**DEPARTMENT OF INFORMATION SCIENCE ENGINEERING DAYANANDA SAGAR COLLEGE OF ENGINEERING** (An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified) Accredited by National Assessment & Accreditation Council (NAAC) with ‘A’ grade, Accredited by NBA

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078.



**Web Development Lab (22ISL473)**

**Mini Project**

on

**“Hospital Management System”**

Submitted By

**VISHESH KUMAR 1DS22IS184**

**VIDYASAGAR G 1DS22IS182**

**Under the Guidance of**

**Mr. Sureshkumar M & Mrs. Vani KA**

**Assistant Professor**

**Dept. of ISE, DSCE**

# DAYANANDA SAGAR COLLEGE OF ENGINEERING

## Shavige Malleshwara Hills, Kumaraswamy Layout

**Bangalore-560078**

**Department of Information Science and Engineering**

**2023-2024**

# CERTIFICATE

This is to certify that the Mini Project Work entitled **“Hospital Management System”** is a bonafide work carried out by Vishesh Kumar (1DS22IS184), VIDYASAGAR G (1DS22IS182) in partial fulfillment for the 4th semester of Bachelor of Engineering in Information Science & Engineering of the Visvesvaraya Technological University, Belgavi during the year 2023-2024. The Web Development Lab Mini Project(22ISL473) report has been approved as it satisfies the academic requirements prescribed for the Bachelor of Engineering degree.

|  |  |
| --- | --- |
| **Signature of Examiner 1** | **Signature of Examiner 2** |
|  |  |

**CONTENTS**

**Page No**

**CONTENTS**

00

Abstract

01

1. **Introduction**
   1. Problem statement
   2. Proposed Solution
   3. Motivation of Project

03

1. **Requirements**

2.1 Existing and Proposed System

2.2 Software & Hardware Requirements

1. **System Design Modelling**

05

3.1 Entity-Relationship Diagram

3.2 Schema Diagram

1. **Implementation**

10

4.1 Implementation of operations

4.2 Implementation of SQL statements

1. **Snapshots**

12

14

1. **Conclusion**

14

**References**

**ABSTRACT**

The Hospital Management System project aims to enhance the efficiency and effectiveness of hospital operations by leveraging web technologies. Developed using HTML, CSS, Flask, and a database, this system provides a comprehensive solution for managing patient information, appointments, and medical records. The project addresses the need for a streamlined, user-friendly interface for both hospital staff and patients. Key features include patient registration, appointment scheduling, and secure data management. The back-end is implemented using Flask, ensuring a robust application layer that interacts seamlessly with the database, managed via SQLAlchemy ORM. Security measures, such as user authentication and data encryption, are integrated to protect sensitive information. The system also includes role-based access control, ensuring that only authorized personnel can access specific data and functionalities. Through thorough testing and iterative design, the system ensures reliability and usability. This report outlines the development process, from initial design to deployment, highlighting the challenges encountered and the solutions implemented. The Hospital Management System demonstrates a significant improvement in hospital administration efficiency and patient data management, with potential for future enhancements such as telemedicine integration, real-time patient monitoring, and advanced data analytics to support decision-making and improve patient outcomes.

**CHAPTER 1**

**INTRODUCTION**

**1.1 Problem Statement**

The healthcare industry faces challenges in efficiently managing patient information, appointment scheduling, and medical records due to outdated systems and manual processes. There is a need for a modern, web-based Hospital Management System (HMS) that can streamline these administrative tasks, improve data accuracy, and enhance overall operational efficiency. This project aims to develop such a system using HTML, CSS, Flask, and a database, deployed on a XAMPP server, to provide a user-friendly interface for hospital staff and patients alike. The system will automate patient registration, facilitate seamless appointment scheduling, and ensure secure and reliable management of medical records. By addressing these challenges, the HMS seeks to optimize hospital operations and improve the quality of patient care.

**1.2 Proposed Solution**

The proposed solution is a comprehensive hospital management system developed using HTML, CSS, and Flask, integrated with a database on an XAMPP server. This system aims to streamline hospital operations by providing efficient patient management, appointment scheduling, staff management, billing, and reporting functionalities. It enhances accessibility, accuracy, and security of data, ensuring that authorized personnel can easily manage and access patient and hospital information. Additionally, the system is designed to be scalable, allowing for future enhancements such as mobile app integration and telemedicine features to further improve hospital services.

