UIT2511---Software Development Project – II

Hospital workforce scheduling and management system

Software Requirements Specification Document

Sri Sai Ankit V-3122215002118
Vasundhara B-3122215002119
Vemula Muni Karthik-3122215002120

Table of Contents

1. Introduction

1.1 Purpose:

The primary objective of the “Hospital Workforce Scheduling and Management System" (HWSMS) is to revolutionize the allocation and management of hospital staff, encompassing medical professionals, nursing personnel, and administrative staff. It is designed to establish a sophisticated, automated, and adaptable platform that efficiently organizes and assigns staff duties. By focusing on streamlining scheduling procedures, overseeing staff shifts, efficiently managing leave requests, and ensuring a well-balanced workforce, the system aims to elevate the quality of patient care. This document serves as a comprehensive guide for developers, providing an outline to construct the system effectively. Additionally, it acts as a reference point for potential clients, offering insights into the system's functionalities and how it significantly enhances hospital operations for improved patient outcomes and operational efficiency.

1.2 Scope:

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

- Managing and scheduling the workforce of a hospital, including doctors, nurses, receptionists, and administrators.
- The system offers a real-time view of schedules for different departments and staff roles.
- The system manages staff leave requests, ensuring replacement coverage.
- It reallocates schedules and ensures continuous departmental coverage when staff members are on leave.
- Providing administrators with the authority to view and manage the schedules of all hospital staff and approve leave requests.
- Enabling users (doctors, nurses, and receptionists) to view their own schedules, receive notifications, and request leave through the system.

- Utilizing Constraint Satisfaction Problem (CSP) algorithm to generate schedules based on staff availability, departmental requirements, and shift preferences.
- Storing and managing all relevant data, including employee information, schedules, and leave records, in a MongoDB database.

Out of Scope:

- Patient medical records management and EHR systems.
- Healthcare insurance management or claims processing.
- Medical training and education programs for healthcare staff.

1.3 Definitions, Acronyms, and Abbreviations:

Acronyms and Abbreviations:

A. HWSMS: Hospital Workforce Scheduling and Management System
B. SRS: Software Requirements Specification.
C. CSP: Constraint Satisfaction Problem
D. UI: User Interface
E. EHR: Electronic Health Records

Definitions:

A. Schedule: A predetermined plan or timetable for assigning hospital staff, including nurses, doctors, and receptionists, to their respective shifts and responsibilities.
B. Workforce: The collective group of employees, including doctors, nurses, receptionists, and administrators, responsible for various functions within the hospital.
C. EHR: Detailed digital records of a patient's health information that includes medical history, diagnoses, medications, treatment plans, immunization dates, allergies, radiology images, and laboratory test results.

D. NoSQL Database (MongoDB): A type of database system that uses a non-relational approach to data storage. MongoDB, specifically used in this system, provides a flexible and scalable database structure suitable for storing diverse hospital workforce scheduling and management data, allowing easy retrieval and manipulation.
E. Optimization: The process of making the HWSMS more efficient by improving algorithms, methodologies, or processes for staff scheduling, leave management, and overall resource utilization.
F. Shift: A specific period during which hospital staff members are assigned to work. Shifts can be defined by time blocks (e.g., morning, afternoon, night) and specific days of the week.

1.4 References:

- https://eczasopisma.p.lodz.pl/JACS/article/view/413
- https://www.researchgate.net/publication/330026395_Development_of_a_Doctor_Scheduling_System_A_Constraint_Satisfaction_and_Penalty_Minimisation_Scheduling_Model

1.5 Overview:

The Hospital Workforce Scheduling and Management System (HWSMS) document is structured as follows to provide a comprehensive overview of the system. Section 2 offers a system overview, detailing user proficiency expectations, general constraints, and assumed dependencies. Section 3 outlines specific system requirements, including functional aspects, performance criteria, and design constraints. Section 4 discusses potential future system enhancements. Lastly, Section 5 includes appendices for supplementary information.

2. Overall Description:

2.1 Product Perspective:

The "Hospital Workforce Scheduling and Management System" (HWSMS) is designed to address the complex task of staff scheduling and management within healthcare institutions. This system stands as a pivotal tool in the healthcare industry, positioned to efficiently handle the complexities associated with staff management. Its contextual significance lies in streamlining and optimizing staff allocation, addressing the critical need for a comprehensive and adaptable platform within healthcare institutions. In the marketplace, HWSMS distinguishes itself as an innovative, technology-driven system that enhances operational efficiency and quality of care. This section provides an understanding of the product's context and its positioning concerning related systems, both in the marketplace and in terms of research-oriented projects.

