A PROJECT ON

**Veterinary Clinic Management System**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC



**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY**

Hinjawadi

**SUBMITTED BY:**

Ahire Bhavesh Sunilbhai, Gade Onkar Popat,

Anubhav Tewari,

Samarth Cherian

**UNDER THE GUIDENCE OF:**

Ms.Lalita Shinde

Faculty Member

Sunbeam Institute of Information Technology, Pune

**ACKNOWLEDGEMENT**

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Ahire Bhavesh Sunilbhai,

Onkar Gade Popat,

Anubhav Tewari,

Samarth Cherian

SIIT Pune

**A PROJECT ON**

**“Veterinary Clinic Management System”**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE COURSE OF

DIPLOMA IN ADVANCED COMPUTING FROM CDAC



**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY**

Hinjawadi

**SUBMITTED BY:**

Ahire Bhavesh Sunilbhai, Gade Onkar Popat,

Anubhav Tewari,

Samarth Cherian

**UNDER THE GUIDENCE OF:**

Ms.Lalita Shinde

Faculty Member

Sunbeam Institute of Information Technology, PUNE.

**CERTIFICATE**

This is to certify that the project work under the title ‘**Veterinary Clinic Management System**’ is done by Ahire Bhavesh Sunilbhai,Gade Onkar Popat,Anubhav Tewari,Samarth Cherian in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

**Ms.Lalita Shinde**

**Project Guide**

Date: 12-02-2025

**Mr. Yogesh Kolhe Course Co-Coordinator**

Table of Contents:

1. INTRODUCTION 2
2. [REQUIREMENTS 3](#_TOC_250005)
   1. [Functional Requirements 3](#_TOC_250004)
   2. Customer Module 4
   3. Vendor Module 6
   4. Delivery Boy Module 9
   5. Admin Module 11
3. [Non-Functional Requirements 12](#_TOC_250003)
   * 1. Hardware and Software Interfaces 12
4. [DESIGN 13](#_TOC_250002)
   1. [Database design 13](#_TOC_250001)
5. CODING STANDARD IMPLEMENTED 19
6. [TEST REPORT 22](#_TOC_250000)
7. APENDIX A
8. Entity Relationship Diagram 27
9. Data Flow Diagram 28
10. Class Diagram 29
11. APENDIX B

UI Screenshots 30

1. REFERENCES

# INTRODUCTION TO PROJECT

The **Veterinary Clinic Management System** is an application that helps to manage the day-to-day activities of a clinic, like adding appointments,managing patient records, prescriptions, and billings.The system ensures efficient management of veterinary services by automating administrative tasks, reducing paperwork, and improving overall workflow. It includes functionalities like digital medical records for pets, invoice generation, and secure data storage.

The Veterinary Clinic Management System is an application that helps manage day-to-day activities of a clinic, such as adding appointments, managing patient records, prescriptions, and billings. Built using Spring Boot for backend logic, REST APIs, and secure data processing, it uses JWT authentication for secure login and access control. The front end is developed with React for dynamic user interfaces, styled with Bootstrap to enhance design and responsiveness. Data is securely stored using MySQL as the database for reliable record handling. The system ensures efficient management of veterinary services by automating tasks, reducing paperwork, and improving workflow. It includes functionalities like digital medical records for pets, invoice generation, and secure data storage.

# REQUIREMENTS

# FUNCTIONAL REQUIREMENTS

# 

## Admin Flow

* + 1. **Home Page**
* **Objective**: Display a list of Doctors,Receptionists,Add Staff

## Features:

* + Click to view List of Doctors,Receptionists,and with functionality to add staff

## Doctors

* **Objective**: Allow admin to view and manage Doctors

## Features:

* + Click on a Doctors to see the list of Doctors.

## Receptionists

* **Objective**: Allow admin to view and manage Receptionists.

## Features:

* + Click on Receptionists to see the list of Receptionists.

## Add Staff

* **Objective**: Allow admin to view and manage Staff of the clinic.

## Features:Admin can add or delete staff by adding their email and password.

## Sign out

* **Objective**: Just to sign out.

## Profile Page

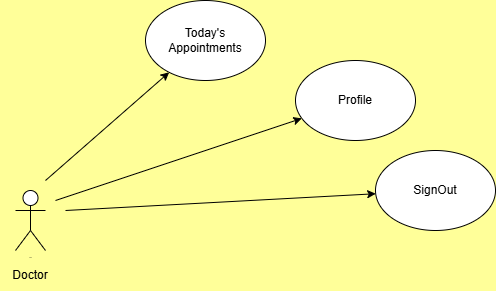
* **Objective**: Manage user profile and view order history.

## Features:

* + **If Not Logged In**:
    - Display a sign-in prompt.

## If Logged In:

* + - View and update profile details.
    - View order history and addresses.
    - Submit reviews for orders.
    - Logout.



## Doctor Flow

* + 1. **Home Page**
* **Objective**: Allow Doctors to manage their appointments.

## Features:

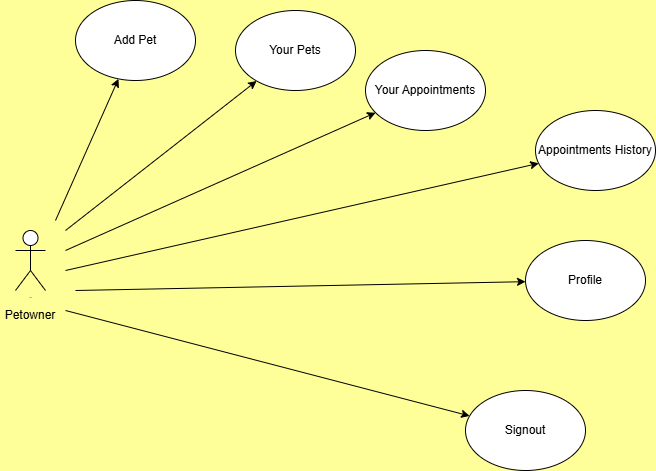
* + Displays the list of upcoming appointments.

