

Hospital Workforce Scheduling and Management System

UIT2511 – SOFTWARE DEVELOPMENT PROJECT – I

A PROJECT REPORT

Submitted by

Sri Sai Ankit V (3122 21 5002 118)

Vasundhara B (3122 21 5002 119)

Vemula Muni Karthik (3122 21 5002 120)

**SSN COLLEGE OF ENGINEERING,
KALAVAKKAM**

DECEMBER 2023

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project titled “HOSPITAL WORKFORCE SCHEDULING AND MANAGEMENT SYSTEM” is the bonafide work of “Sri Sai Ankit V - 3122215002118, Vasundhara B - 3122215002119, Vemula Muni Karthik - 312221500120”, and is submitted for project viva-voce examination held on

.....

Signature of examiner(s)

INDEX

1) Abstract.....
2) Introduction.....
I. Motivation
II. Objective
III. Problem Statement
IV. Deliverables
3) Requirements engineering.....
4) Risk Management.....
5) Project Management.....
6) Project Outcomes.....
7) Conclusion and Further Scope.....
8) Courses of Outcome.....
9) References.....

1. ABSTRACT

The Hospital Workforce Scheduling and Management System (HWSMS) is a solution centered around AI-driven scheduling, designed to optimize staff allocation, enhance patient care, and streamline hospital operations. This system integrates advanced scheduling algorithms that prioritize staff preferences, ensure designated days off, and dynamically allocate doctors, nurses, and receptionists based on demand and availability.

Utilizing the CSP Scheduling Algorithm enhanced with Forward Checking, the HWSMS refines assignments, resolves conflicts, and generates optimized schedules while considering individual staff preferences and specific time-off requests. It provides a user-friendly interface for staff to view schedules, request leaves, and receive approvals efficiently.

Built on HTML, CSS, JavaScript for the frontend, Python (Flask) for the backend, and MongoDB for the database where the details of doctors, nurses, admin officials and receptionists are stored, the system emphasizes flexibility and scalability. Embracing an Agile development approach, it enables iterative enhancements and efficient collaboration among team members.

It streamlines operations, improves staff well-being by automating routine tasks and reducing burnout, while significantly enhancing patient care through reduced waiting times and optimized resource allocation. The HWSMS represents a transformative solution that harmonizes staff allocation, boosts operational efficiency, and prioritizes both staff and patient satisfaction within healthcare settings.

2. INTRODUCTION

Motivation

The purpose of developing a Hospital Workforce Scheduling and Management System stems from the pressing need within healthcare institutions to streamline operations, enhance patient care, and prioritize the well-being of healthcare professionals. The existing challenges in scheduling staff efficiently, ensuring optimal resource allocation.

Problem Statement

The problem addressed by the proposed Hospital Management System revolves around the inefficiencies and challenges prevalent in traditional healthcare staff scheduling processes. Current systems often rely on manual methods that struggle to accommodate the complex and dynamic needs of a healthcare facility. These challenges include:

- Inefficient Staff Allocation
- Lack of Consideration for Individual Preferences
- Operational Inefficiencies
- Staff Well-being and Burnout

Addressing these issues necessitates the development of an innovative Hospital Management System that leverages advanced AI-driven scheduling algorithms. Such a system would not only streamline staff scheduling processes but also cater to individual preferences, ensure equitable workload distribution, and enhance overall operational efficiency.

Objectives

1. Efficient Staff Scheduling: Developing an advanced scheduling algorithm to efficiently allocate doctors, nurses, and receptionists based on their availability, preferences, and leave requests.
2. Staff Well-Being: Prioritizing the well-being of healthcare professionals by considering their preferences and allowing them to request leaves without disrupting the hospital's operations.

3. Dynamic Staff Allocation: The system dynamically allocates doctors, nurses, and receptionists based on real-time demand and availability. This responsiveness ensures that the right personnel are assigned to meet patient needs effectively.

4. Operational Efficiency: Utilizing AI-based scheduling to improve overall operational efficiency, reduce scheduling conflicts, and minimize disruptions in day-to-day hospital operations

Deliverables

The Hospital Management System's core is its **AI-Driven** Scheduling Algorithm, utilizing **Constraint Satisfaction Problem** methods with Forward Checking and Backtracking. This algorithm dynamically allocates staff based on real-time data, considering individual preferences and designated days off for fair and **efficient scheduling**. To support this algorithm, a scalable **MongoDB** database architecture is implemented. This architecture efficiently stores and manages scheduling data, staff preferences, and operational information required for the scheduling algorithm. This ensures that all relevant data is easily accessible and organized for the algorithm to make informed decisions. User-Friendly Interfaces enable seamless access to schedules and leave requests, interacting effortlessly with the database and scheduling algorithm. Integration of Analytics Tools provides continuous performance evaluation, enabling data-driven decision-making for optimizing operational efficiency and enhancing patient care.

