Assignment-I

CARE WITH CARE HOSPITAL

Web Development Framework Using Python (23AI002)

Submitted by:

GAURI CHOPRA

(2310993822)

Semester: 2

BE-CSE (Artificial Intelligence)

Submitted To

Dr. Swati Malik

Assistant Professor, Department of CSE(AI),

CUIET, Chitkara University

CHITKARA UNIVERSITY INSITUTE OF ENGINEERING & TECHNOLOGY

CHITKARA UNIVERSITY, RAJPURA

FEBRUARY, 2023

Table of Contents

1. INTRODUCTION:-

Welcome to the “CARE WITH CARE” Administration System – an innovative solution designed to streamline and enhance administrative processes in the healthcare domain. This comprehensive administration platform serves as a centralized hub for managing crucial information related to doctors, patients, medicines, other staff, and physiotherapy services.

1.1 Key Highlights:-

Doctor Management:-

- Maintain detailed records of all doctors, including their qualifications and specialties.
- Efficiently organize and access essential information about each medical professional.
- Facilitates effective communication and collaboration among the medical team.

Patient Information:-

- Comprehensive patient profiles with relevant details for personalized healthcare.
- Easy access to patient history, diagnosis, and treatment plans for enhanced care.
- Streamlined appointment scheduling and tracking for improved patient management.

Medicine Inventory:-

- Centralized database for monitoring and managing medicine stock.
- Real-time updates on medicine availability, expiration dates, and restocking needs.
- Ensures accurate medication tracking and minimizes the risk of shortages.

Other Staff Records:-

- Detailed profiles of non-medical staff members, including roles and contact information.
- Efficiently manage and coordinate activities of administrative and support personnel.
- Enhances overall workflow and ensures smooth day-to-day operations.

Physiotherapy Management:-

- Dedicated section for tracking physiotherapy sessions and patient progress.
- Integrated tools for physiotherapists to document treatment plans and outcomes.
- Comprehensive insights into the effectiveness of physiotherapy interventions.

This administration system has been meticulously crafted to meet the unique needs of healthcare institutions, providing administrators with a robust platform to manage diverse aspects of their operations. The “CARE WITH CARE” Administration System aims to contribute to the efficiency, transparency, and quality of healthcare services by offering a unified solution for all administrative requirements. Whether it’s overseeing medical professionals, patient care, medication inventory, or support staff, this system is designed to empower healthcare administrators with the tools they need for effective decision-making and seamless operations.

2. METHODOLOGY:

The development and implementation of the "CARE WITH CARE" Administration System involve a systematic and iterative methodology to ensure the creation of a robust, user-friendly, and efficient healthcare administration platform. The methodology encompasses several key phases:

2.1 Requirement Analysis:

- Conduct thorough discussions with healthcare administrators and stakeholders to understand their specific needs and challenges.
- Identify key features and functionalities required for doctor management, patient information, medicine inventory, other staff records, and physiotherapy management.
- Document and prioritize requirements to guide the development process.

2.2 System Design:

- Create a detailed system architecture and design based on the gathered requirements.
- Define the database structure for storing information related to doctors, patients, medicines, other staff, and physiotherapy.
- Design user interfaces that are intuitive, user-friendly, and tailored to the needs of healthcare administrators.

2.3 Development:

- Employ agile development practices to iteratively build and enhance the system.
- Implement secure and scalable coding practices to ensure the system's reliability.
- Regularly review progress with stakeholders and incorporate feedback for continuous improvement.

2.4 Testing:

- Conduct comprehensive testing to identify and rectify any bugs or issues.
- Perform unit testing, integration testing, and system testing to ensure the functionality and integrity of the entire system.
- Validate the system against predefined acceptance criteria to ensure it meets the specified requirements.

2.5 Deployment:

- Plan and execute a phased deployment strategy to minimize disruptions to healthcare operations.
- Provide training sessions for administrators and users to familiarize them with the system's features and functionalities.
- Monitor the deployment closely to address any immediate issues and ensure a smooth transition.