**1.3 Motivation of Project**

The system aims to streamline hospital operations by automating patient registration, appointment scheduling, and medical record management. Key objectives include enhancing administrative efficiency through reduced manual processes, ensuring data security with robust

measures like user authentication and encryption, and providing a user-friendly interface for hospital staff and patients. Additionally, the HMS is designed for scalability to accommodate future expansions such as electronic medical records (EMR) integration and telemedicine functionalities, ultimately aiming to improve overall patient care and comply with healthcare regulations.

**CHAPTER 2**

**REQUIREMENT**

**2.1 Existing and Proposed System**

The traditional system of hospital management is inefficient and prone to errors. In the conventional setup, hospital staff must manually manage patient records, appointments, billing, and other administrative tasks. This process often involves maintaining paper records, which can be lost or damaged, leading to significant issues in patient care and hospital operations.

Moreover, the manual system is time-consuming and requires a substantial number of staff to perform simple tasks, such as scheduling appointments, updating patient records, and processing bills. This not only increases operational costs but also results in delays and errors in service delivery. Patients often experience long wait times, and there is a higher likelihood of administrative mistakes.

.To address these inefficiencies, a web-based computerized hospital management system is essential. The proposed system offers a comprehensive solution that integrates various hospital functions into a single platform. It streamlines patient record management, appointment scheduling, billing, and other administrative tasks, making the process more efficient and accurate.

With the new system, there is a significant reduction in the number of staff required, as many tasks are automated. The software's high performance and speed enhance overall operational efficiency, saving time and reducing the chances of errors. Additionally, the system ensures data security and minimizes the risk of corruption by staff. This modern approach to hospital management benefits both patients and the hospital, providing a smoother, more reliable service experience.

**2.2 System Requirements**

**Hardware Requirements:**

**Hardware Specification:** Processor Intel Pentium V or higher

**System Bus:** 64 bits

**RAM:** Minimum 4 GB (8 GB recommended)

**Storage:** Minimum 500 GB HDD (SSD recommended for better performance)

**Display:** 14-inch monitor with a resolution of 1080x1920 or higher

**Network:** Ethernet card or Wi-Fi adapter for network connectivity

**Peripheral Devices:** Keyboard, Mouse, and Printer (optional)

**Software Requirements:**

**Operating System:** Windows 10 or later, or any Linux distribution

**Web Server:** Apache (integrated with XAMPP)

**Software:** Microsoft SQL Server, VS Code, XAMPP server

**Front End:** HTML, CSS, PHP (integration)

**Back End:** MySQL

**CHAPTER 3**

**SYSTEM DESIGN AND MODELLING**

**3.1 Entity-Relationship Diagram**

**Entities:**

1. **Doctor**
   * Attributes: DoctorID (Primary Key), Email, DoctorName, Department
2. **Patient**
   * Attributes: PatientID (Primary Key), Email, Name, Gender, Slot, Disease, AppointmentTime, AppointmentDate, Department, PhoneNumber
3. **User**
   * Attributes: UserID (Primary Key), Username, UserType, Email, Password
4. **Trigger Log (trigr)**
   * Attributes: TriggerID (Primary Key), PatientID, Email, Name, Action, Timestamp
5. **Test (for testing purposes)**
   * Attributes: TestID (Primary Key), Name, Email

**Relationships:**

1. **Doctor - Patient**
   * Relationship: A doctor can have multiple patients assigned to them.
   * Cardinality: 1-to-Many (One doctor can have many patients)
2. **User - Patient**
   * Relationship: A user can be a patient.
   * Cardinality: 1-to-1 (One user can be one patient)
3. **User - Doctor**
   * Relationship: A user can be a doctor.
   * Cardinality: 1-to-1 (One user can be one doctor)