2.1.1 System Interfaces:

The integration with MongoDB offers a scalable and efficient solution for handling hospital staff records. Its NoSQL architecture allows flexibility in accommodating diverse data types, ensuring seamless storage and retrieval of intricate staff scheduling details. Flask's role in handling HTTP requests streamlines user interactions by mapping them to specific functions, ensuring a responsive and dynamic user experience.

2.1.2 Interfaces:

The web-based interface tailored for hospital staff ensures ease of use and accessibility. It empowers staff members to actively participate in managing their schedules and viewing shifts, fostering a sense of ownership and control. Additionally, the administrative interface facilitates efficient oversight of staff management, providing comprehensive tools for leave handling and staff allocation.

2.1.3 Software Interfaces:

Python's versatility enables robust backend logic implementation, ensuring efficient and scalable processing of hospital workforce-related tasks. MongoDB's NoSQL framework aligns with the dynamic nature of scheduling data, providing a flexible storage environment. Flask, functioning as the web framework, simplifies the development of interfaces, allowing for rapid development and deployment of user-friendly interfaces. JSON's role in data interchange promotes compatibility and ease of data sharing across various system components.

2.1.4 Communications Interfaces:

Flask's integration with HTTP protocols ensures seamless communication between the user interface and the backend functionality. This interaction enables quick and reliable data transfer, allowing users to access and modify scheduling information with minimal delay.

2.1.5 Memory Constraints:

The system dynamically manages memory allocation based on the operational demands and scale of data processed. MongoDB's flexibility in managing memory ensures efficient handling of varied data volumes, optimizing memory usage to sustain system performance.

2.1.6 Operations:

User authentication operations provide secure access, ensuring that only authorized personnel interact with the system. The scheduling and leave management operations facilitate efficient organization and planning of staff shifts, ensuring adequate coverage while handling employee absence effectively. Data retrieval from MongoDB enables quick access to scheduling details, enabling real-time decision-making regarding staff allocation and management. These operations, integrated and executed through Flask's defined functions, ensure seamless execution of staff-related tasks.

2.2 Product Functions:

The product (HWSMS) offers multiple functions including user authentication, scheduling, leave management, scheduling optimization using CSP, data storage, retrieval, and presentation. These functions collectively facilitate effective staff management, ensuring a balanced workforce and optimized schedules for improved patient care.

Class of Use Cases	Use Cases	Description of Use Cases
User Authentication	Login	Allows users (doctors, nurses, administrators) to securely log into the system using their credentials.
	Logout	Enables users to securely log out of their active sessions, ensuring data security and confidentiality.
Scheduling	View Schedule	Allows staff to view their assigned shifts and schedules for a specified period, aiding in planning their workdays.
	Allocate Shifts	Permits administrators to allocate shifts to staff members based on availability and departmental requirements.
Leave Management	Apply for Leave	Allows staff to submit leave requests, specifying the dates and reasons, for approval and scheduling adjustments.

	Approve/Reject Leave	Requests Enables administrators to review and manage leave requests by approving or rejecting them based on organizational policies.
Scheduling Optimization	Constraint Satisfaction Problem (CSP) Implementation	Optimizes scheduling by solving constraints such as shift rotations, day-offs, and doctor-patient allocation for efficient staff coverage.
Data Storage & Retrieval	Store Scheduling Data	Stores staff schedules, leave requests, and related data in MongoDB, ensuring secure and organized data storage.
	Retrieve Scheduling Information	Allows swift retrieval of scheduling information from the database for real-time monitoring and decision-making.
Presentation	Display Schedule	Presents staff schedules and allocation details in an organized and user-friendly manner via web-based interfaces.

2.3 User Characteristics:

The diverse user base of the HWSMS encompasses healthcare professionals in distinct roles, each engaging with the system based on their responsibilities. Doctors primarily access the system to oversee patient schedules and coordinate treatment plans. Nurses rely on the system for shift assignments and patient care coordination. Receptionists use it to manage appointments and handle patient inquiries. Administrators, having overarching system control, oversee staff management, handle leave requests, and ensure seamless system operation. Each user group interacts with tailored interfaces and functionalities, optimizing their workflow within the hospital environment.