3. Requirements Engineering

Functional Requirements:

1. Predefined Employee Data Representation:

- Design the system to represent fixed, predefined data for doctors, nurses, and receptionists, including their availability, shift preferences, and days off.

2. Constraint Management:

- Implement constraints based on the predefined data, such as fixed shift preferences and specific days off for each employee.

3. Leave Request Simulation Mechanism:

- Develop a simulated leave request feature within the system to test its ability to adjust schedules based on predefined scenarios.

4. Automated Schedule Adjustment:

- Create algorithms within the scheduling system to automatically adjust the predefined schedule when simulated leave requests occur, ensuring proper shift coverage while adhering to predefined constraints.

5. Replacement Assignment Logic:

- Define logic to manage replacements in the predefined schedule when an employee is on leave, adhering to the predefined rules for assigning alternative staff or general duty employees.

6. Admin Override Capability:

- Enable administrative overrides to manually adjust the predefined schedule in exceptional circumstances or unexpected changes.

Non-functional Requirements:

1. Performance with Fixed Data:

- Ensure the scheduling algorithm performs efficiently despite the fixed, predefined nature of the data, maintaining a reasonable schedule generation time.

2. Reliability of System Operations:

- Ensure the system operates reliably without errors, consistently producing scheduling outcomes based on the predefined constraints and rules.

3. Usability of User Interface:

- Design an intuitive user interface that allows viewing and interaction with the predefined schedule, enabling ease of use for administrators to oversee and manually adjust the schedule if necessary.

4. Accuracy in Constraint Adherence:

- Validate that the system accurately manages and respects the predefined constraints, ensuring the generated schedule complies with the fixed rules.

5. Compatibility with Hospital Environment:

- Ensure compatibility of the scheduling system with the hospital's existing technology and environment.

6. Documentation and Maintenance Support:

- Provide comprehensive documentation for the system's functionality and maintenance procedures to facilitate ongoing support and updates

Epic	Sprint	User Story #	User Story	Essential/Desirable	Need of requirement	Description of requirement
1	1	1	Research on Domain	Essential	Gain insights for informed system development	Understand the CSP scheduling algorithm and how to implement it for the problem statement
		2	Data Collection	Critical	Collect relevant data for implementing backend	Gather specific scheduling data for nurses, doctors, and receptionists. Collect information on their shift preferences.

		3	Data Preprocessing	Essential	Clean data for effective model training.	Cleanse and prepare the collected data to facilitate the scheduling algorithm. Ensure the data is formatted appropriately for effective utilization in the scheduling model.
2	4	4	Login Page UI Dev	Essential	Create an intuitive login and register page interface.	Develop a prototype for the main page UI
	5	5	Admin Page UI Dev	Essential	Design an informative admin page interface.	Backend functionalities, such as handling user inputs, leave requests ,storing data, and implementing the CSP algorithm were developed.
	6	6	Employee page UI DEV	Essential	Implement all the employee pages.	Implement the interface between the pages and backend functionalities
	7	7	Deal with data storage , choosing appropriate database options	Essential	Study about MongoDB and use it for data storage	Create database for storing details of employees, storing the requests for leave and storing details of schedule

2	3	8	Implementing CSP algorithm	Essential	Develop CSP algorithm for working schedule of employee	Implement CSP algorithm to ensure that a schedule is created keeping in mind the shift preferences and the day of leave for the employees
		9	Handle leave requests	Essential	Implement a way to handle leave requests keeping some constraints and compensation for the leave of employee.	Scheduling a employee working in same shift if only a day of leave taken else appointing a 24/7 general duty employee(doctor/nurse). Ensuring that leave of employee is approved only if it is informed prior to 2 days and less than or equal to 4 leaves a month
		10	Testing with Various cases	Essential	Assess system performance with various testcases.	Assess algorithms performance in handling edge cases and check if it is working as expected
	4	11	Integration with Flask Phase 1	Essential	Integrate Backend with flask	Integrate the system with Flask (Phase 1) for backend functionality.

		12	Integration with Flask Phase 2	Essential	Ensure seamless frontend-backend communication	Integrate the system with Flask (Phase 2) for seamless frontend-backend communication.
		13	Deployment	Desirable	For public access.	Deploy the system online for public access and use.