## Vendor Home Page

* **Objective**: Manage menu items and view orders.

## Features:

* + Add menu items in breakfast, lunch, and dinner categories.
  + View placed and delivered orders.
  + Update or delete menu items.
  + Logout.

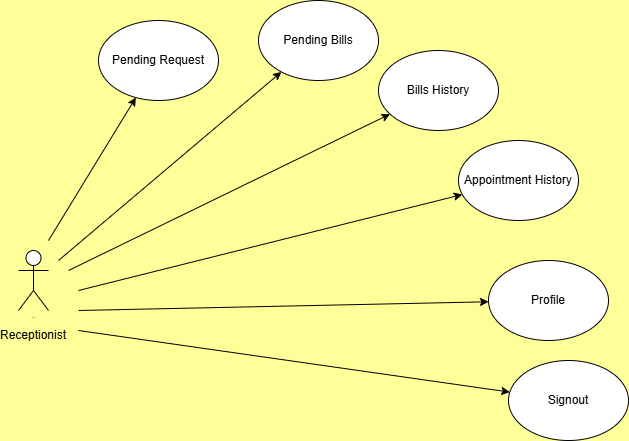


## Delivery Boy Flow

* + 1. **Home Page**
* **Objective**: Manage delivery boy account.
* **Features**:
* **Sign up and sign in functionalities.**
  + 1. **Delivery Boy Home Page**
* **Objective**: Manage assigned and delivered orders.

## Features:

* + View and assign placed orders.
  + Mark Placed Order as Delivered after delivering the order.
  + View delivered orders.
  + Logout.



## Admin Flow

* + 1. **Home Page**
* **Objective**: Admin access to system functionalities.

## Features:

* + Sign in to access admin functionalities.

## Admin Home Page

* **Objective**: Oversee and manage system operations.

## Features:

* + View counts of customers, vendors, and delivery boys.
  + View lists of customers, vendors, and delivery boys.
  + View orders and customer reviews/feedback.

## Non-Functional Requirements

* 1. **Interface**
* User interfaces must be intuitive and user-friendly. Detailed designs are provided in Appendix B.

## Performance

* **Number of Concurrent Users**: The system should handle at least 1000 transactions/inquiries per second.
* **System Resilience**: The application should be resilient to temporary server failures.

## Constraints

* The system should maintain performance standards of handling 1000 transactions/inquiries per second.
  1. **Other Requirements**
     1. **Hardware Interfaces**

**Requirements: Intel Core i5 or higher (or AMD equivalent),**

**8 GB RAM, 512 GB SSD or larger.**



## Software Interfaces

* **Operating Systems**: MS Windows 11, Ubuntu 22.04.
* **Database**: MySQL.
* **Server**: Embedded Tomcat.
* **Browsers**: Compatible with modern web browsers.

## System Design

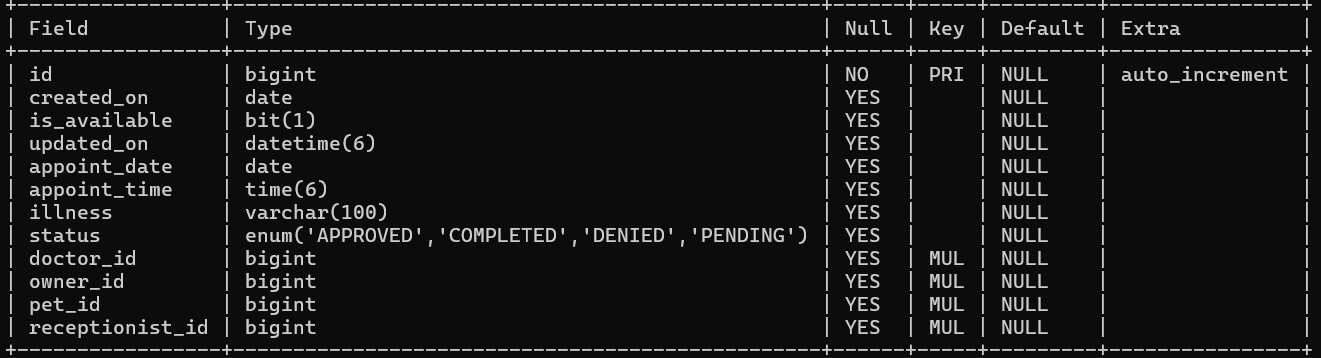
* 1. **Architecture**
* **Front-End**: Developed using React.js and Redux for state management.
* **Back-End**: Built with Spring Boot for server-side logic.
* **Database**: MySQL for storing user data, orders, and other system information.
* **Server**: Embedded Tomcat for hosting the application.

# DESIGN

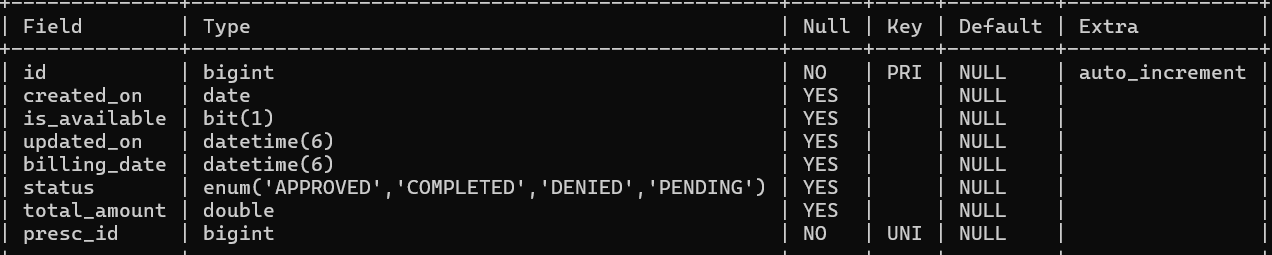
## Database Design

The following table structures depict the database design.