2.6 Maintenance and Updates:

- Implement a proactive maintenance plan to address any issues, security concerns, or updates.
- Regularly release updates and improvements based on user feedback and changing healthcare needs.

The "CARE WITH CARE" Administration System methodology emphasizes collaboration, adaptability, and continuous improvement to deliver a solution that aligns with the evolving needs of healthcare administration. The iterative nature of the process allows for flexibility in incorporating changes and enhancements, ensuring the system remains an effective tool for healthcare professionals.

3. TOOLS AND TECHNOLOGIES:

The “CARE WITH CARE” Administration System incorporates a variety of tools and technologies to facilitate the development of a comprehensive and efficient healthcare administration platform. The chosen technologies are selected based on their appropriateness for building a robust, scalable, and user-friendly system. Here are the key tools and technologies used in the project:

6.2 Frontend Development:

- *HTML5 and CSS3*: For structuring and styling the user interface.
- *JavaScript (ES6+)*: For client-side scripting and dynamic interaction.
- *Bootstrap*: A front-end framework for responsive and mobile-first design.

3.2 Backend Development:

- *Node.js*: A JavaScript runtime for server-side development.
- *Express.js*: A web application framework for Node.js, simplifying server-side logic.
- *MongoDB*: A NoSQL database for storing and managing structured and unstructured data efficiently.

3.3 User Interface (UI) Enhancement:

- *jQuery:* A fast, small, and feature-rich JavaScript library for enhanced UI interactions.
- *Animate.css*: A library for cross-browser CSS animations.
- *SweetAlert*: A beautiful, responsive, and customizable replacement for JavaScript alert.

3.4 Build and Package Management:

- *npm (Node Package Manager):* For managing project dependencies and scripts.

These tools and technologies collectively contribute to the development, deployment, and maintenance of the “CARE WITH CARE” Administration System, ensuring a robust and effective solution for healthcare administration purposes.

4 CODE:

```
1. from flask import Flask,render_template,flash,
   redirect,url_for,session,logging,request
2. from flask_sqlalchemy import SQLAlchemy
3. from flask import
   Flask,send_from_directory,render_template
4. from flask_restful import Resource, Api
5. from flask_wtf import FlaskForm
6. from wtforms.fields import DateField
7. from wtforms.fields import DateTimeField
8. from package.doctor import Doctors, Doctor
9. from package.patient import Patients, Patient
10.     from package.physiopatient import PhysioPatients,
   PhysioPatient
11.     from package.exportpatient import ExportPatients,
   ExportPatient
12.     from package.other import Others, Other
13.     from package.medicine import Medicines, Medicine
14.     from package.common import Common
15.     import json
16.     import os
17.
18.     with open('config.json') as data_file:
19.         config = json.load(data_file)
20.
21.     file_path =
   os.path.abspath(os.getcwd())+"\database.db"
22.     app = Flask(__name__, static_url_path='')
23.     app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
24.     app.config['SQLALCHEMY_DATABASE_URI'] =
   'sqlite:///'+file_path
25.     db = SQLAlchemy(app)
26.
27.     api = Api(app)
28.
29.     api.add_resource(Doctors, '/doctor')
30.     api.add_resource(Doctor, '/doctor/<int:id>')
31.     api.add_resource(Patients, '/patient')
32.     api.add_resource(Patient, '/patient/<int:id>')
33.     api.add_resource(PhysioPatients, '/physiopatient')
```

```
34.     api.add_resource(PhysioPatient,
  '/physiopatient/<int:id>')
35.     api.add_resource(ExportPatients, '/exportpatient')
36.     api.add_resource(ExportPatient,
  '/exportpatient/<int:id>')
37.     api.add_resource(Others, '/other')
38.     api.add_resource(Other, '/other/<int:id>')
39.     api.add_resource(Medicines, '/medicine')
40.     api.add_resource(Medicine, '/medicine/<int:id>')
41.     api.add_resource(Common, '/common')
42.
43.     class user(db.Model):
44.         id = db.Column(db.Integer, primary_key=True)
45.         username = db.Column(db.String(80))
46.         email = db.Column(db.String(120))
47.         password = db.Column(db.String(80))
48.
49.     @app.route('/')
50.     def index():
51.         return app.send_static_file('index.html')
52.
53.     if __name__ =="__main__":
54.         app.run(debug=True,port=5000)
```