2.4 Constraints:

Several constraints dictate the system's functionality and performance. Scalability stands as a crucial aspect, ensuring the system efficiently handles a growing user base and increasing data volume without compromising performance. Maintaining robust security measures is paramount to safeguard sensitive medical information stored within the system. Furthermore, optimizing scheduling algorithms to adapt to diverse shift preferences, varying staff availability, and specific departmental requirements remains a critical constraint for the system's effectiveness.

2.5 Assumptions and Dependencies:

The HWSMS operates under the assumption of consistent and reliable internet connectivity for uninterrupted user interactions and system functionalities. It relies on MongoDB for data storage, ensuring independence from external data sources or dependencies. Additionally, the system assumes that user access privileges are adequately managed based on roles and responsibilities, ensuring authorized access to relevant system features and data.

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

Description of purpose: Provide an intuitive interface for doctors, nurses, receptionists, and
administrators to interact with the system.

Source of input or destination of output: Users via computer terminals and devices.

Valid range, accuracy and/or tolerance: Intuitive and user-friendly design with error handling for
invalid inputs.

Units of measure: N/A

Timing: Real-time responsiveness to user interactions.

Relationships to other inputs/outputs: Integration with backend algorithms for scheduling and data retrieval.

Screen formats/organization: Clear and organized layout for easy navigation.

Window formats/organization*:* Intuitive window management for multitasking.

Data formats: Presentation of data in a readable format for users.

User-friendly commands for system interaction.

End messages: Confirmation messages for completed actions.

3.1.2 Hardware Interfaces

Description of purpose: Define hardware requirements for running the system.

Source of input or destination of output: Servers, user devices (computers, tablets, etc.).

Valid range, accuracy and/or tolerance:Compatible with standard hardware configurations.

Units of measure: N/A

Timing*:* N/A

Relationships to other inputs/outputs: Compatibility with various hardware components.

Screen formats/organization*:* N/A

Window formats/organization: N/A

Data formats: N/A

Command formats*:* N/A

End messages: N/A

3.1.3 Software Interfaces

Description of purpose: Define software requirements and integrations for the system.

Source of input or destination of output: Interaction with external software systems (e.g.,databases, AI algorithms).

Valid range, accuracy and/or tolerance: Compatibility with specified software versions and APIs.

Units of measure: N/A

Timing: Synchronous communication with external systems.

Relationships to other inputs/outputs: Interactions with databases, AI algorithms, and external services.

Screen formats/organization: N/A

Window formats/organization: N/A

Data formats: JSON formats for data exchange.

Command formats: N/A

End messages: N/A

3.1.4 Communications Interfaces

Description of purpose: Define communication protocols and requirements.

Source of input or destination of output*:* Network protocols for data transmission.

Valid range, accuracy and/or tolerance: Reliable
and secure communication protocols (e.g., HTTPS).

Units of measure: N/A

Timing: Real-time data transmission for critical
Operations.

Relationships to other inputs/outputs: Integration with network infrastructure for data exchange.

Screen formats/organization: N/A

Window formats/organization: N/A

Data formats: N/A

Command formats: N/A

End messages: N/A

3.2 Functions

3.2.1 Scheduling Algorithm

Description of purpose: Implement an AI-based CSP algorithm to efficiently schedule doctors,nurses, and receptionists on a weekly basis.

Input(s): Availability of doctors, nurses, and receptionists, patient appointment data, emergency cases.

Output(s): Weekly schedules for doctors, nurses, and receptionists, notifications for schedule changes.

Processing sequence: Validate input data, apply CSP algorithm to generate schedules, handle emergency cases, notify users of changes.

Responses to abnormal situations: Handle cases of conflicting schedules, ensure emergency cases are attended to promptly.

Relationships to other functions*:* Integrates with user interfaces for input and output,
communicates with database for data retrieval and storage.

Formulas for input to output conversion: N/A

3.2.2 Leave Management

Description of purpose: Provide a system for
doctors, nurses, and receptionists to request leaves and for administrators to approve/reject them

Input(s): Leave requests, admin approvals/rejections.

Output(s): Updated schedules, notifications to affected parties.