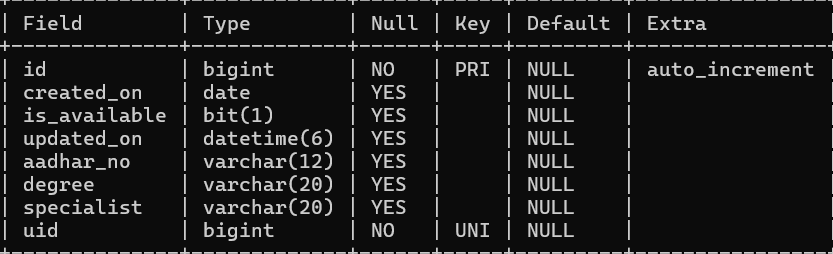
**Table 1: Appointments**

****

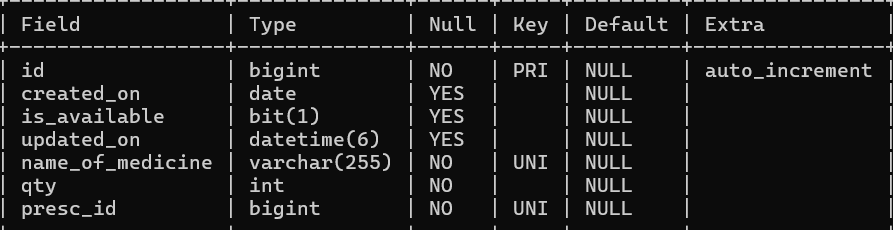
**Table 2: Billings**

****

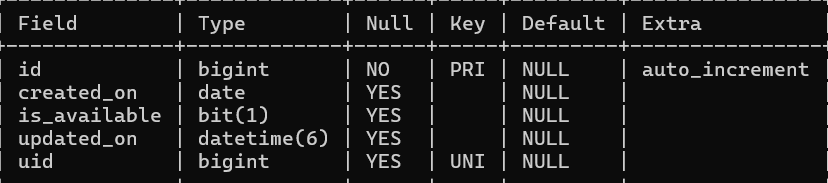
**Table 3: Doctors**

****

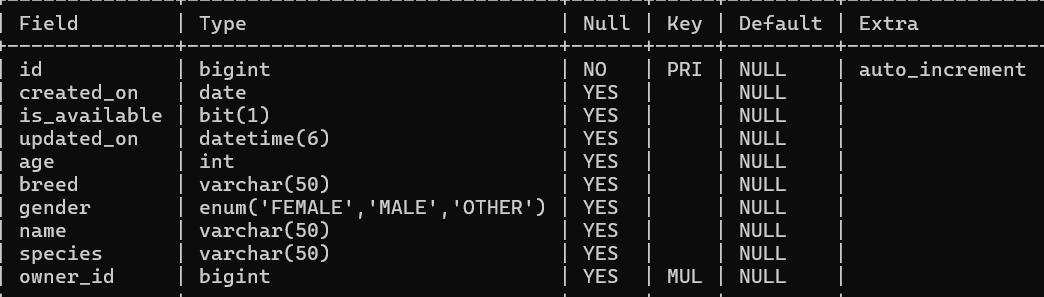
**Table 4:Medicines**

****

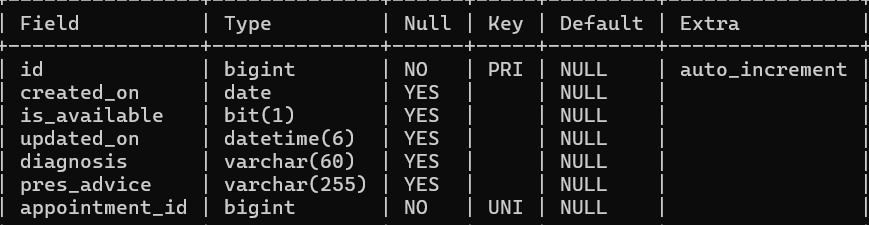
**Table 5:Pet Owners**

****

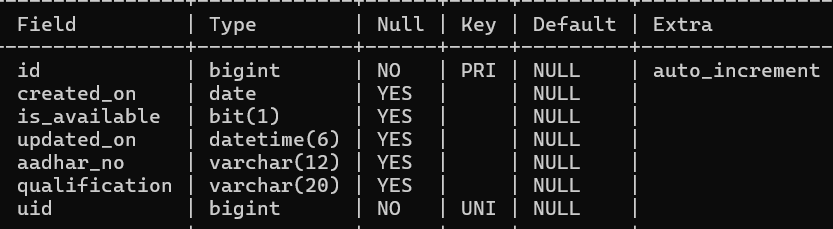
**Table 6:Pets**

****

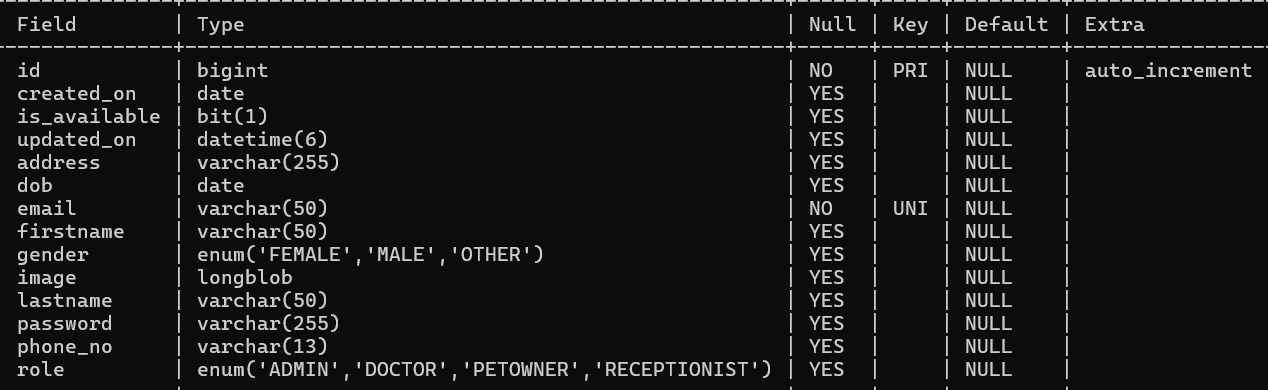
**Table 7:Prescriptions**

****

**Table 8 : receptionists**

****

**Table 9: Users**



**E-R Diagram,Dataflow diagram and Class Diagram:**

Go to Appendix A

# CODING STANDARDS IMPLEMENTED

## Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | User, Order, UserController | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type  prefixes like ‘C’ for class. |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Camel | SignUp, SignIn,  addReviews | Methods should use **verbs** or verb  phrases. |
| Parameter | Camel | firstName, lastName, email, password | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in  most scenarios. |
| Interface | Pascal with "I" prefix | UserRepository, OrderRepository  ,  MenuRepository | Do not use the ‘\_’ sign |
| Annotation | Pascal | SpringBootAppli  cation | Use @ at start of annotation |
| DTOs | Camel | ApiResponseDTO, SignUpReqDTO, OrderDetailsRes DTO | Use to transfer data between the processes |
| Exception Class | Pascal with "Exception"  suffix | ResourceNotFoun dException |  |

## Comments

* Comment each type, each non-public type member, and each region declaration.
* Use end-line comments only on variable declaration lines.

End-line comments are

comments that follow code on a single line.

* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

# TEST REPORT

**Another group called Linux did the testing and the report of the testing is given hereunder.**

# GENERAL TESTING:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR-NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
| 1 | SignUp Page | Signup  successfully message | OK | Nothing |
| 2 | SignIn Page | Pop-up will come | Ok | Please enter username and password again . |
| 3 | HomePage | Vendor List fetched from db | Ok | Failed to fetched vendor list |
| 4 | Menu List page | Gives all menu item for selected  vendor | Ok | Nothing |
| 5 | Cart Page | Give a list of  added item in the cart | Ok | Please log in first |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | Customer Order history | Order history list render successfully | Ok | Failed to fetched orders |
| 7 | Add Menu Item | Menu added successfully | Ok | Nothing |