4.Risk Management:

Anticipated Risks:

1. Inadequate or insufficient scheduling data may affect the accuracy of the CSP algorithm's scheduling output. (Moderate)
2. Difficulty in meeting essential scheduling requirements within the designated time frame. (Moderate)
3. Potential alterations in labor or healthcare regulations might impact the scheduling platform's compliance. (Moderate)

Mitigation Plan:

1. Apply robust data preprocessing techniques to handle discrepancies and ensure the quality of scheduling data for accurate algorithmic outcomes.
2. Utilize Agile methodologies or similar project management frameworks effectively, breaking down tasks into manageable units to ensure essential scheduling requirements are met within stipulated timelines.
3. Maintain constant awareness and vigilance regarding changes in labor and healthcare regulations. Design the scheduling platform with adaptability to swiftly accommodate evolving regulatory requirements.

Implementation and Risk Management

Name: Sri Sai Ankit V

Register Number: 3122 21 5002 118

Roles: Developer, Scrum Master

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of requirement
	1		Research on Domain	Essential	Gain insights for informed system development.	Understand the problem statement and which AI algorithm is best to implement
	2	3	Implementing CSP algorithm	Essential	Develop CSP algorithm for working schedule of employee.	Implement CSP algorithm to ensure that a schedule is created keeping in mind the shift preferences and the day of leave for the employees

		Handle leave requests	Essential	Implement a way to handle leave requests keeping some constraints and compensation for the leave of employee.	Scheduling a employee working in same shift if only a day of leave taken else appointing a 24/7 general duty employee(doctor/nurse) Ensuring that leave of employee is approved only if it is informed prior to 2 days and less than or equal to 4 leaves a month
--	--	-----------------------	-----------	---	---

Risk Management 1:

Risk #	Risk Description	Probability	Impact	Mitigation Plan
1	Scrum Master's lack of availability due to conflicting responsibilities	High	High	Plan and prioritize the Scrum Master's responsibilities to minimize conflicts with backend programming tasks. Allocate additional resources or delegate specific Scrum Master duties to other team members when necessary to ensure the smooth

				functioning of the Scrum process
2	Insufficient knowledge or expertise in Scrum processes impacting project efficiency	Moderate	Moderate	<p>Provide training and resources to enhance the Scrum Master's knowledge of Scrum methodologies and best practices.</p> <p>Encourage participation in relevant workshops or certification programs.</p> <p>Collaborate with the Scrum Master to create a learning plan to continuously improve their Scrum expertise.</p>

Test Log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	Trying for different testcases that is when the employee requests for two days leave or only one day leave etc	Leave request from Employee	Leave request has been handled successfully and schedule has been generated	PASS

2	2	Creating a proper schedule subjecting to all constraints from the employees	Predefined constraints and domain from users	Proper schedule	PASS
3	3	Sprint planning meeting	User stories and backlog	Prioritized backlog and sprint plan created	PASS
4	4	Sprint review meeting	Completed user stories and demos	Stakeholder satisfaction and feedback	PASS

Name: Vasundhara B

Register Number: 3122 21 5002 119

Roles: Developer

Implementation

Epic	Sprint	User Story #	User Story	Essential/Desirable	Need of requirement	Description of requirement
2	4	11	Project Integration w/ Flask Phase 1	Essential	Integrate project with Flask for backend functionality.	Integrate the project with Flask (Phase 1) for backend functionality.

		12	Integration w/ Flask Phase 2	Essential	Ensure seamless frontend-backend communication	Integrate the system with Flask (Phase 2) for seamless frontend-backend communication.
		13	Deployment	Essential	For public access.	Deploy the system online for public access and use.

Risk Management 2

Risk #	Risk Description	Probability	Impact	Mitigation Plan
1	Problem in enabling seamless frontend backend communication	High	High	Plan and have a complete understanding of both the frontend and the backend, and the overall flow of the program logic to ensure the integration goes on smoothly

Test log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	Try various test cases	Give different inputs	Seamless and Error free frontend-backend communication	PASS

Implementation and Risk Management

Name: Vemula Muni Karthik

Register Number: 3122 21 5002 120

Roles: Developer

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of requirement
1	1		Home for	Essential	Gain insights for informed	understanding requirements of user

			Admin and employee		system development.	
2	2	3	View requests page for admin	Essential	To view the current requests of employees	Implementing the view request page using html,css, and js
3	3	9	Request leave page	Essential	Implement a way to request leave to take leave to admin	The page will display a calender only days with has duty and employee will select data from calender and sends request
3	1		View schedule	Essential	Implement Page to the view schedule of all employees for a week in all departments	The page will display a table contains departments and assigned staff with respect to the day

Risk Management3

Risk #	Risk Description	Probability	Impact	Mitigation Plan
1	An uninteractive frontend for viewing schedule	High	High	Apply concepts of HTML,CSS and JS to ensure the frontend is designed well and working as per requirements
2	Ineffective request leave page	Moderate	Moderate	Ensure the frontend and the database concepts are applied properly so leave requests are taken and added to the database.