Processing sequence: Receive leave requests,validate and forward to admin, update schedules upon approval, notify relevant parties.

Responses to abnormal situations: Handle cases of conflicting leave requests, ensure compensatory schedules are set.

Relationships to other functions: Integrates with user interfaces for leave requests and approvals, communicates with scheduling algorithm for updates.

Formulas for input to output conversion: N/A

3.3 Performance Requirements

1)95% of scheduling operations shall be completed in less than 2 seconds.

2)The system shall support up to 500 simultaneous users.

3.4 Logical Database Requirements

Types of information: User profiles, schedule data, patient appointments, leave records.

Frequency of use: Daily for scheduling, occasional for leave management.

Accessing capabilities: Read and write access for authorized users.

Data entities and their relationships: Users,schedules, appointments, leaves (with appropriate relationships).

Integrity constraints: Ensure data consistency through appropriate constraints.

Data retention requirements*:* Keep historical scheduling data for at least 2 years.

3.5 Design Constraints

3.5.1 Standards Compliance

The system shall comply with relevant healthcare data privacy standards (e.g., HIPAA).

The UI shall follow industry best practices for accessibility.

3.6 Software System Attributes

3.6.1 Reliability

The system shall have a Mean Time Between Failures (MTBF) of at least 10,000 hours.

3.6.2 Availability

The system shall have an availability of 99.9% during normal operating hours.

3.6.3 Security

The system shall employ encryption for sensitive data transmission.
User authentication shall be required for access.

3.6.4 Maintainability

The system shall be modular for ease of maintenance and updates.

3.6.5 Portability

The system shall be compatible with Windows, macOS, and Linux operating systems.

3.7 Organizing the Specific Requirements

3.7.1 System Modes

Training Mode:
The system shall operate in a training mode to develop and enhance scheduling algorithms. During this mode, the system analyzes historical scheduling data, employee preferences, and leave patterns to refine the scheduling model.
Prediction Mode:

The system shall operate in a prediction mode for real-time scheduling generation. It accepts user inputs regarding shift requirements, employee availability, and leave requests to generate optimized schedules in real-time for the upcoming week.

3.7.2 User Class:We describe the functional requirements by giving various use cases.

Use case related to installation:

Use Case 1: Login

Primary Actor: User

Pre Condition: Nil

Main Scenario :

1. Start the application. User prompted for login and password.
2. User gives the login and password.
3. System does authentication.

4. Main screen is displayed.

Alternate Scenario :

1. Authorization fails

1(a). Prompt the user that he typed the wrong password

1(b). Allow him to re-enter the password. Give him 3 chances.

Use case 2: Entering employee details

Primary Actor: Admin

Precondition: Admin should be known about employee details

Main scenario

1. User initiates the “add Employee” functionality.

2. System prompts the user to enter details such as I, email,Name,Category,Department etc..

3. System stores the inputted Employee details

Alternate Scenario:

1 Invalid data is entered.

1(a)System displays an error message and prompts the user to enter the correct health data.

Use case 3: Entering Department

Primary Actor: Admin

Precondition: Admin should be known No of employee need for that department

Main scenario

1. User initiates the “add department” functionality.

2. System prompts the user to enter details such as Id,name,specialization,,number of employees

3. System stores the inputted department details

Alternate Scenario:

1 Invalid data is entered.

1(a). System displays an error message and prompts the user to enter the correct health data

Use case 4: View requests

Primary Actor: Admin

Precondition: Nil

Main scenario

1. User initiates the "View requests" functionality.
2. System prompts the leave requests sent by employees
3. System stores the inputted given input

Use case 5: Request leave

Primary Actor: User

Precondition: Schedule of the employee

Main Scenario

1. User initiates the "Request leave" functionality.
2. System send's notification to admin

Use case 6: Emergency

Primary Actor: Admin/User

Precondition: Place of EMERGENCY

Main Scenario

1. User initiates the “Emergency” functionality.
2. System ask to enter emergency details
3. System will send notification to all employees in data base

3.7.3 Objects:

Employee Object: Represents an individual (nurses, doctors, receptionists) interacting with the Scheduling System.

Leave Request Object: Represents a submitted request for absence from an assigned shift.

Schedule Object: Represents the scheduled shifts and assigned tasks for employees within the Scheduling System.