| 8 | See placed orders by vendor | Placed order list view | Ok | No orders is  placed |
| 9 | Generate Token after first  login | Token generated successfully | Ok | Failed to generate token |
| 10 | View all  users list by admin | Seeing the list of all users | Ok | Nothing |
| 11 | Placing Order by customer | Order placed successfully and redirect to home page | Ok | Nothing |
| 12 | Logout | It will logout from user profile. | Ok | Nothing |
|  | **STATIC TESTING** |  |  |  |
| **SR-NO** | **Deviation** | **Program** |  |  |
| 1 | Commenting not followed | All Web Application |  |  |

1. **PROJECT MANAGEMENT RELATED STATISTICS**

## DATE WORK PERFORMED SLC PHASE Additional Notes

Project Allotment and User Requirements Gathering

|  |  |
| --- | --- |
| Sept 2024 | 11, |
| Sept 2024 | 17, |
| Sept | 30, |

Initial SRS Document Validation and Team

Feasibility Study

Requirement Analysis

Our team met the client Mr. Nitin Kudale (CEO, SIIT Pune) to know his requirements.

The initial SRS was presented to the client to

Structure Decided

(Elicitation) understand his requirements

better.

Designing the use- Requirement Database Design completed.

## DATE WORK PERFORMED SLC PHASE Additional Notes

2024 cases, Class Diagram, Collaboration Diagram, E-R Diagram, and User Interfaces

Business Logic Oct 5, 2024 Component Design

Started

Analysis & Design Phase

Design Phase

---------------------------

-

Oct 15, 2024 Coding Phase Started Coding Phase 70% of Class Library

implemented.

Oct 30, 2024

Implementation of Web Application and Window Application Started

Coding Phase

Class Library Development going on.

Nov 16, 2024 Off Off Off

Dec 1, 2024

Dec 15, 2024

Implementation of Web Application and Window Application Continued

|  |  |  |
| --- | --- | --- |
| Dec 2024 | 24, | After Ensuring Proper Functioning the Required Validations were Implemented |
| Dec | 30, | The Project was Tested by the respective Team |
| 2024 |  | Leaders and the Projec Manager |
|  |  | The Project was Submitted to Other |

Implementation of Web Application and Window Application Continued

Coding Phase and Unit Testing

Coding Phase and Unit Testing

Coding Phase and Unit Testing

Testing Phase (Module

Class Library Modified as per the need.

---------------------------

-

Module Integration was done by the Project Manager

---------------------------

Testing) -

|  |  |  |
| --- | --- | --- |
| 2025 |  | Project Leader of Oth  Project Group For Testing |
| Jan 17 2025 | , | The Errors Found were Removed |
| Feb 2025 | 12, | Final Submission of Project |