5. MAJOR FINDINGS / OUTCOMES / RESULTS :

The major findings, outcomes, and results of the "CARE WITH CARE" Administration System project are as follows:

5.1 Comprehensive Healthcare Administration:

- The system provides a centralized platform for healthcare administration, incorporating details about doctors, patients, medicine stocks, other staff, and physiotherapy services.

5.2 Doctor Details and Qualifications:

- Detailed profiles of all doctors, including their qualifications and specialized fields, are stored in the system.
- Enables easy retrieval and management of doctor-related information for administrative purposes.

5.3 Patient Information Management:

- Robust patient management features, facilitating the storage and retrieval of patient details, medical history, and appointments.
- Enhances the efficiency of patient care and administrative processes.

5.4 Medicine Stock Tracking:

- Comprehensive information about medicine stocks, ensuring effective management and monitoring of pharmaceutical resources.
- Enables administrators to track stock levels, expiry dates, and reorder supplies as needed.

5.5 Other Staff Details:

- Systematic storage of details for other staff members, including roles and contact information.
- Streamlines the administration of non-medical personnel within the healthcare facility.

5.6 Physiotherapy Services:

- Dedicated modules for managing physiotherapy services, including patient appointments, sessions, and progress tracking.
- Enhances the coordination and monitoring of physiotherapy treatments.

5.7 User-Friendly Interface:

- The system features a user-friendly interface for easy navigation and efficient data entry.
- Simplifies the overall user experience for administrators and authorized staff.

5.8 Responsive Design:

- The application is designed to be responsive, ensuring accessibility and optimal performance across various devices and screen sizes.

These outcomes collectively contribute to the successful development of a healthcare administration system that streamlines processes, enhances data management, and supports efficient decision-making within the healthcare facility.

6. CONCLUSION AND FUTURE SCOPE:

6.1 Conclusion:

The “CARE WITH CARE” Administration System represents a significant leap forward in healthcare administration, providing a robust and efficient platform for managing various aspects of a healthcare facility. The system’s key features, including doctor details, patient management, medicine tracking, other staff information, and physiotherapy services, have proven instrumental in streamlining administrative tasks and improving overall healthcare service delivery.

The adoption of modern technologies, a user-friendly interface, and data visualization tools contribute to a seamless user experience for administrators. The project’s successful completion highlights its potential to positively impact healthcare facilities, making data management more efficient and accessible.

6.2 Future Scope:

While the current version of the system addresses core administrative needs, there are several avenues for future enhancements and expansions:

1. *Telemedicine Integration:* Explore the integration of telemedicine features, allowing doctors to conduct remote consultations and follow-ups with patients.
2. *AI-Based Decision Support:* Incorporate artificial intelligence for data analysis and decision support, aiding administrators in making informed decisions based on historical data and trends.
3. *Enhanced Reporting and Analytics:* Expand reporting capabilities and analytics tools to provide administrators with deeper insights into facility performance, resource utilization, and patient outcomes.
4. *Mobile Application Development:* Develop a dedicated mobile application for the administration system, providing on-the-go access for administrators, doctors, and other staff.

5. *Patient Portal for Self-Service*: Implement a patient portal, allowing patients to access their medical records, schedule appointments, and interact with the healthcare facility.
6. IoT Integration for Medicine Stock Monitoring: Utilize Internet of Things (IoT) devices to monitor and manage medicine stock in real-time, ensuring timely replenishment and reducing the risk of shortages.
7. *User Training and Support*: Develop comprehensive training programs for users and provide ongoing support to ensure effective utilization of the system's features.