3.7.4 Features:

Scheduling Input Processing:

Submit Leave Request: Users can request leave by submitting the duration and reason for absence.

Shift Assignment: The system automatically assigns shifts based on employee availability and preferences.

Extract Schedule Metrics:

Extract Mean Working Hours: The system computes the mean working hours for each employee over a specified period.

Shift Coverage Analysis: Analyzes shift coverage to ensure adequate staffing levels across departments.

Leave Prediction:

Evaluate Leave Requests: The system uses historical data and predefined rules to evaluate the impact of pending leave requests on shift coverage.

Display Replacement Assignment: Displays replacement assignments for absent employees based on leave duration and shift requirements.

3.7.5 Stimulus:

An employee submits a leave request through the system's user interface.

The system processes submitted leave requests and analyzes the impact on shift coverage.

3.7.6 Response:

The system evaluates the submitted leave request, checking its duration and assessing its impact on scheduled shifts.

If necessary, the system generates replacement assignments based on predefined rules and available resources.

3.7.7 Functional Hierarchy:

(a) User Interface (UI) Layer:

Leave Request Module: Manages the user interface for employees to submit leave requests.

Shift Assignment Module: Displays assigned shifts and allows for employee preferences to be set.

(b) Scheduling Engine:

Leave Impact Analysis Module: Assesses the impact of pending leave requests on existing shift coverage.

Replacement Assignment Module: Automatically assigns replacements for absent employees based on leave duration and shift requirements.

(c) Reporting and Integration Layer:

Shift Metrics Generation Module: Generates reports summarizing working hours and shift coverage metrics.

(d) Data Management Layer:

Data Update Module: Updates the scheduling dataset with new leave requests, shift assignments, and replacements.

4)Change Management Process

In the context of the Scheduling System, a change management process is fundamental to accommodate enhancements and adjustments while ensuring scheduling accuracy and efficiency.

Change Request Submission:

Employees, team members, or administrators can submit change requests based on their experiences and system interactions. These requests may involve refining scheduling algorithms, adjusting shift assignment parameters, or proposing improvements to the user interface.

Impact Assessment:

Each proposed modification undergoes a comprehensive assessment to determine its potential impact on scheduling accuracy and operational efficiency. This evaluation includes analyzing how changes may affect the system's ability to generate optimal schedules, manage leave efficiently, and ensure adequate shift coverage.

Continuous Learning from Scheduling Data:

The system integrates adaptive learning mechanisms to continually evolve based on scheduling data and outcomes. This ongoing learning process aims to enhance

the system's capability to interpret varying scheduling patterns, preferences, and historical data, refining its ability to generate more effective and tailored schedules.

Validation and Testing:

Changes and enhancements undergo rigorous testing to validate their impact on scheduling accuracy and overall system performance. Thorough testing procedures guarantee that modifications improve scheduling efficiency without compromising the core functionality of the system.

5. Document Approvals

Version	Date	Approved By	Changes Made
1.0	2/10/2023	Dr. G.Sornavalli	Initial version
1.1	16/10/2023	Dr. G.Sornavalli	Initial raw features
1.2	6/11/2023	Dr. G.Sornavalli	Added performance requirements

5.1 Change Control

All proposed changes to this documentation were submitted to the Project Lead or designated authority. The Project Lead reviewed and assessed the potential impact of changes on the scheduling system. Upon approval, updates were incorporated into the system's development roadmap and managed through the version control system.

6.Supporting Information

(a) Real-time Example - Workforce Optimization System:

Overview: The Workforce Optimization System is an AI-driven platform used in workforce management, optimizing schedules and task assignments.

Process: Employees input availability and preferences, and the system generates optimized schedules considering staffing needs and individual preferences.

User Influence: Employee feedback on shift assignments and schedule preferences contributes to system learning, enhancing its ability to generate tailored and efficient schedules.

Output: The system produces optimized schedules, improving workforce productivity and operational efficiency.

(b) Real-time Example - ShiftPlanning:

Overview: ShiftPlanning is an AI-powered scheduling platform used in various industries to manage shifts and employee schedules.

Process: Managers input employee availability and skill sets, and ShiftPlanning generates optimized schedules considering skill requirements and shift coverage.

User Influence: Employee feedback on schedule preferences and skill development contributes to the continuous improvement of ShiftPlanning's scheduling algorithms.

