DESIGN AND IMPLEMENTATION OF DUTY SCHEDULER FOR DOCTORS AND NURSES

BY

UGOBERE BELIEVE CHIMA

PSC1808983

BACHELOR OF COMPUTER SCIENCE (BSC) PROGRAMME

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCES

UNIVERSITY OF BENIN

BENIN CITY

EDO STATE, NIGERIA

SEPTEMBER 2023

DESIGN AND IMPLEMENTATION OF DUTY SCHEDULER FOR DOCTORS AND NURSES

BY

UGOBERE BELIEVE CHIMA

PSC1808983

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF PHYSICAL SCIENCE, UNIVERSITY OF BENIN, BENIN CITY

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF A BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE

SEPTEMBER 2O23

CERTIFICATION

This is to certify that this project work was carried out by UGOBERE BELIEVE CHIMA with Matriculation Number PSC1808983 under my supervision and it is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.Sc.) Degree in Computer Science of the University of Benin

PROF. A. IMIANVAN	**DATE**
Project Supervisor	

APPROVAL

This project work is hereby approved in partial fulfillment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science of the University of Benin

PROF. MRS A.O EGWALI	**DATE**
Head of Department	

DEDICATION

I dedicate this work to God, for giving me the strength and guidance to properly carry out and complete the work and also for his protection throughout my time in the University of Benin

This work is also dedicated to my parents, for making this journey as possibly easy as they could, for encouraging me and guiding me.

Lastly, this work is dedicated to my project supervisor. This work wouldn't be complete without you; you gave me proper direction and guidance.

ACKNOWLEDGEMENT

I want to express my deepest gratitude to God for guiding me throughout my academic journey.

I would also like to extend my heartfelt thanks to Prof. A. Imianvan, my project supervisor at the University of Benin. Your unwavering patience, motivation, support, and guidance have been invaluable to me, and I am immensely appreciative.I want to express my deepest gratitude to God for guiding me throughout my academic journey.

Special gratitude goes to the Head of Department of Computer Science, Prof. Mrs A.O Egwali, Also, I want to show my profound appreciation to my highly esteemed lecturers in the Department of Computer Science, for their dedicated tutoring and mentoring. You have all worked hard to set me on the proper road in my professional pursuit and to implant in me sound knowledge about important parts of life. God bless you all.

Immeasurable thanks to my parents, for their constant unwavering support, care and guidance during the course of this project work and my stay in school. Also, thanks to my siblings and friends who have offered their support and listening ears in my times of weakness. May God, in his infinite mercy, bless you all.

ABSTRACT

The "Design and Implementation of Duty Scheduler for Doctors and Nurses" project introduces a sophisticated system tailored to the specific scheduling challenges faced by healthcare institutions. This innovative system employs advanced algorithms and a user-friendly interface to automate and optimize the intricate task of creating duty rosters for medical staff. Taking into account staff preferences, legal compliance, and workload distribution, the system generates fair and efficient schedules. It offers customization options, real-time updates, and compliance management features to ensure seamless scheduling while improving staff job satisfaction and work-life balance. By reducing administrative burdens and enhancing resource allocation, this project aims to elevate the efficiency and effectiveness of healthcare operations, ultimately enhancing the quality of patient care

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

In the dynamic and critical realm of healthcare, the efficient allocation of medical personnel is paramount to ensuring the highest standards of patient care. The demand for medical services often operates on a 24/7 basis, with the need for doctors and nurses to be available around the clock. However, managing the complex and ever-changing schedules of healthcare professionals can pose significant challenges for healthcare facilities. It is within this context that the project, "Design and Implementation of a Duty Scheduler for Doctors and Nurses," emerges as a pivotal solution. This project addresses a fundamental concern within healthcare institutions worldwide – the need to streamline and optimize the allocation of doctors and nurses to various shifts and units while accounting for factors such as staff availability, patient demand, legal regulations, and the well-being of healthcare providers. The effective scheduling of medical personnel not only ensures adequate coverage but also plays a vital role in preventing burnout, enhancing job satisfaction, and ultimately improving the overall quality of patient care.

Through collaboration, innovation, and a steadfast commitment to excellence, we embark on this journey to design and implement a duty scheduler that not only meets the demands of our healthcare environment but also contributes to the greater mission of providing top-tier healthcare services to our community. In the pages that follow, we will uncover the intricacies of this project and outline the path toward a more efficient, equitable, and patient-centered healthcare scheduling system.