Test log Report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	Trying for to put leave on same day	Leave request to admin	The leave is already added in database and date blocked in calendar	PASS
2	2	Removing the requests form screen if he/she click either Accept or deny	Leave requests from employee	Unambiguous	PASS

5.. PROJECT MANAGEMENT

Sprint 1:

The screenshot shows a project management interface with the following details:

- Sprint Summary:** User Authentication-Database, 13 Oct – 27 Oct (3 issues). Status: 0 / 0 / 3. Buttons: Complete sprint, ...
- Tasks:**
 - HWSMS-15 Implement User Login Functionality (Status: DONE, assigned to user icon)
 - HWSMS-16 Create Collections for doctor, nurses and admins (Status: DONE, assigned to user icon)
 - HWSMS-17 Test User Authentication (Status: DONE, assigned to user icon)
- Buttons:** + Create issue

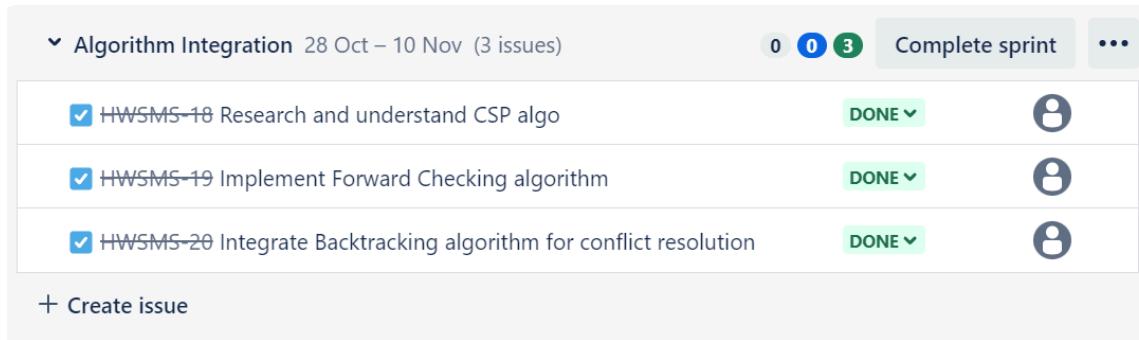
Sprint 2:

Algorithm Integration 28 Oct – 10 Nov (3 issues)

0 0 3 Complete sprint ...

<input checked="" type="checkbox"/> HW5MS-18 Research and understand CSP algo	DONE	
<input checked="" type="checkbox"/> HW5MS-19 Implement Forward Checking algorithm	DONE	
<input checked="" type="checkbox"/> HW5MS-20 Integrate Backtracking algorithm for conflict resolution	DONE	

+ Create issue



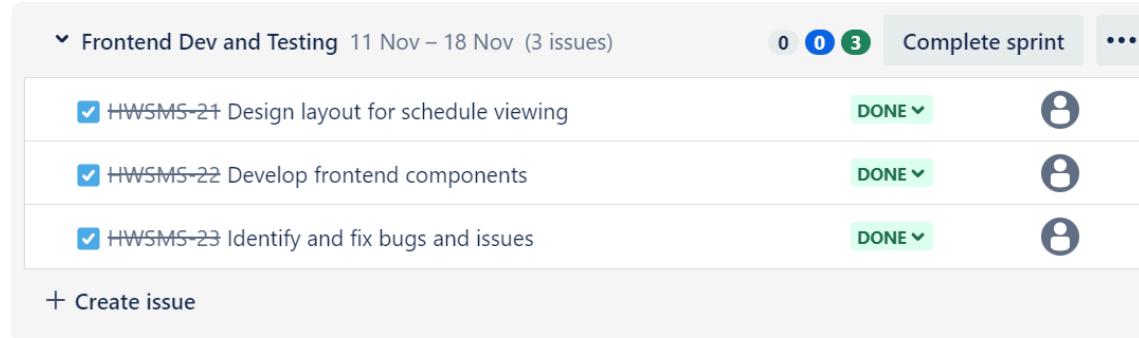
Sprint 3:

Frontend Dev and Testing 11 Nov – 18 Nov (3 issues)

0 0 3 Complete sprint ...