Jan 2

Testing Phase The Project of Other Team

er (Acceptance was Taken up by the Team Testing) for Testing

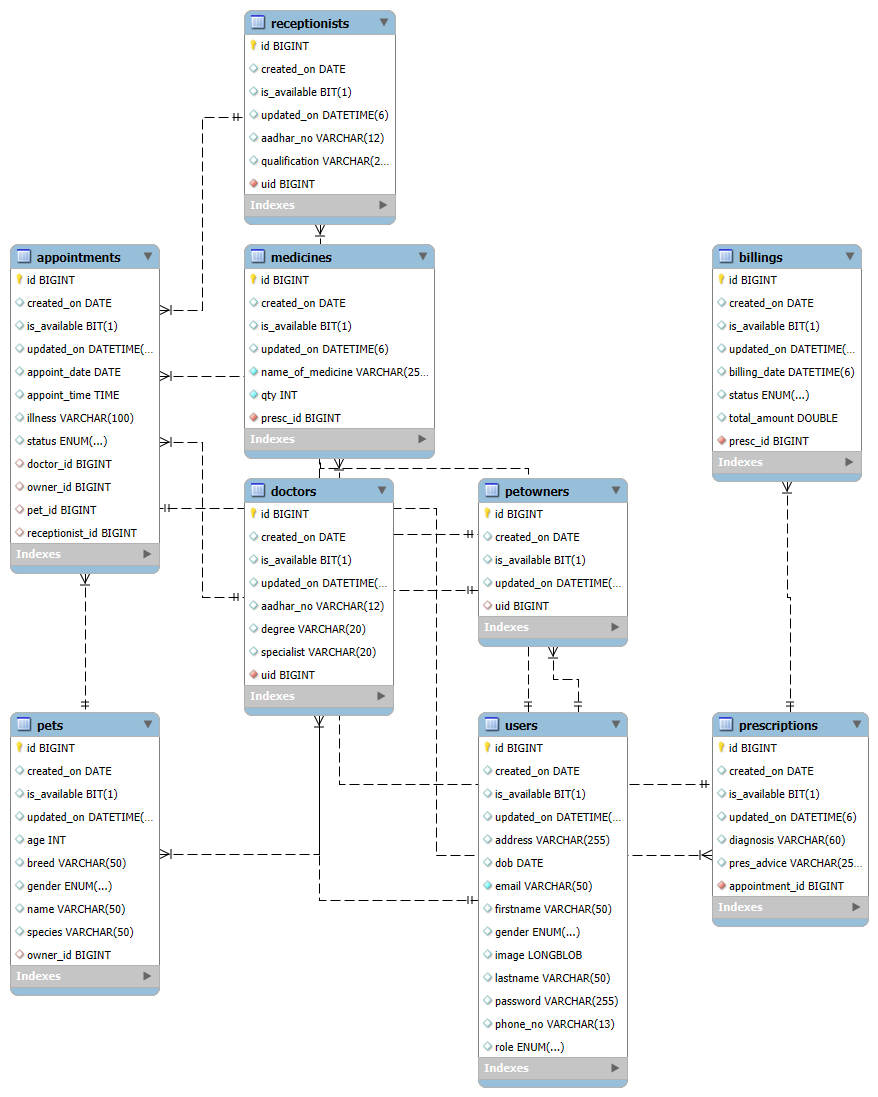
Debugging The Project was complete

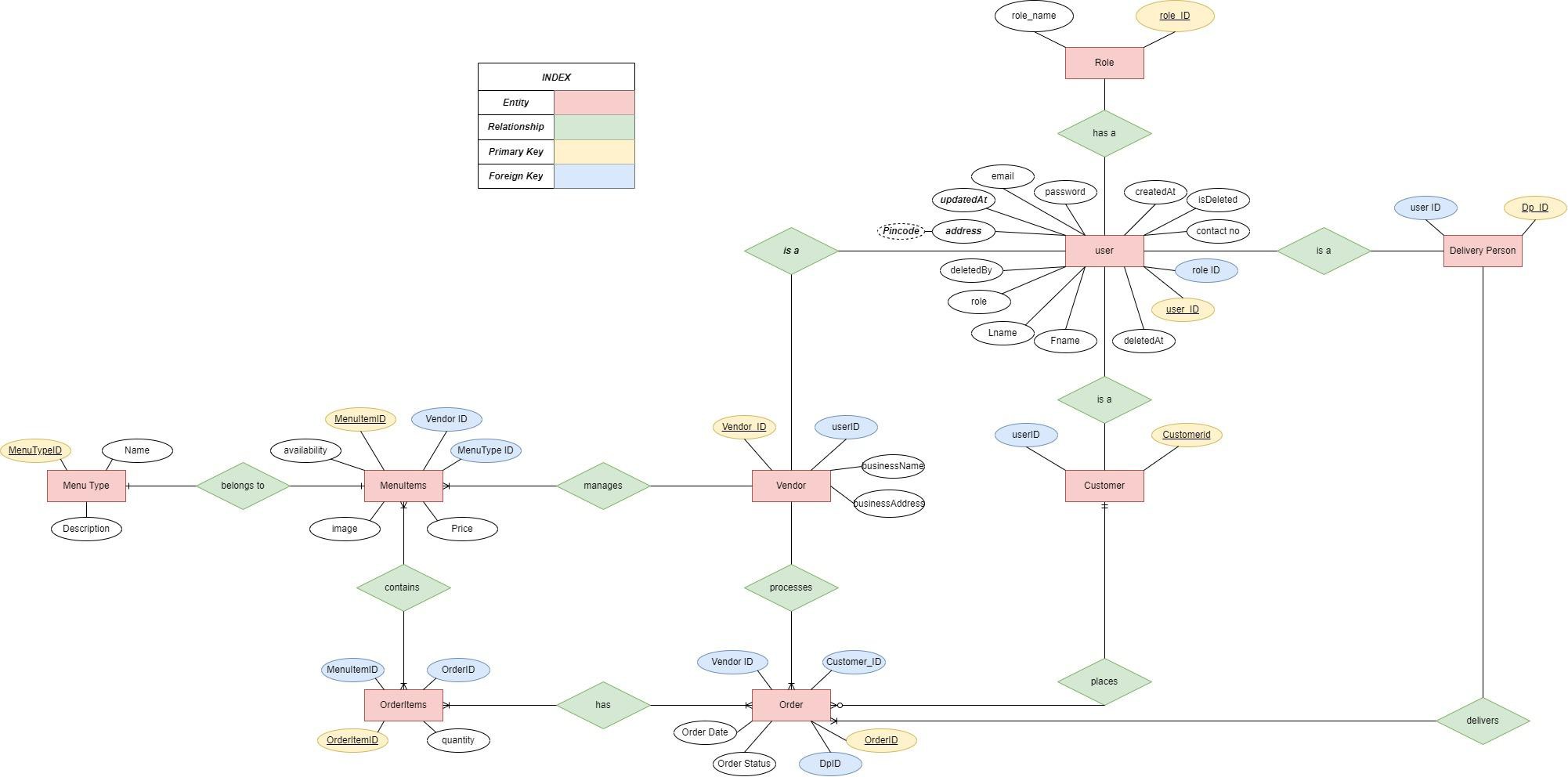
for submission

-----

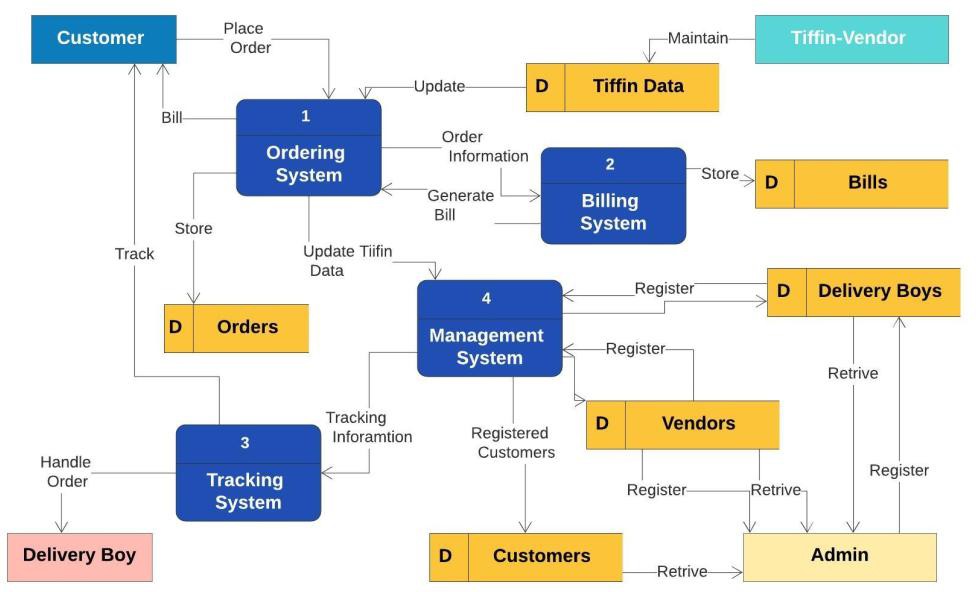
-

**Entity Relationship Diagram**





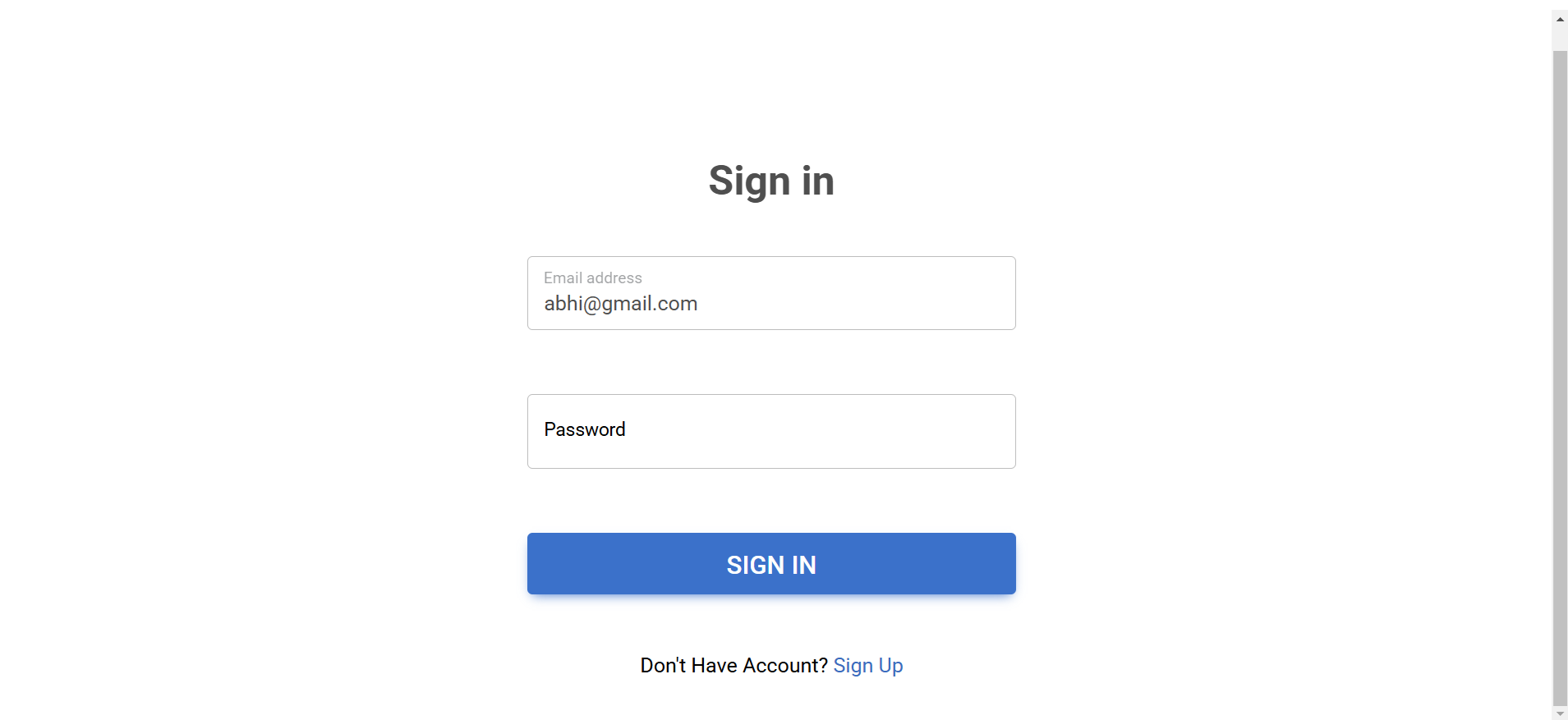