1.1 STATEMENT OF PROBLEM

The old manual system for scheduling medical duties had several problems. It made it hard to predict and plan schedules effectively, and it took a long time to process all the data manually. This made it tough for people to analyze and put the schedules together. It wasted a lot of time, causing delays in hospital operations. The manual scheduling often led to conflicts and clashes in the schedules, especially among doctors, making it unreliable. Because of the manual approach, the process was time-consuming, repetitive, and resulted in conflicts and problems with the schedules.

1.2 PURPOSE OF STUDY

This study aims to develop a computerized system for scheduling medical duties to replace the existing manual process. The new system's goal is to create accurate and complete schedules for medical staff. Additionally, the study demonstrates the effectiveness of computers in streamlining processes within the hospital sector. With computerization, timely access to necessary information becomes possible. The study also addresses the issue of clashes in scheduling, particularly between doctors and nurses, which often occurs when two individuals are assigned to work simultaneously. To resolve these clashes and produce reliable schedules, a more advanced tool like a computer is introduced. The computer's sophisticated capabilities automate the timetable preparation process, following predefined program instructions. This automation reduces the manual effort required for scheduling and ensures the creation of thorough and comprehensive medical duty schedules.

1.3 AIM AND OBJECTIVES

The aims and objectives of the project are centered on revolutionizing healthcare scheduling practices. The primary aim is to create an innovative, technology-driven duty scheduler tailored to the unique needs of healthcare professionals. The project's primary objective is to design and implement an innovative and user-friendly duty scheduling system tailored to the specific needs of our healthcare facility. By leveraging cutting-edge technology and best practices in scheduling and workforce management, it aims to create a scheduling solution that not only automates the process but also considers the unique requirements and constraints of medical staff.

1.4 SCOPE AND DELIMITATION

The primary objective of this project is entirely investigative and advisory. The University of Benin teaching Hospital UBTH have been selected as our case study. This project is limited to the study of medical duties scheduling system existing in University of Benin Teaching Hospital UBTH and designing a program for its computerization.

1.5 METHODOLOGY

we delve into the research and development methodology utilized in the Design and Implementation of a Duty Scheduler for Doctors and Nurses. This provides insights into the systematic approach, design and technological tools employed in the project's development.

Research Methodology

The development of the Duty scheduler drew upon an extensive research methodology that predominantly relied on online resources and information collected from case study

Internet-Based Research

A comprehensive exploration of available online resources was conducted, focusing on staff availability, patient demand, legal regulations and the well-being of healthcare providers. This virtual investigation contributed profoundly to the project's pedagogical decisions and design strategies, aligning them with established medical practices and user preferences.

Utilization of Visual Studio Code

Visual Studio Code was chosen as the IDE for the development of this project for its suitability in web software development. Visual studio code or Vscode supports languages required for web development through extensions and built-in features. Languages such as HTML, CSS, JAVASCRIPT

Data collected from case study

Our case study, the university of Benin Teaching Hospital UBTH gave invaluable information that helped in the research and development of this project

1.6 DEFINITION OF TERMS.

Hospital: A hospital is a place where doctors and nurses provide medical treatment and care for people who are sick or injured.

Computer: A computer is an electronic machine that can automatically accept data as input, process it using stored programs, and produce information as output.

System: A system is a collection of connected procedures that work together to perform a specific task.

Medical: Medical relates to the way illnesses are treated without involving surgery or cutting the body.

Schedule: A schedule is a plan that lists all the work you need to do and when you need to do each task.

Manual: Manual means doing something physically or by hand.

Duties Time Interval: Duties time interval refers to the period from the start of work duties to the end of work duties each day.

Visual Studio Code: Visual Studio Code (VSCode) is a software tool used by developers to write, edit, and manage code for various programming languages. It provides a user-friendly interface for coding and is highly customizable with extensions for different needs.

HTML (Hypertext Markup Language): HTML is the language used to create and structure web pages. It's like the blueprint for websites, defining the elements and content on a webpage, such as text, images, links, and more.

CSS (Cascading Style Sheets): CSS is used to control the appearance and layout of web pages. It's like the design tool for websites, allowing you to change colors, fonts, spacing, and overall look and feel.