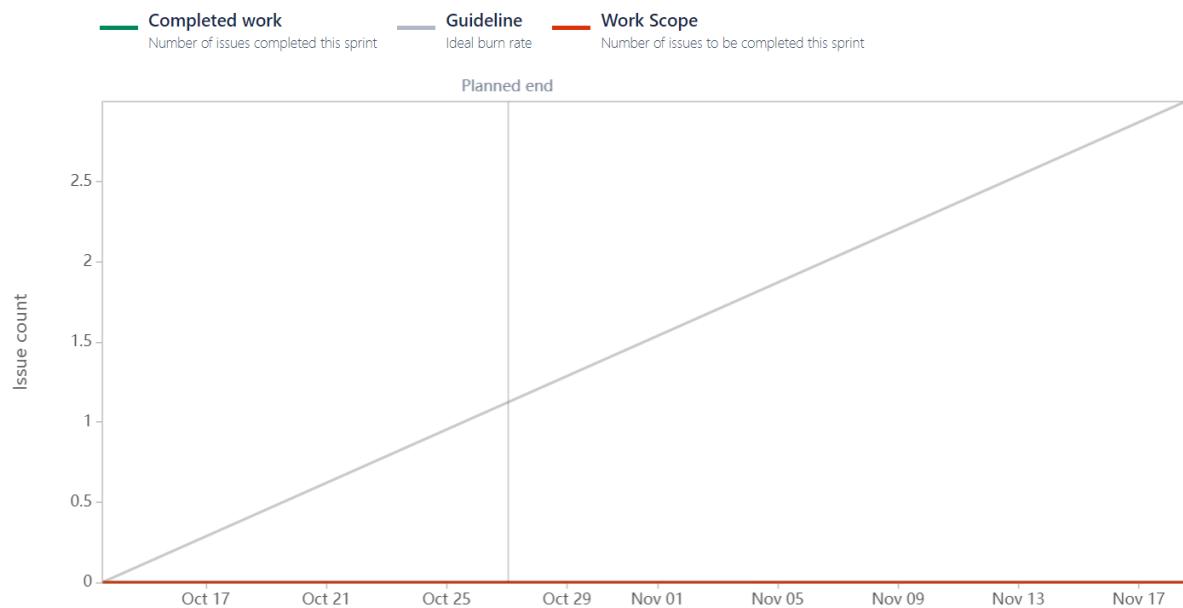
<input checked="" type="checkbox"/> HW5MS-21 Design layout for schedule viewing	DONE	
<input checked="" type="checkbox"/> HW5MS-22 Develop frontend components	DONE	
<input checked="" type="checkbox"/> HW5MS-23 Identify and fix bugs and issues	DONE	