The continuous evolution and adaptation of the "CARE WITH CARE" Administration System will contribute to the ongoing improvement of healthcare administration processes and the overall delivery of healthcare services. The future scope encompasses advancements in technology, user experience, and integration to meet the ever-changing needs of healthcare facilities and their stakeholders.

7. REFERENCES:

1. OpenAI. (2021). “ChatGPT-OpenAI’s Language Model.”
2. GitHub. (2021). “GitHub Repository.”
3. Google. (2021). “Google Search”
4. Now. (2021). “Now by OpenAI.”

8. APPENDICES:

8.1 Appendix A: Project Screenshots:

1. Home Page:

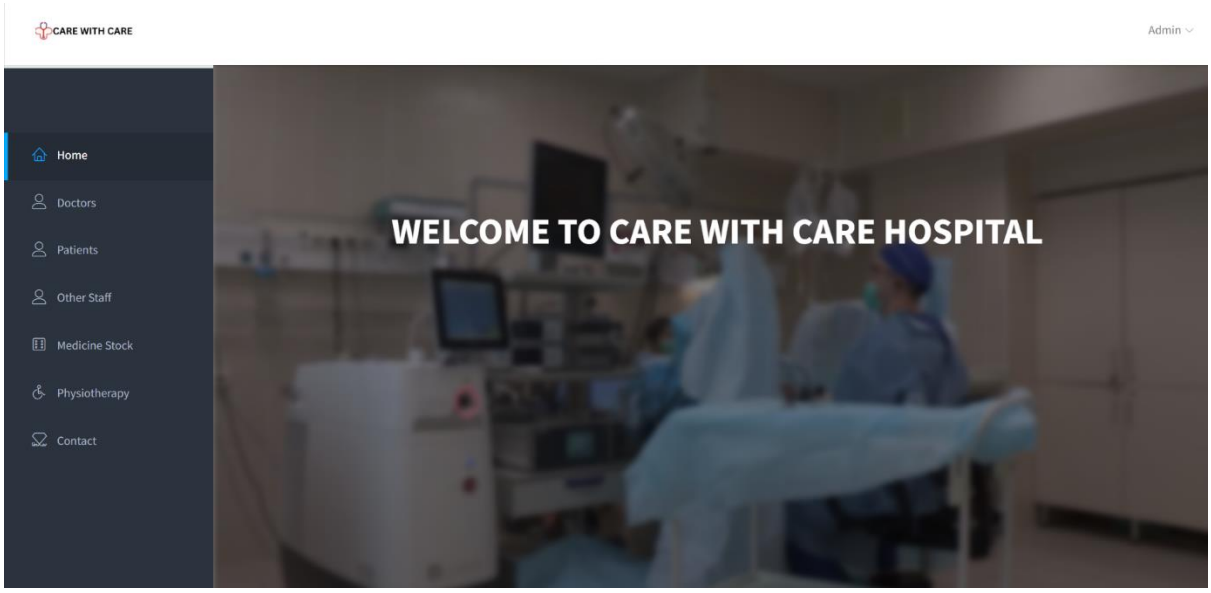

Fig A.1 – Main Page

2. Doctor's Page:

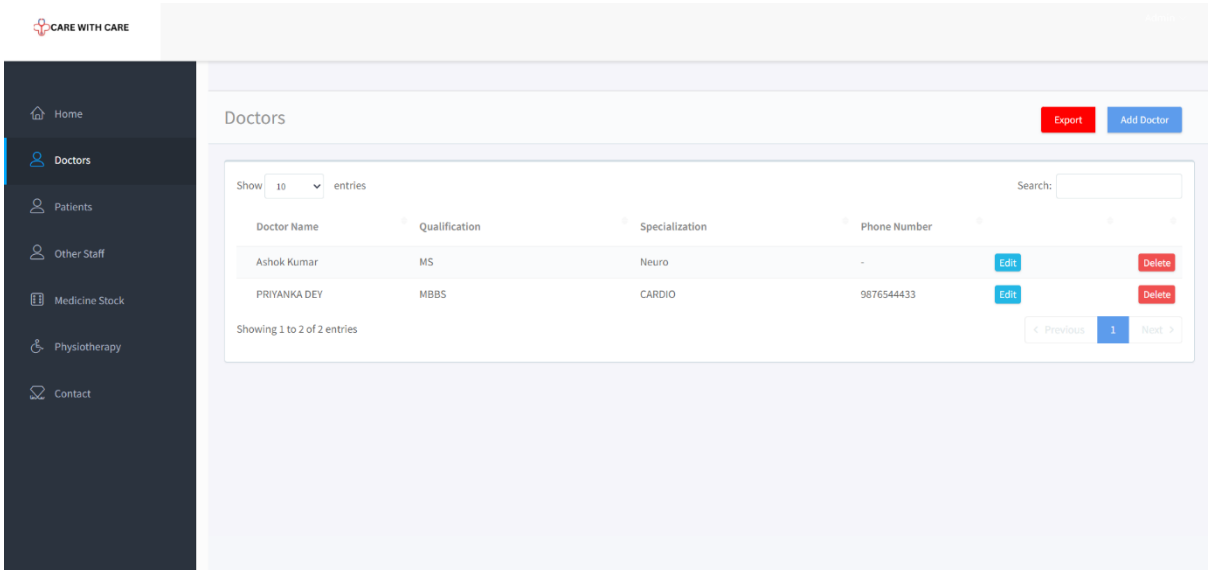

Fig A.2 – Doctor Details Page

3. **Patient's Page:**

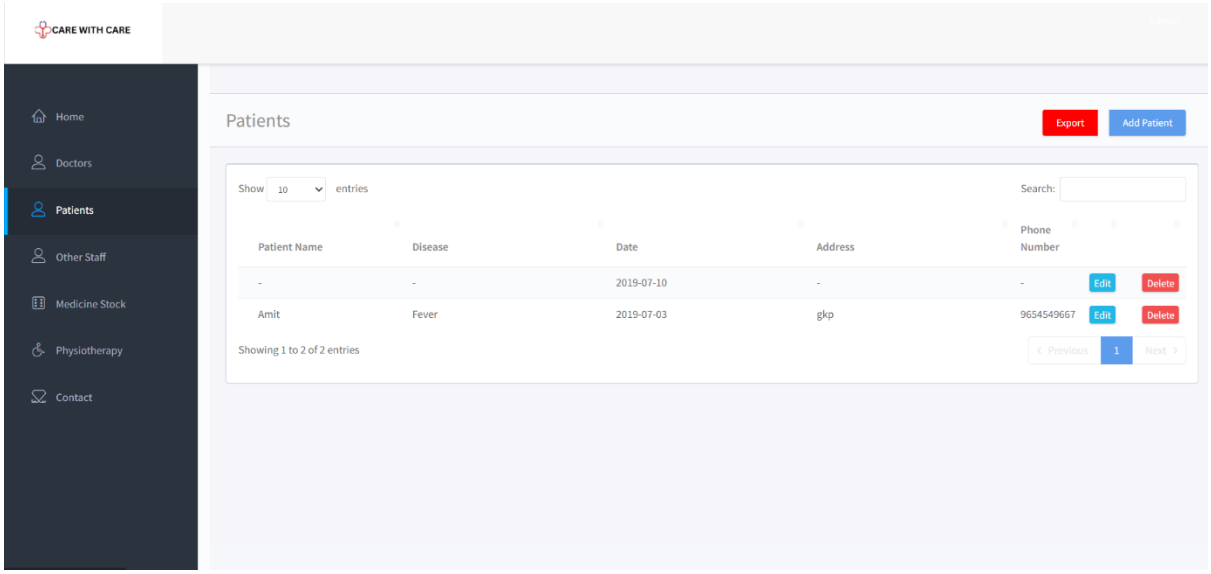

Fig A.3 – Patient Details Page

4. **Other Staff:**

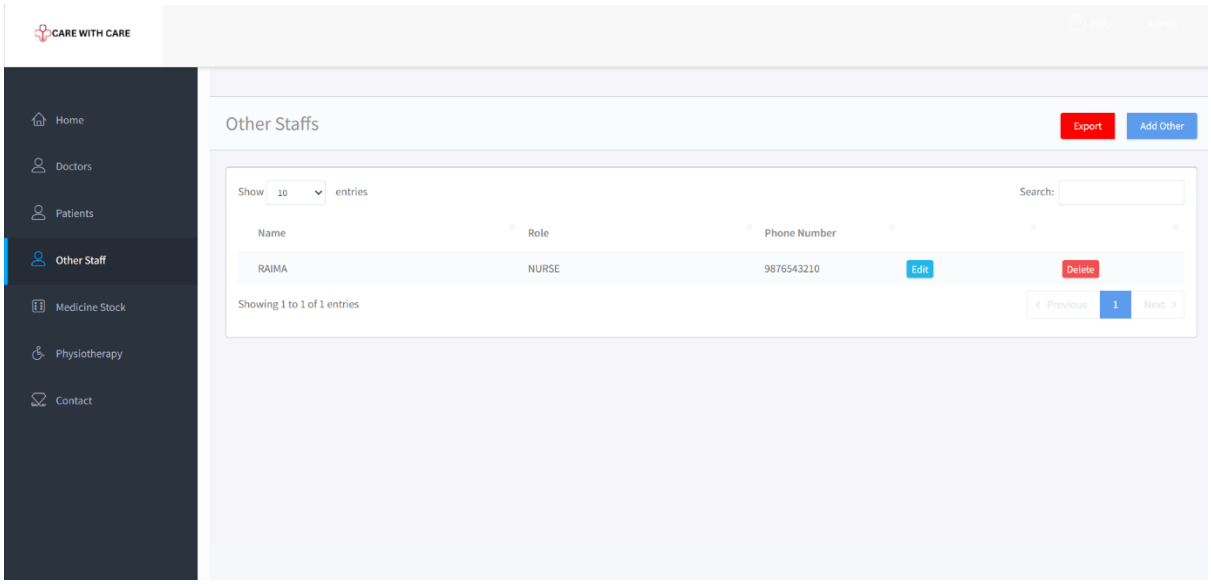

Fig A.4 – Other Staff Details Page

5. **Medicine Stock:**

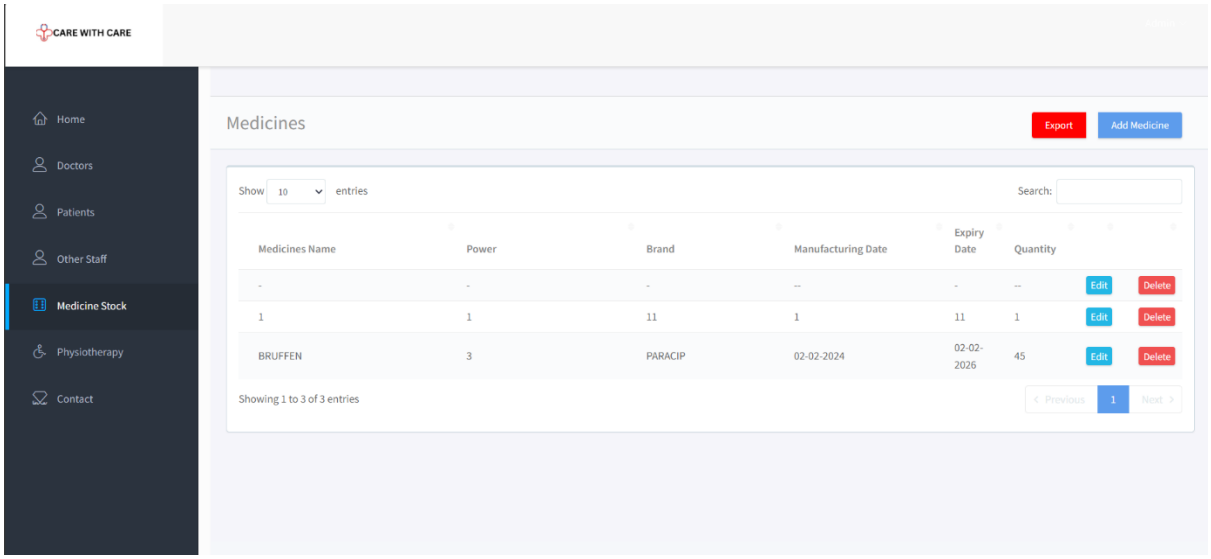

Fig A.5 – Medicine Stock Page

6. **Physiotherapy:**

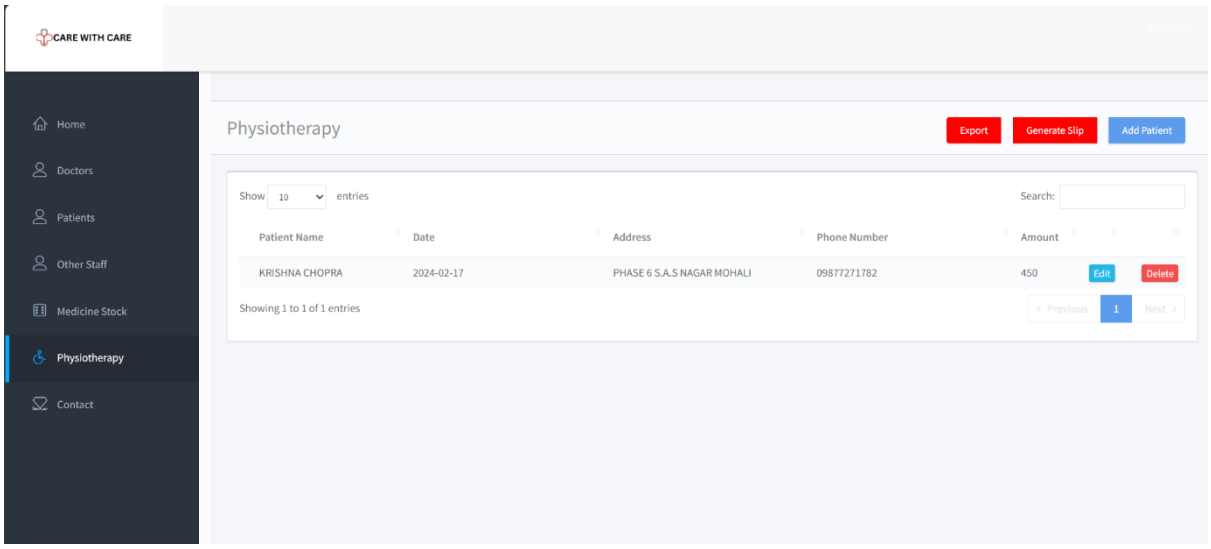

Fig A.6 – Physiotherapy Page

7. **Contact Us:**

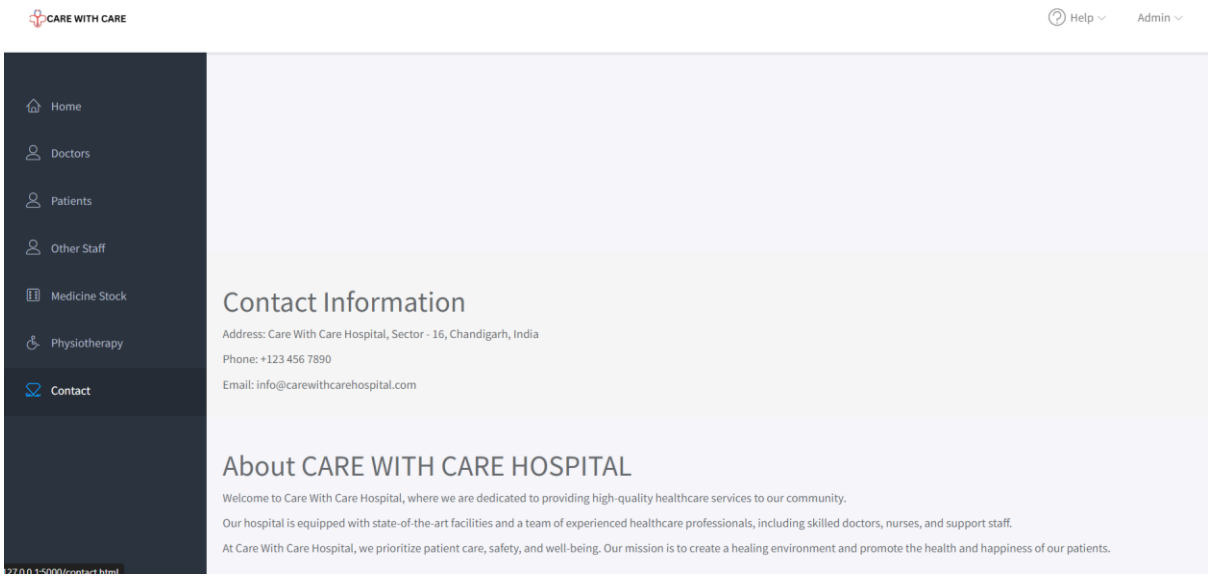

Fig A.7 – Appointments Page

8.2 Appendix B: USER MANUAL :