Output: The platform provides optimized schedules, ensuring efficient task allocation and maximum productivity.

Benefits of Scheduling System AI:

Efficient Resource Allocation: AI-driven scheduling optimizes workforce utilization, ensuring optimal staffing levels and task distribution, enhancing operational efficiency.

Adaptive Learning: The system learns from scheduling outcomes and user preferences, continually improving scheduling accuracy and flexibility.

Enhanced Productivity: AI in scheduling streamlines task assignments, reduces conflicts, and ensures smoother operations, resulting in increased productivity.

7.Future Extensions

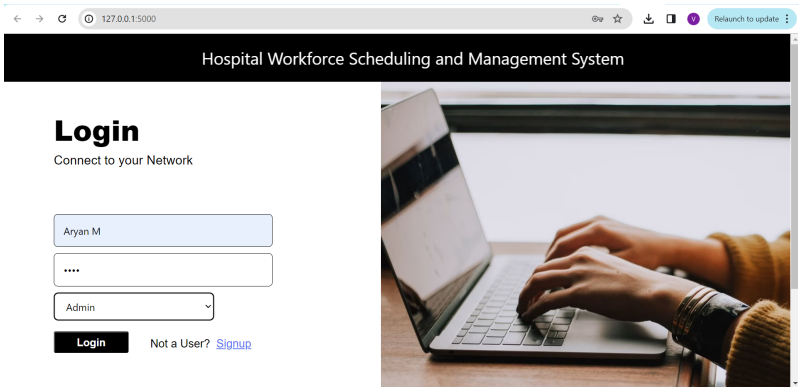
Advanced Algorithm Exploration: Explore advanced optimization algorithms or machine learning techniques to enhance scheduling accuracy and efficiency further.

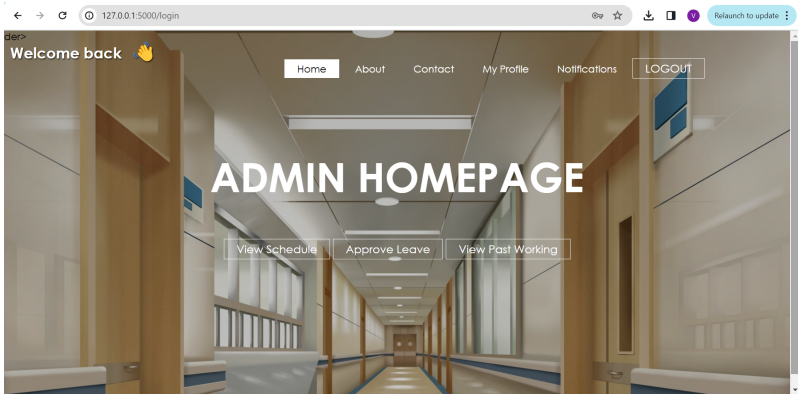
Mobile Application Integration: Develop a mobile application for users to access scheduling information on-the-go, ensuring seamless integration with the existing web platform.

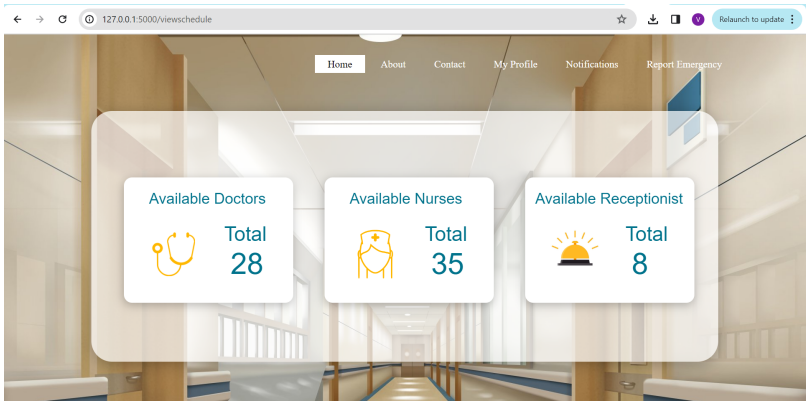
Real-Time Monitoring and Analytics: Implement real-time monitoring features for ongoing analysis of scheduling data, enabling swift adjustments and improvements. Explore cloud integration for scalability and enhanced data processing capabilities.