+ Create issue



Sprint 1 Burnup Chart

Date - October 13th, 2023 - October 27th, 2023



Sprint 2 Burnup Chart

Date - October 28th, 2023 - November 10th, 2023

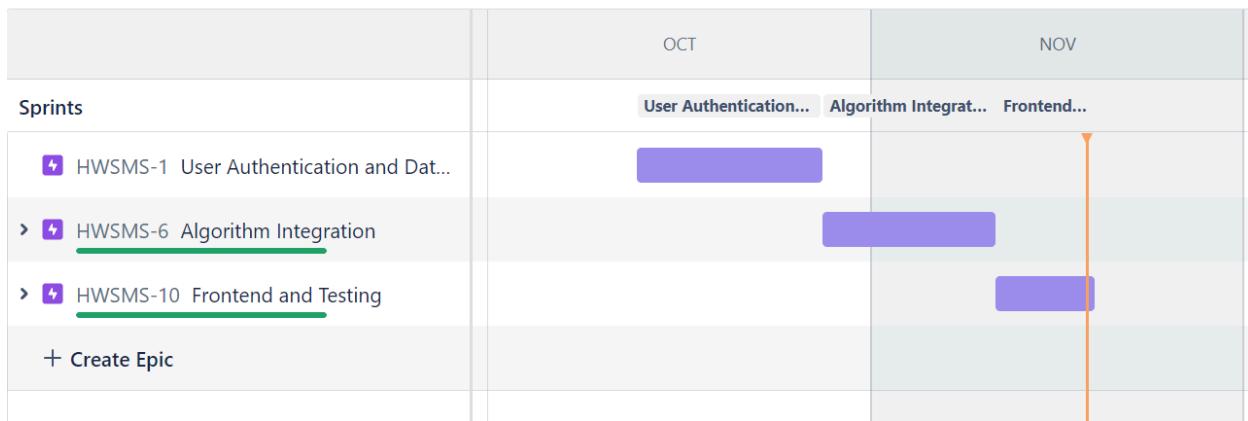


Sprint 2 Burndown Chart

Date - October 28th, 2023 - November 10th, 2023



Timeline



6. Project Outcomes

Scheduling System Development:

Developed a comprehensive scheduling system utilizing a Constraint Satisfaction Problem (CSP) algorithm to manage and optimize schedules for nurses, doctors, and receptionists over a seven-day period.

Leave Management and Replacement System:

Implemented a robust leave management system allowing employees to request leave and administrators to approve or reject requests. Developed a replacement algorithm to seamlessly fill in for absent employees based on the duration and nature of leave.

Real-time Schedule Analysis:

Enabled real-time analysis and visualization of schedules, displaying shift coverage, leave status, and replacements to ensure continuous and efficient operations.

Heuristic Model for Optimal Scheduling:

Designed a heuristic model to predict and optimize scheduling outcomes based on various parameters such as employee preferences, working hours, and shift rotations. This model aids in generating efficient schedules while meeting staffing requirements.

Integration of Agile Methodologies for Optimization:

Employed Agile methodologies for project management, utilizing tools like JIRA to streamline tasks, manage timelines, and ensure the completion of essential scheduling requirements within the specified time frame.

Data Analysis and Model Refinement:

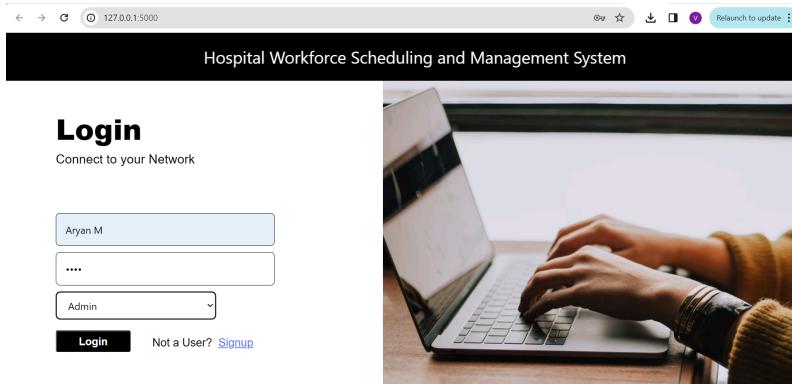
Analyzed historical scheduling data to refine and validate the heuristic model, ensuring its accuracy and adaptability to dynamic scheduling needs.

Testing and Validation:

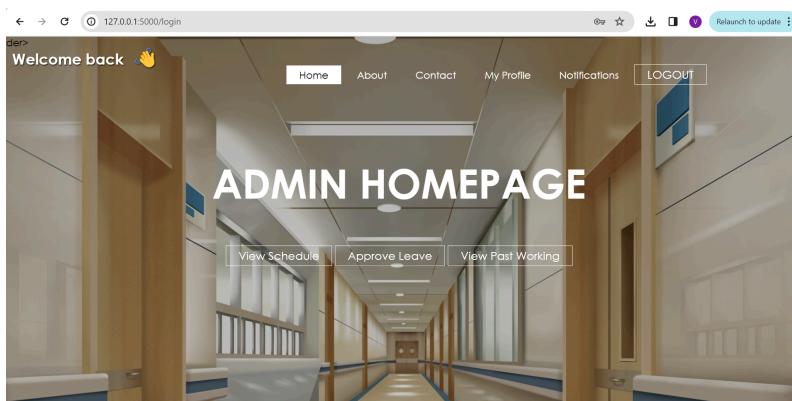
Conducted rigorous testing and validation of the scheduling system to ensure reliability, accuracy, and compliance with specified constraints and regulations in the healthcare environment.