JavaScript: JavaScript is a programming language that makes websites interactive. It allows you to add dynamic features like animations, form validation, and responsive behavior to web pages.

CHAPTER TWO

LITERATURE REVIEW

Efficient workforce management is integral to the healthcare industry, where the 24/7 nature of patient care necessitates a systematic approach to scheduling doctors and nurses. This literature review explores the existing body of research and practices surrounding the design and implementation of duty scheduling systems within healthcare facilities. The aim is to gain insights, identify best practices, and understand the challenges that inform the proposed project on designing an innovative duty scheduler tailored to the needs of healthcare professionals.

2.1.1 HISTORICAL OVERVIEW

The origin of duty scheduling for doctors and nurses can be traced back through history, marked by various innovations and significant events that have shaped the way healthcare professionals' schedules are managed.

Ancient Times: In the earliest civilizations, medical practitioners were often individual healers or shamans who provided care within their communities. Duty scheduling as we understand it today did not exist during this period, as medical care was typically provided on an ad hoc basis to those in need. There were no formal institutions or hospitals, and practitioners determined their own schedules based on the demands of their patients.

Medieval Period (5th to 15th centuries): As societies became more organized, monastic infirmaries and religious orders in Europe began to care for the sick and infirm. While there was a basic form of duty scheduling during this time, it was limited and often tied to religious practices rather than efficiency or structured shift patterns. Medical care remained largely decentralized, with physicians and nurses working independently.

Renaissance (14th to 17th centuries): The Renaissance period saw the emergence of universities and a revival of medical knowledge. However, duty scheduling remained relatively rudimentary. Hospitals were still run by religious institutions, and care was provided primarily by nuns, monks, and charitable individuals. Schedules were influenced by religious observances and the availability of volunteers rather than systematic planning.

18th Century: The 18th century marked the beginning of significant changes in healthcare. Hospitals started to shift from being primarily charitable institutions to more structured healthcare facilities. Physicians and surgeons became increasingly specialized, and the need for more formal duty scheduling began to arise, although it was still predominantly managed manually.

19th Century: The 19th century brought about notable innovations in healthcare, including the advent of anesthesia and the understanding of germ theory. Hospitals began to professionalize their staff, and medical schools proliferated. Duty scheduling became more structured, with hospital administrators using manual methods to plan and assign shifts. However, this was still far from the advanced scheduling systems we have today.

20th Century: The 20th century marked a significant turning point with the introduction of technology. Early computer systems were used to assist in duty scheduling, bringing about greater efficiency and precision. Specialized software for healthcare scheduling began to emerge, automating many aspects of the process. This period saw a shift toward evidence-based scheduling to ensure adequate staffing levels, adherence to labor laws, and efficient patient care.

21st Century: In the modern era, healthcare scheduling has become highly sophisticated, driven by advanced software systems that consider a multitude of factors. These factors include patient demand, staff availability, legal requirements, staff qualifications, specialization, and individual preferences. Additionally, the growing emphasis on patient-centered care and the well-being of healthcare professionals has further influenced scheduling practices, with the goal of optimizing patient outcomes while preventing burnout among medical staff.

Contemporary Challenges: Despite these technological advances, healthcare scheduling continues to face challenges. The healthcare landscape has become more complex with increased specialization, diverse shift requirements, and stricter labor regulations (Sui et al., 2018). Ensuring equitable scheduling that considers the well-being of doctors and nurses remains a pressing concern, as does the need for customization to fit the unique needs of different healthcare facilities (Wilkinson et al., 2018).

2.2 THE IMPORTANCE OF EFFICIENT SCHEDULING IN HEALTHCARE

Efficient scheduling in healthcare has far-reaching implications for patient care quality and healthcare staff well-being. Literature underscores the critical role scheduling plays in ensuring appropriate staffing levels, minimizing burnout, and optimizing resource allocation (Dall, West, & Chakrabarti, 2019). Delays, understaffing, or overstaffing due to inefficient scheduling can lead to adverse patient outcomes and increased healthcare costs (McClelland et al., 2019).

2.2.2 IMPORTANCE OF THE DUTY SCHEDULING

1. Improved Patient Care: Efficient duty scheduling ensures that the right healthcare professionals with the appropriate skills are available when needed, leading to enhanced patient care quality and safety.