**Data Flow Diagram:**



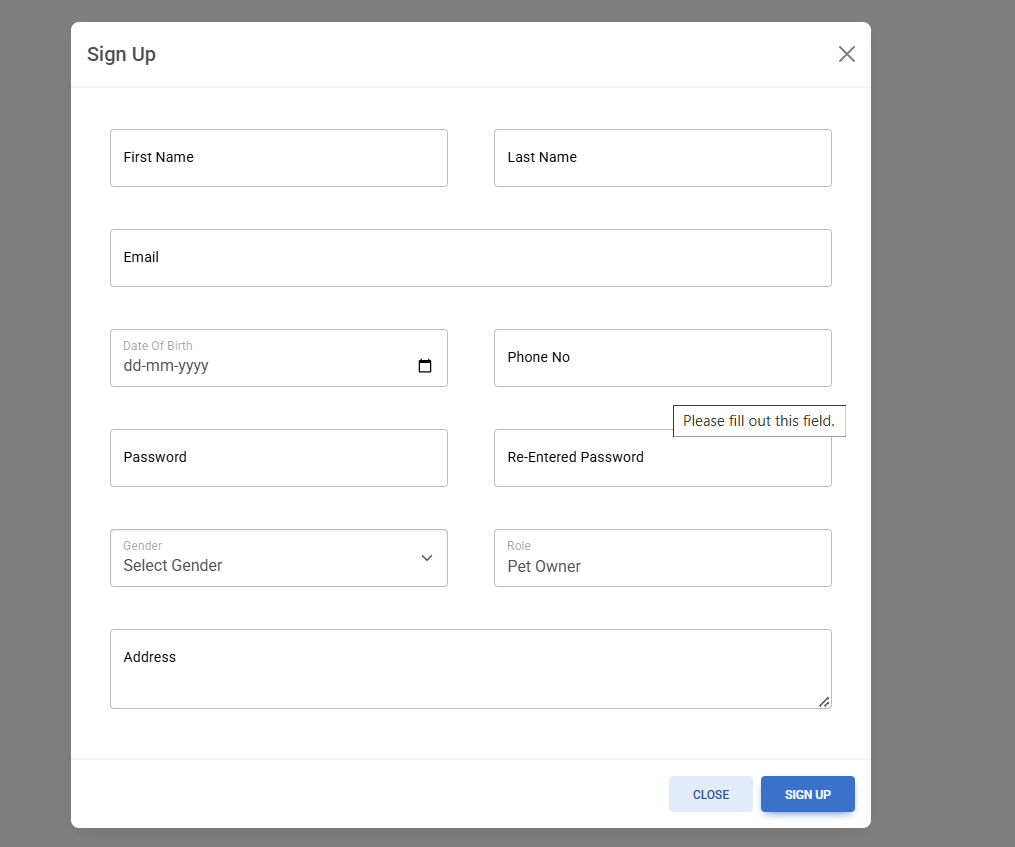
**Class Diagram**

**UI:**

**Signin Page:**



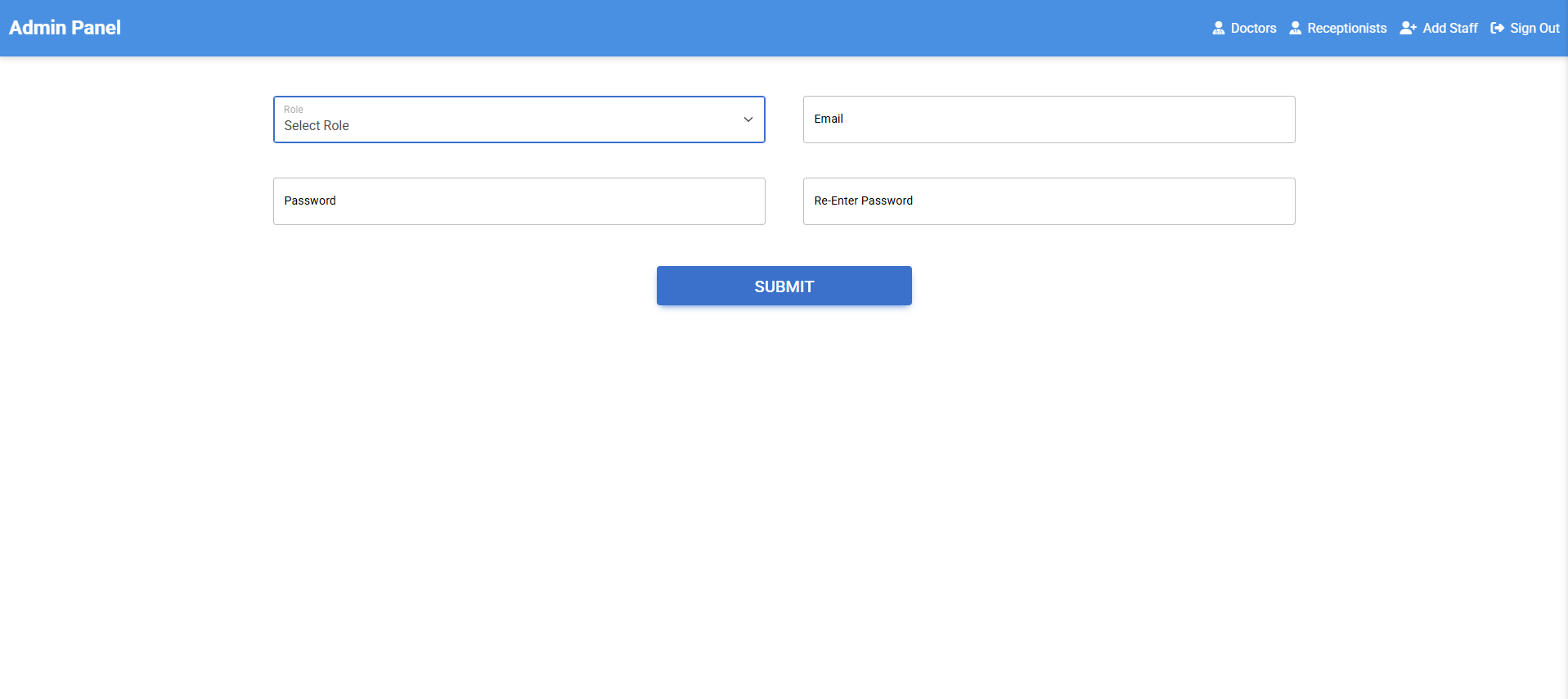
**SignupPage(PetOwner):**

****

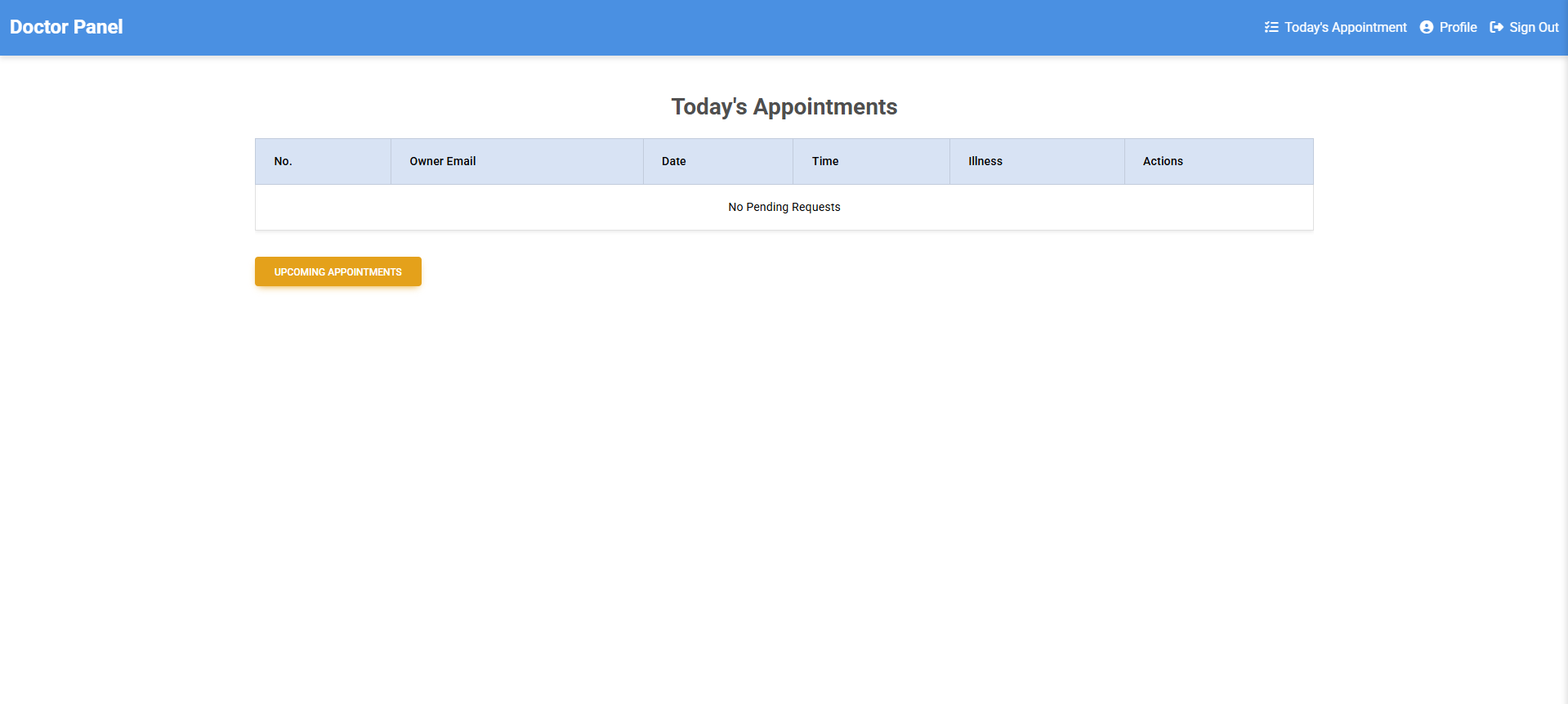
**Admin Page:**



**Add Staff :**

****

**Doctor’s Page:**

****

# REFERENCES

## Spring Boot Documentation

URL: <https://spring.io/projects/spring-boot>

## React.js Documentation

URL: https://reactjs.org/docs/getting-started.html

## Redux Documentation

URL: https://redux.js.org

## Java Programming Language

URL: <https://www.oracle.com/java/>

## MySQL Workbench Documentation

URL: <https://dev.mysql.com/doc/workbench/en/>

## Spring Boot with React and Redux

URL: https://[www.baeldung.com/spring-boot-react-and-redux](http://www.baeldung.com/spring-boot-react-and-redux)

## Java Persistence API (JPA) Documentation

URL: https://[www.eclipse.org/eclipselink/documentation/2.7/](http://www.eclipse.org/eclipselink/documentation/2.7/)

## Swagger Documentation for Spring Boot

URL: <https://springdoc.org/>

## MDN Web Docs

URL: <https://developer.mozilla.org/>

## React Redux Integration Guide

URL: https://react-redux.js.org/