OUTPUT SCREENSHOTS

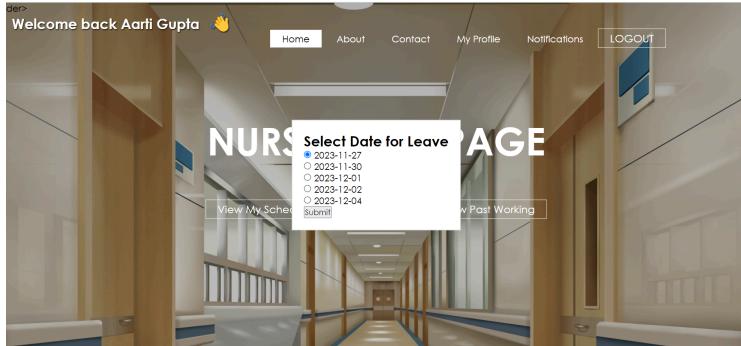
Login Page for Employee



Homepage for Admin



Homepage for Nurse



Schedule for Doctor

Schedule Table							
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. Priya Singh (S2)	Dr. Anjali Sharma (S1) Dr. Rajesh Gupta (S2)	Dr. Sunil Verma (S1) Dr. Rajesh Gupta (S2)
Nephrology	Dr. Kara B (S1) Dr. Shruti P (S2)	Dr. Sunita B (S1) Dr. Sameer Yadav (S2)	Dr. Sunita B (S1) Dr. Shruti P (S2)	Dr. Kara B (S1) Dr. Sameer Yadav (S2)	Dr. Kara B (S1) Dr. Shruti P (S2)	Dr. Kara B (S1) Dr. Sameer Yadav (S2)	Dr. Sunita B (S1) Dr. Shruti P (S2)
Neurology	Dr. Raghu M (S1) Dr. Ashish K (S2)	Dr. Ramesh Yadav (S1) Dr. Anil Kumar (S2)	Dr. Ramesh Yadav (S1) Dr. Anil Kumar (S2)	Dr. Raghu M (S1) Dr. Anil Kumar (S2)	Dr. Raghu M (S1) Dr. Anil Kumar (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. Kavita H (S2)	Dr. Surya K (S1) Dr. Anita Kumar (S2)	Dr. Surya K (S1) Dr. Rohan S (S2)	Dr. Suresh S (S1) Dr. Kavita H (S2)	Dr. Suresh S (S1) Dr. Rohan S (S2)	Dr. Suresh S (S1) Dr. Rohan S (S2)	Dr. Surya K (S1) Dr. Kavita H (S2)
Endocrinology	Dr. Shankar G (S1) Dr. Meena R (S2)	Dr. Priyanka P (S1) Dr. Mridula A (S2)	Dr. Priyanka P (S1) Dr. Shaam D (S2)	Dr. Shankar G (S1) Dr. Meena R (S2)	Dr. Shankar G (S1) Dr. Mridula A (S2)	Dr. Shankar G (S1) Dr. Shaam D (S2)	Dr. Priyanka P (S1) Dr. Meena R (S2)

Schedule for Nurses

Schedule Table							
Department	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Cardiology	Aarti Gupta (S1) Aarti Gupta (S2)	Priya Singh (S1) Preeti Y (S2)	Priya Singh (S1) Rajesh Sharma (S2)	Aarti Gupta (S1) Meera Kapoor (S2)	Meera Kapoor (S1) Priya Singh (S2)	Aarti Gupta (S1) Rajesh Sharma (S2)	Priya Singh (S1) Meera Kapoor (S2)
Nephrology	Harish M (S1) Suresh Verma (S2)	Deepika M (S1) Sanjay G (S2)	Deepika M (S1) Gita Yadav (S2)	Harish M (S1) Suresh Verma (S2)	Harish M (S1) Sanjay G (S2)	Harish M (S1) Gita Yadav (S2)	Deepika M (S1) Suresh Verma (S2)
Neurology	Sonia Singh (S1) Sarita Sharma (S2)	Rajesh Kumar (S1) Anjali S (S2)	Rajesh Kumar (S1) Anil Verma (S2)	Sonia Singh (S1) Sarita Sharma (S2)	Sonia Singh (S1) Anil Verma (S2)	Sonia Singh (S1) Anil Verma (S2)	Rajesh Kumar (S1) Sarita Sharma (S2)
Orthopedics	Anil Kumar (S1) Meena S (S2)	Sunita Verma (S1) Kartik L (S2)	Sunita Verma (S1) Ram Yadav (S2)	Anil Kumar (S1) Meena S (S2)	Anil Kumar (S1) Kartik L (S2)	Anil Kumar (S1) Ram Yadav (S2)	Sunita Verma (S1) Meena S (S2)
Oncology	Aparna Sharma (S1) Samyukta Jain (S2)	Dhruv M (S1) Hariini D (S2)	Dhruv M (S1) Rohan J (S2)	Aparna Sharma (S1) Samyukta Jain (S2)	Aparna Sharma (S1) Hariini D (S2)	Aparna Sharma (S1) Rohan J (S2)	Dhruv M (S1) Samyukta Jain (S2)
Endocrinology	Akshaya J (S1) Varsha Nair (S2)	Sangeeta P (S1) Divya Sharma (S2)	Sangeeta P (S1) Pooja S (S2)	Akshaya J (S1) Varsha Nair (S2)	Akshaya J (S1) Divya Sharma (S2)	Akshaya J (S1) Pooja S (S2)	Sangeeta P (S1) Varsha Nair (S2)

