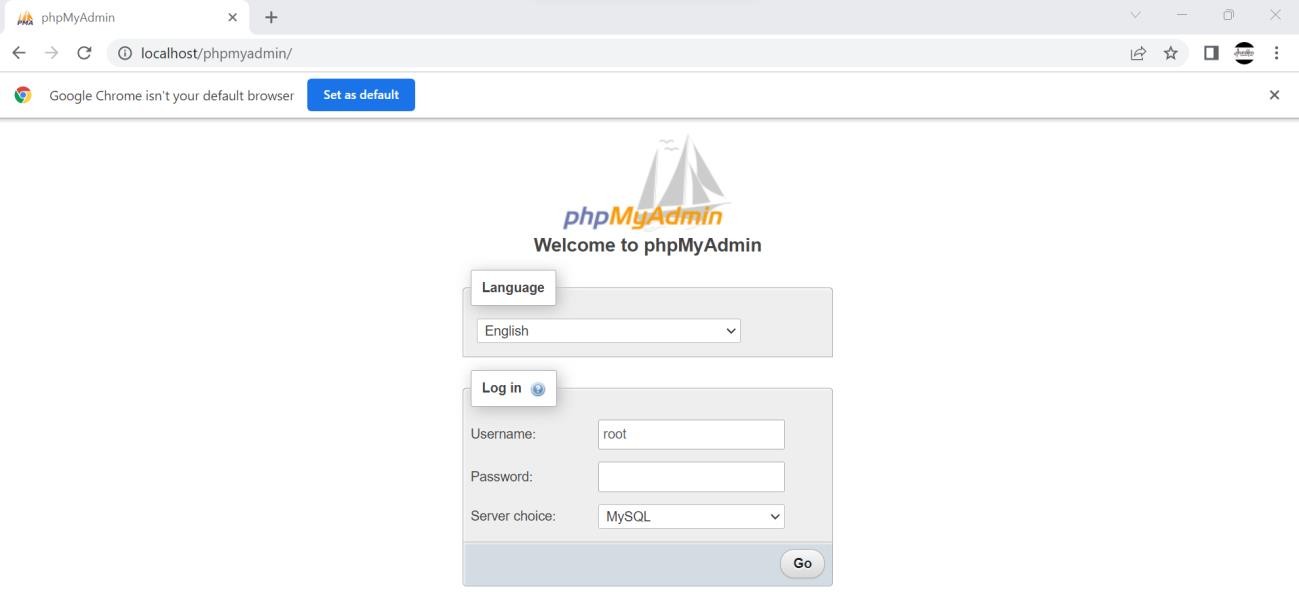




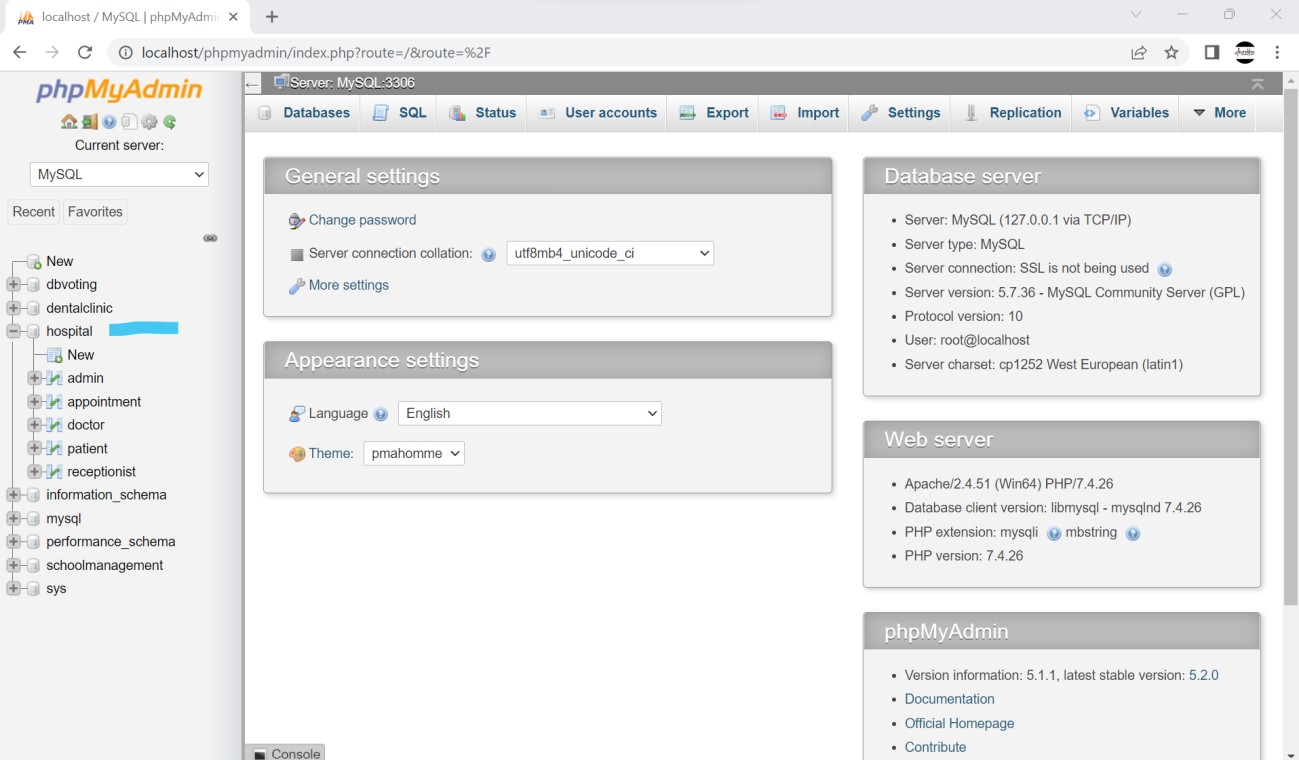
And so on…

# SNAPSHOTS OF PHP Database STRUCTURE

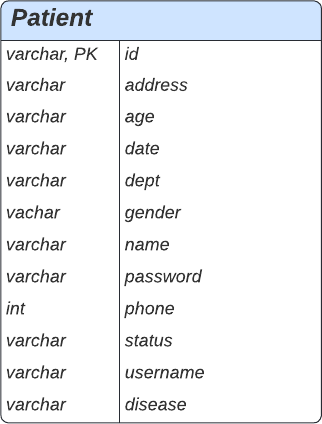
PHP is used to prepare the database for the project



We have created “hospital” database to connect with our project



Below are the attributes included in each table DATABASE: hospital

Table: ADMIN Table: PATIENT

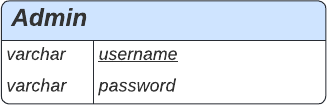


Table: APPOINTMENT

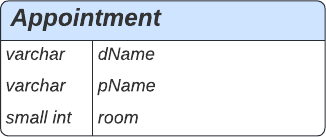


Table: DOCTOR Table: RECEPTIONIST

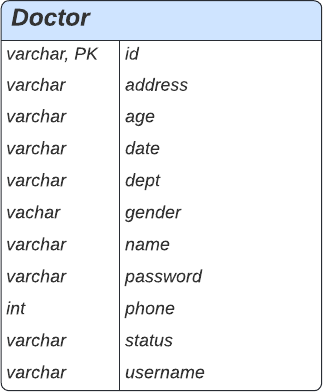


Figure 4: List of tables and attributes used in the project

# CONCLUSION

To sum up, we have built a working prototype of Dr. Reddy’s Dental Healthcare System to help the dentists and administration keep track of the day-to-day operations of the clinic. Now, instead of logging all information manually, the administration will be able to add new patients, delete existing ones, schedule, and reschedule appointments, log and record all patient information for future reference, manage doctor schedules, schedule routine treatments or surgeries, and much more.

In this particular version though, we were able to complete the Admin and Doctor modules of the project, and work is in progress for the remaining two modules. Using our Java knowledge that we gained over the semester, we were able to code and program the presented modules to apply our theoretical knowledge in real time.

During the course of the project, we were also able to sharpen our coding skills, learn design scripts such as HTML and CSS, and gained fundamental knowledge about server operations.

Finally, we would like to express our sincere gratitude to our Professor. Sukla Saha for her constant guidance and support during the course of our project, and for providing us the opportunity and resources to learn something new.

References-

<https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams>

<https://help.one.com/hc/en-us/articles/115005585509#log>

<https://dev.mysql.com/downloads/connector/j/>