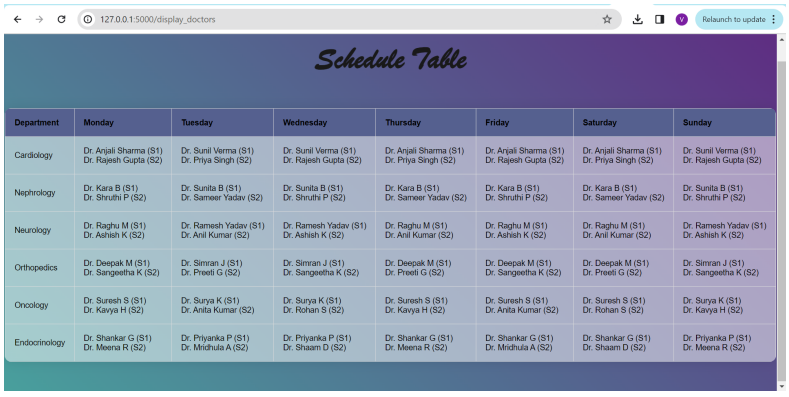
8.Appendix

Appendix A: Sample Outputs









Department	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Cardiology	Dr. Anjali Sharma (S1) Dr. Rajesh Gupta (S2)	Dr. Sunil Verma (S1) Dr. Priya Singh (S2)	Dr. Sunil Verma (S1) Dr. Rajesh Gupta (S2)	Dr. Anjali Sharma (S1) Dr. Priya Singh (S2)	Dr. Anjali Sharma (S1) Dr. Rajesh Gupta (S2)	Dr. Anjali Sharma (S1) Dr. Priya Singh (S2)	Dr. Sunil Verma (S1) Dr. Rajesh Gupta (S2)
Nephrology	Dr. Kara B (S1) Dr. Shruthi P (S2)	Dr. Sunita B (S1) Dr. Sameer Yadav (S2)	Dr. Sunita B (S1) Dr. Shruthi P (S2)	Dr. Kara B (S1) Dr. Sameer Yadav (S2)	Dr. Kara B (S1) Dr. Shruthi P (S2)	Dr. Kara B (S1) Dr. Sameer Yadav (S2)	Dr. Sunita B (S1) Dr. Shruthi P (S2)
Neurology	Dr. Raghu M (S1) Dr. Ashish K (S2)	Dr. Ramesh Yadav (S1) Dr. Anil Kumar (S2)	Dr. Ramesh Yadav (S1) Dr. Ashish K (S2)	Dr. Raghu M (S1) Dr. Anil Kumar (S2)	Dr. Raghu M (S1) Dr. Ashish K (S2)	Dr. Raghu M (S1) Dr. Anil Kumar (S2)	Dr. Ramesh Yadav (S1) Dr. Ashish K (S2)
Orthopedics	Dr. Deepak M (S1) Dr. Sangeetha K (S2)	Dr. Simran J (S1) Dr. Preeti G (S2)	Dr. Simran J (S1) Dr. Sangeetha K (S2)	Dr. Deepak M (S1) Dr. Preeti G (S2)	Dr. Deepak M (S1) Dr. Sangeetha K (S2)	Dr. Deepak M (S1) Dr. Preeti G (S2)	Dr. Simran J (S1) Dr. Sangeetha K (S2)
Oncology	Dr. Suresh S (S1) Dr. Kavya H (S2)	Dr. Surya K (S1) Dr. Anita Kumar (S2)	Dr. Surya K (S1) Dr. Rohan S (S2)	Dr. Suresh S (S1) Dr. Kavya H (S2)	Dr. Suresh S (S1) Dr. Anita Kumar (S2)	Dr. Suresh S (S1) Dr. Rohan S (S2)	Dr. Surya K (S1) Dr. Kavya H (S2)
Endocrinology	Dr. Shankar G (S1) Dr. Meena R (S2)	Dr. Priyanka P (S1) Dr. Mridhula A (S2)	Dr. Priyanka P (S1) Dr. Shaam D (S2)	Dr. Shankar G (S1) Dr. Meena R (S2)	Dr. Shankar G (S1) Dr. Mridhula A (S2)	Dr. Shankar G (S1) Dr. Shaam D (S2)	Dr. Priyanka P (S1) Dr. Meena R (S2)