7. CONCLUSIONS AND FUTURE DIRECTIONS

Challenges Faced

- Scaling the CSP algorithm to handle large-scale scheduling problems while maintaining efficiency and computational resources proved challenging. Optimizing the algorithm to manage numerous variables and constraints was essential.
- Balancing conflicting scheduling constraints, such as preferred shifts and fair workload distribution among staff, required sophisticated constraint handling mechanisms
- Incorporating and organizing diverse data sources within the MongoDB database architecture presented challenges in ensuring seamless data integration and consistency.

What Went Right

- Algorithm Optimization: Overcoming initial complexities, the scheduling algorithm efficiently allocated staff, considering individual preferences and designated days off, leading to fair and efficient scheduling.
- Database Scalability: The implementation of MongoDB proved successful in managing scheduling data and staff information, providing scalability and easy access to pertinent data.
- User Interface Enhancement: Iterative design improvements resulted in user-friendly interfaces, enabling seamless interaction and improved communication among staff members.

What Went Wrong

- Initial Algorithm Implementation: The initial stages of implementing the scheduling algorithm faced complexities that required substantial refinement and testing.
- Data Consistency: Ensuring consistent and updated data across various modules within the system required additional attention and refinement.
- Scope Management: Managing scope creep and ensuring alignment with initial

project objectives proved challenging at times.

Lessons Learned

- Embracing an iterative approach allowed for constant improvements and adaptations, especially in complex algorithmic implementation.
- Prioritizing user feedback and iteratively refining interfaces significantly enhanced user satisfaction and system usability.
- Implementing robust testing protocols at every stage helped identify and rectify issues early in the development cycle.
- Ensuring consistent and clear communication among team members and stakeholders played a vital role in project success.

Future Directions

- Advanced Algorithm Refinements: Continual refinement and integration of more advanced algorithmic techniques to further optimize staff allocation and scheduling.
- Enhanced Data Analytics: Expanding analytics tools for more comprehensive performance evaluation and predictive analytics.
- Telemedicine Integration: Exploring integration possibilities for telemedicine functionalities to broaden the system's capabilities.
- Mobile Application Development: Consideration for a mobile application interface to increase accessibility for staff on-the-go.

8. COURSES OF IMPROVEMENT

Advanced Algorithmic Techniques: Further studies in advanced algorithms beyond CSP, like machine learning algorithms or optimization techniques, could enhance the scheduling algorithm's capabilities. Courses in algorithms, artificial intelligence, or optimization could be beneficial.

Emergency Response Protocols and Healthcare Management: Emergency response protocols in healthcare settings or healthcare management could provide insights into optimizing emergency response systems within the hospital.

Health Informatics: Understanding the intersection of healthcare and information technology through health informatics courses can provide specialized knowledge useful in developing healthcare-related software systems.

9. REFERENCES

1. <https://github.com/nccvector/genetic-algorithm-path-planning/blob/master/main.py>
2. <https://algorithm-visualizer.org/>
3. <https://www.freecodecamp.org/news/path-finding-algorithm-visualizer-tutorial/>
4. <https://iopscience.iop.org/article/10.1088/1742-6596/2246/1/012081/pdf>
5. <https://www.emerald.com/insight/content/doi/10.1108/01445150910972921/full/html?skipTracking=true>

Client Evaluation Report

Rating System - 1: Strongly disagree 2: Disagree 3: Neutral 4: Agree 5: Strongly Agree

Questions	1	2	3	4	5
The problem was well discussed and, the requirements and goals were clear.					
The project plan was well defined and communicated from the start.					
The resources were adequate for achieving the goals.					
The original timeline was realistic and was followed.					
The teamwork was well demonstrated.					
The client was communicated on regular intervals and given updates on the progress of the project.					
The expected project requirements have been satisfied.					