**3.2 Schema Diagram**

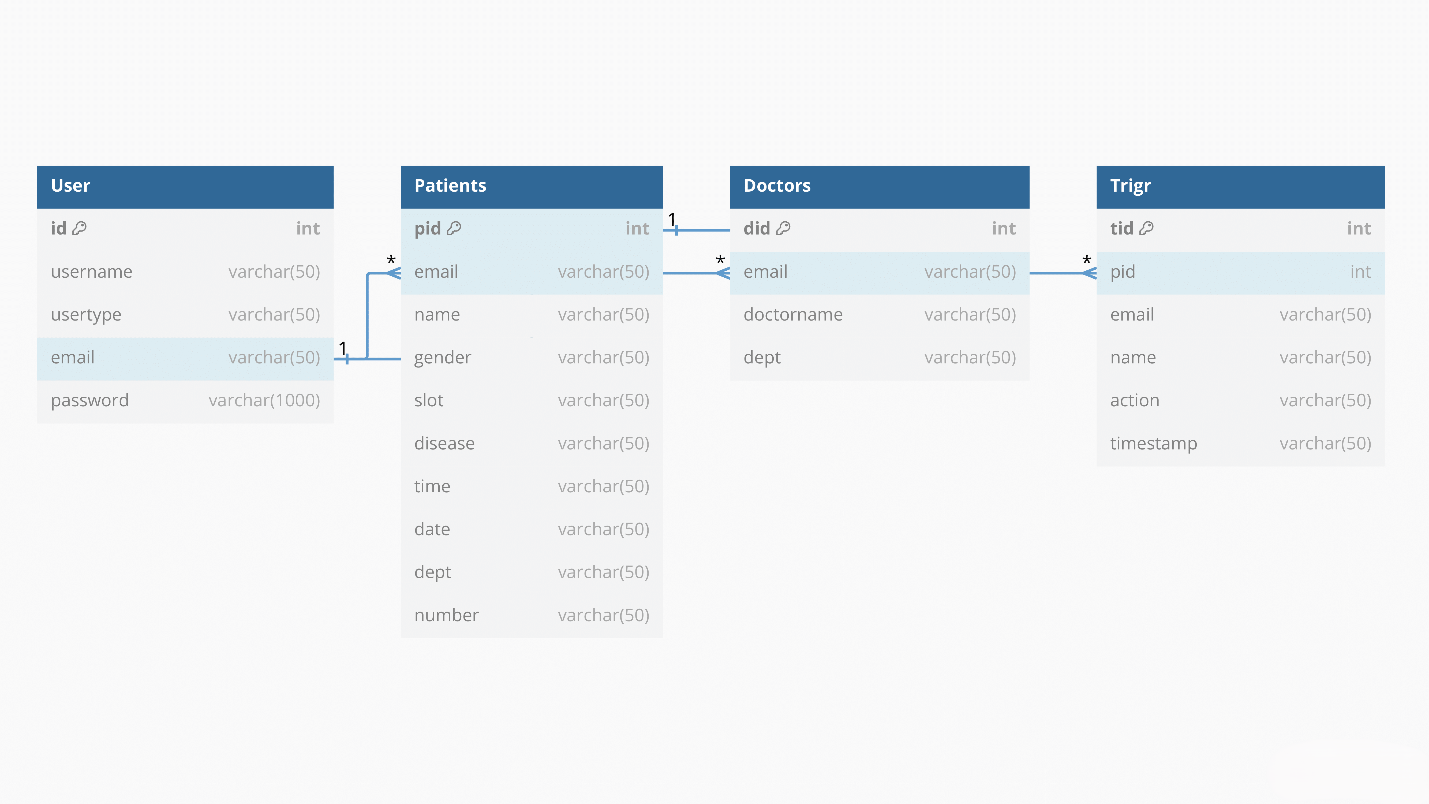


Fig 3.1: Schema Diagram

**CHAPTER 4**

**IMPLEMENTATION & APPLICATION CODE**

**4.1 Implementation of Operations**

**Adding Patient:** The admin can add new patients to the system, including their personal details, appointment slots, and the department they need to visit.

**Searching Appointment Details:** The admin can search for appointment details using patient ID, date, or department to view specific appointment information.

**Adding Doctor:** The admin can add new doctors to the system, specifying their department, contact information, and availability.

**Updating Patient Information:** Patients can update their personal details, such as contact information or medical history, using their patient ID.

**Deleting Patient Records:** The admin can delete patient records based on patient ID, ensuring outdated or irrelevant information is removed from the system.

**Managing User Accounts:** The admin can manage user accounts, including adding new users, updating existing user details, and deleting inactive accounts.

**4.2 Implementation of SQL Statements**

**Insert Statement:** The INSERT INTO statement adds new records to a table using the syntax INSERT INTO table\_name VALUES (value1, value2, value3, ...).