1. Managing Doctors:-

a. Adding a New Doctor:

- Click on "Doctors" in the sidebar.
- Click "Add Doctor" button.
- Fill in the required details.
- Save changes.

b. Updating Doctor Details:

- Navigate to the doctor list.
- Click on the doctor's name to edit.
- Update information.
- Save changes.

c. Viewing Doctor List:

- Access the "Doctors" section.
- View the list of doctors with relevant details.

2. Patient Management:

a. Adding a New Patient:

- Navigate to "Patients."
- Click "Add Patient" button.
- Enter patient details.
- Save changes.

b. Updating Patient Records:

- Go to the patient list.
- Click on the patient's name to edit.
- Update information.
- Save changes.

c. Viewing Patient List:

- Access the "Patients" section.
- View the list of patients with relevant details.

3. Medicine Stock:

a. Adding New Medicine:

- Visit "Medicine Stock."
- Click "Add Medicine" button.
- Input medicine details.
- Save changes.

b. Updating Medicine Details:

- Go to the medicine inventory.
- Click on a medicine to edit.
- Update information.
- Save changes.

c. Medicine Inventory Overview:

- Access "Medicine Stock" to view the inventory list

4. *Other Staff:*
    a. Adding Other Staff:
        - Navigate to "Other Staff."
        - Click "Add Other" button.
        - Enter staff details.
        - Save changes.
    b. Updating Other Staff Details:
        - Go to the staff list.
        - Click on staff name to edit.
        - Update information.
        - Save changes.
    c. Viewing Other Staff List:
        - Access the "Other Staff" section.
        - View the list of staff members.

5. *Physiotherapy Module*:
    a. Recording Physiotherapy Sessions:
        - Visit "Physiotherapy."
        - Click "Record Session" button.
        - Enter session details.
        - Save changes.

6. *Dashboard Analytics:*
    a. Exporting Data Reports:
        - Use the export button in relevant sections for detailed reports