For example: INSERT INTO patients (email, name, gender, slot, disease, time, date, dept, number) VALUES ('john.doe@example.com', 'John Doe', 'Male', 'morning', 'fever', '09:00:00', '2024-07-11', 'Cardiology', '1234567890');

**Update Statement:** The UPDATE statement modifies existing records in a table. It uses the syntax UPDATE table\_name SET column\_name = value WHERE condition.

For example: UPDATE patients SET email = 'john.newemail@example.com' WHERE pid = 1;

**Create Statement:** The CREATE TABLE statement creates a new table with specified columns and constraints. Syntax: CREATE TABLE table\_name (column1 datatype, column2 datatype, ... PRIMARY KEY (one or more columns)).

For example:

CREATE TABLE doctors (

did INT(11) NOT NULL AUTO\_INCREMENT,

email VARCHAR(50) NOT NULL,

doctorname VARCHAR(50) NOT NULL,

dept VARCHAR(100) NOT NULL,

PRIMARY KEY (did) );

**SNAPSHOTS**

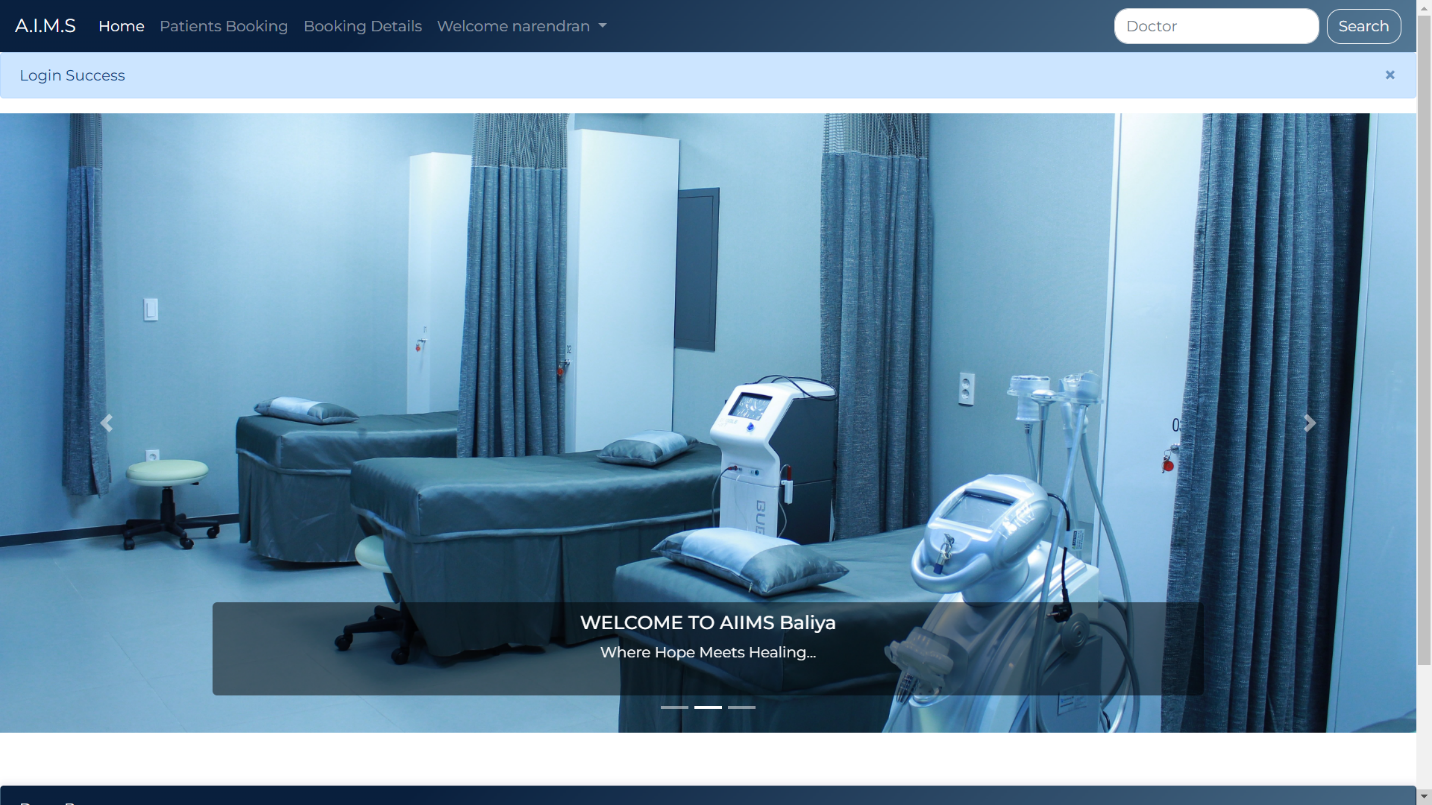
****

Fig 5.1: Home Page

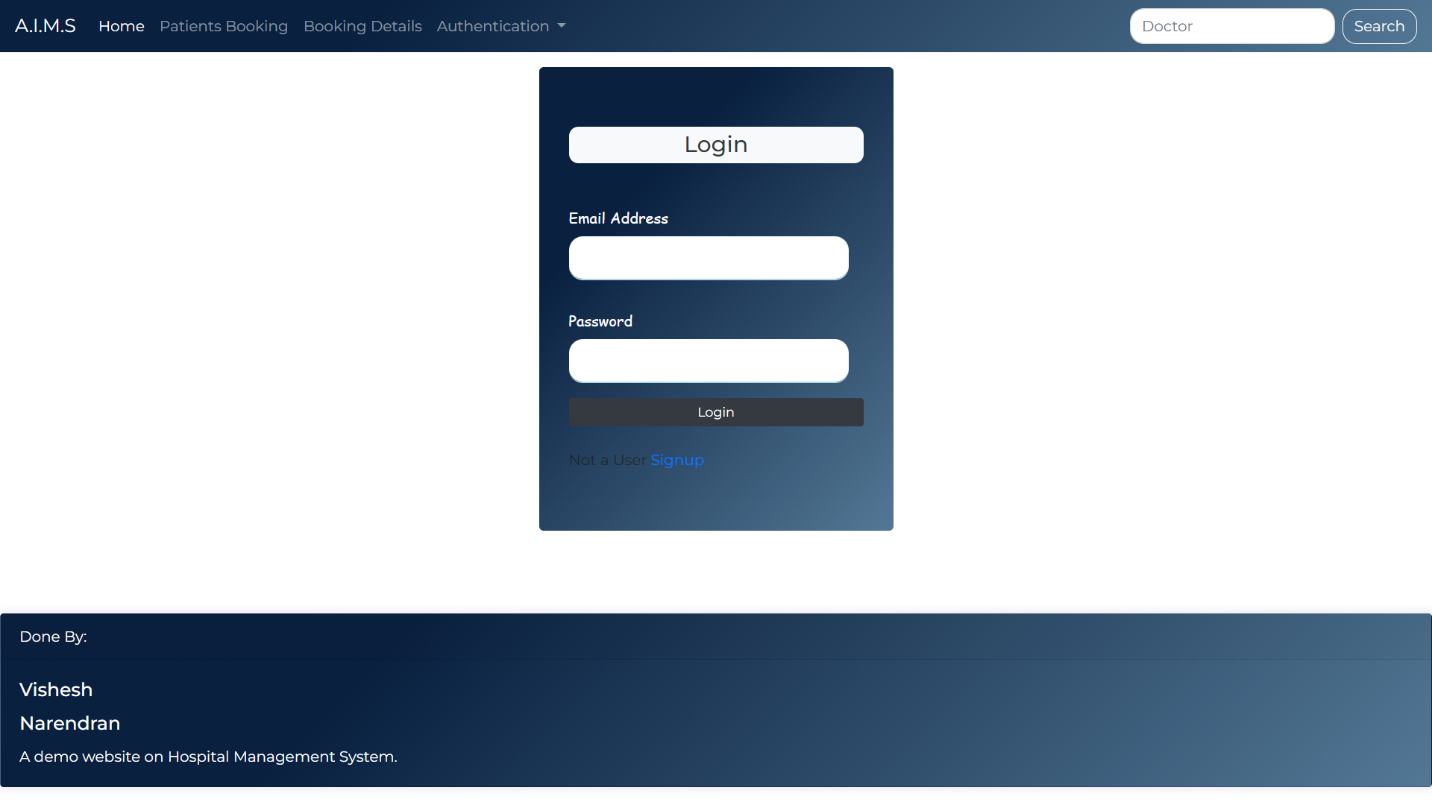
****

Fig 5.2: Login Page

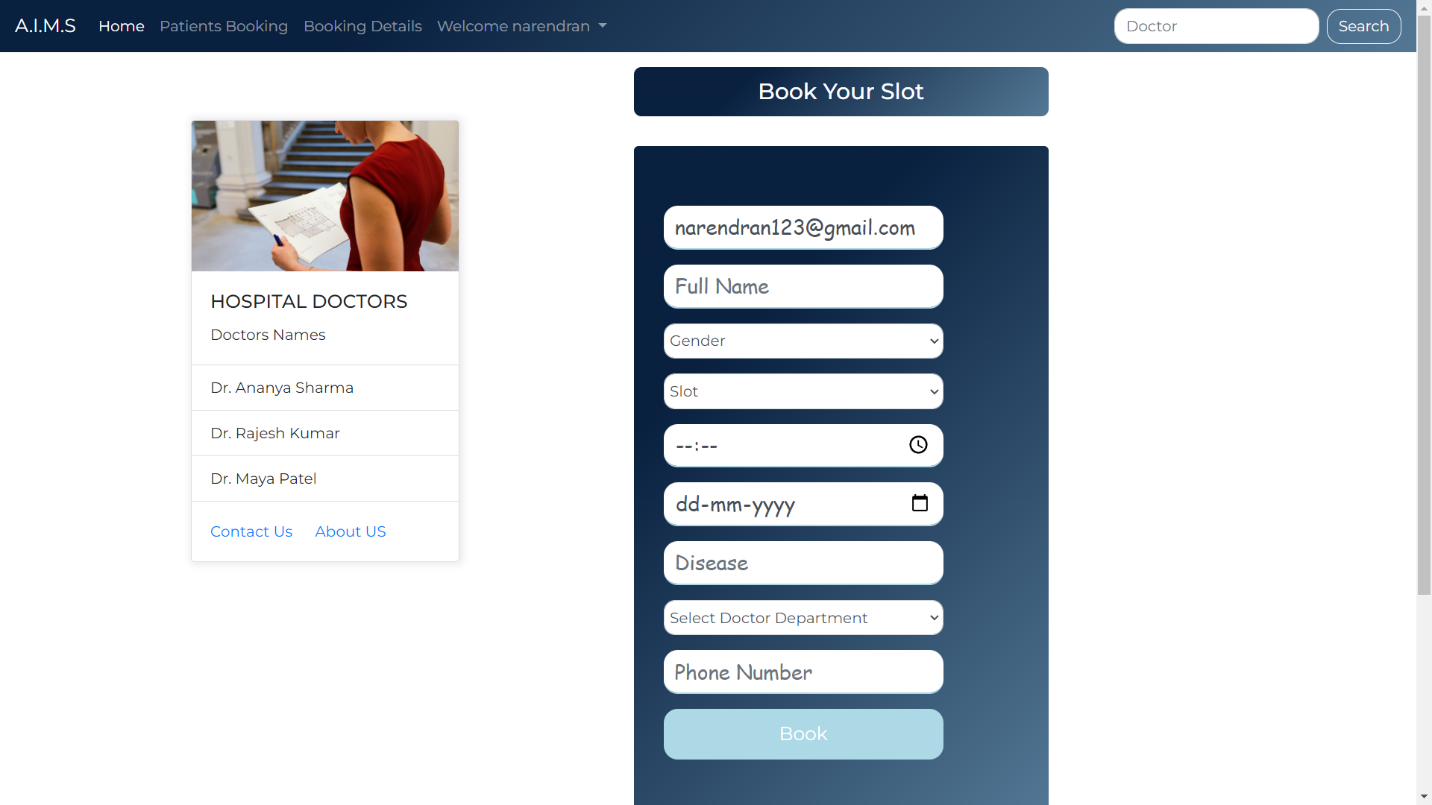
****

Fig 5.3: Booking Page

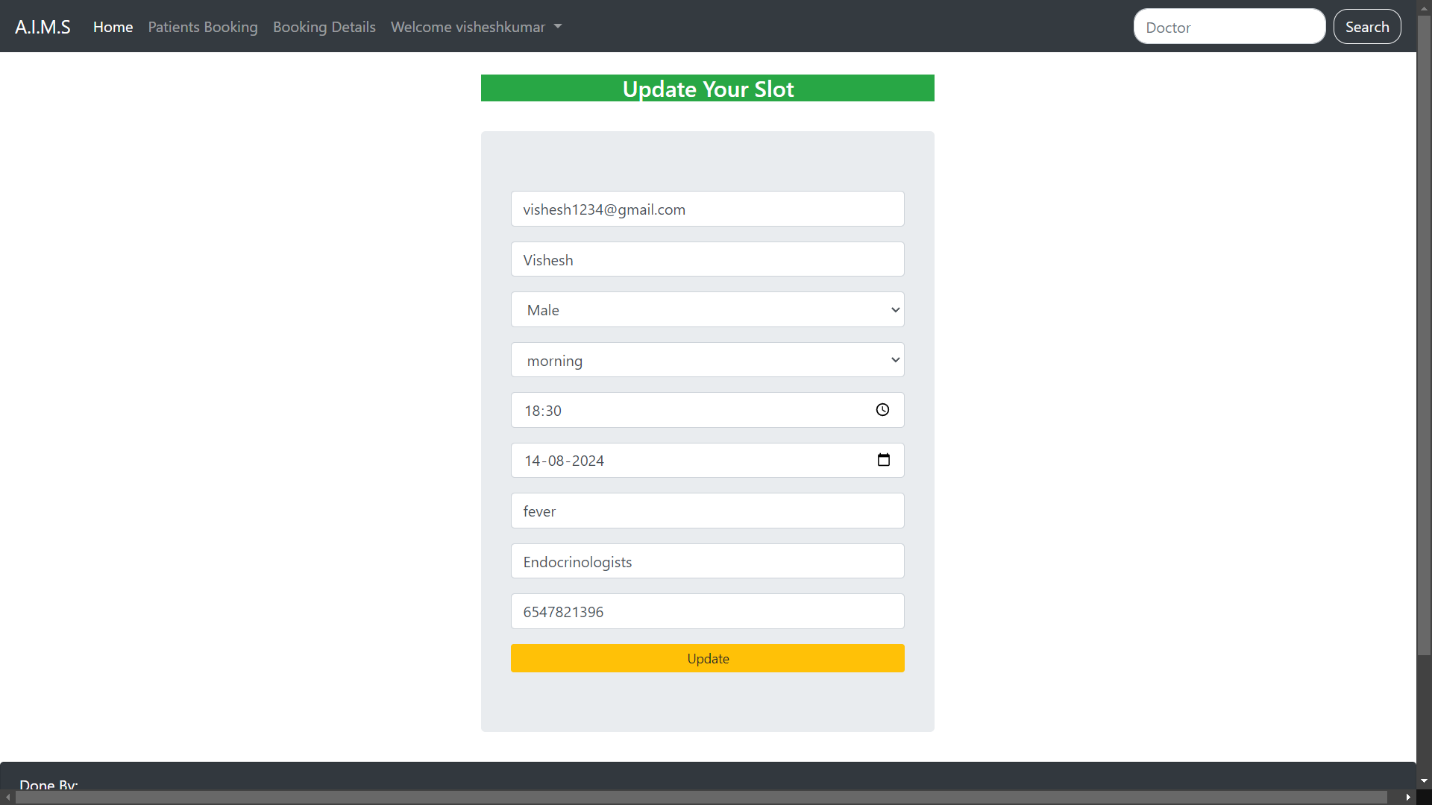


Fig 5.4: Update page

**CONCLUSION**

In conclusion, the development of the hospital management system using HTML, CSS, Flask, and a MySQL database integrated on an XAMPP server provides a robust and efficient solution for managing hospital operations. This system enhances the accuracy and accessibility of patient data, streamlines appointment scheduling and staff management, and improves billing and reporting processes. By leveraging modern web technologies and scalable architecture, the system ensures security and ease of use, making it a valuable tool for hospitals aiming to improve their administrative efficiency and patient care services. Future enhancements like mobile app integration and telemedicine capabilities can further extend its functionality, ensuring it meets the evolving needs of the healthcare industry.

**REFERENCES**

**References**

**Flask Documentation**: Retrieved from flask.pocoo.org/docs

**SQLAlchemy Documentation:** Retrieved from docs.sqlalchemy.org

**Healthcare Management Systems:** Academic research paper discussing effective strategies in hospital management systems.