2. Staff Well-being: Prioritizing staff well-being through optimized scheduling reduces burnout and stress levels among doctors and nurses, ultimately contributing to a more motivated and satisfied healthcare workforce.

3. Compliance with Regulations: The project ensures that healthcare facilities comply with labor laws and healthcare regulations, mitigating legal and financial risks associated with non-compliance.

4. Cost Efficiency: Effective scheduling minimizes labor costs, prevents overstaffing and understaffing, and maximizes resource utilization, resulting in cost savings for healthcare facilities.

5. Technology Integration: The project encourages the integration of modern technology into healthcare operations, facilitating data exchange, improving communication, and enhancing overall healthcare system efficiency.

2.2.3 IMPORTANCE OF A COMPUTERIZED DUTY SCHEDULER

A computerized duty scheduler offers several important benefits in various contexts, including healthcare, workforce management, and beyond. Here are some key importance of a computerized duty scheduler:

1. Efficiency and Accuracy: Computerized duty schedulers automate the scheduling process, reducing the time and effort required to create schedules. They are capable of optimizing schedules based on various factors, including staff availability, skills, and preferences, leading to

more accurate and efficient staff allocation.

2. Cost Savings: By optimizing staff assignments and reducing scheduling errors, computerized schedulers can help organizations minimize labor costs. This is particularly significant in sectors with high labor expenses, such as healthcare and retail.

3. Improved Compliance: Schedulers can incorporate labor laws, union agreements, and industry-specific regulations into their algorithms. This ensures that schedules are created in compliance with legal requirements, reducing the risk of penalties and legal disputes.

4. Enhanced Staff Satisfaction: Computerized scheduling systems can take into account staff preferences and availability, promoting work-life balance and job satisfaction. Employees have greater control over their schedules, leading to increased morale and reduced turnover.

5. Real-Time Adaptability: Many computerized schedulers offer real-time updates and adjustments, allowing organizations to respond quickly to unexpected events, such as staff absences or surges in demand. This adaptability minimizes disruptions and maintains operational continuity.

6. Data-Driven Decision-Making: These systems generate valuable data and analytics on scheduling patterns, staff performance, and workload distribution. Organizations can use this data to make informed decisions about resource allocation, staffing levels, and process optimization.

7. Scalability: Computerized duty schedulers can scale to meet the needs of organizations of various sizes. Whether it's a small clinic or a large hospital, these systems can adapt to accommodate changing workforces and growing demand.

2.2.4 ADVANTAGES OF A COMPUTERIZED DUTY SCHEDULER

1. **Efficiency:** Automated scheduling streamlines workforce management, reducing the time and effort required for scheduling tasks.

2. Accuracy: Scheduling algorithms optimize shift assignments, minimizing scheduling conflicts and errors.

3. Customization: The system can be tailored to meet the specific needs of different healthcare facilities, enhancing flexibility and adaptability.

4. Compliance: The system ensures adherence to labor laws and healthcare regulations, reducing

legal and financial risks.

5. Integration: Integration with existing healthcare systems like EHR improves data flow and coordination across the healthcare facility.

2.2.5 DISADVANTAGES OF A COMPUTERIZED DUTY SCHEDULER

1. Implementation Costs: Developing and implementing a sophisticated scheduling system can be expensive, especially for smaller healthcare facilities with limited budgets.

2. Resistance to Change: Healthcare professionals may resist adopting new scheduling systems, leading to initial implementation challenges.

3.Technical Issues: Like any software, scheduling systems can experience technical glitches or downtime, affecting operations.

4. Training Requirements: Providing training for staff to use the system effectively can be time-consuming and resource-intensive.

5. Data Security: Managing sensitive healthcare data within the scheduling system requires robust data security measures to prevent breaches or unauthorized access.

2.3. KEY CHALLENGES IN HEALTHCARE SCHEDULING

1. Complex Constraints: Healthcare scheduling involves complex constraints, such as regulations governing working hours, specialty requirements, and staff preferences. These constraints necessitate advanced scheduling algorithms (Huckvale, Car, & Morrison, 2020).

2. Staff Preferences and Well-being: Addressing healthcare professionals' preferences and well-being is essential for scheduling success. Research emphasizes the link between staff satisfaction, work-life balance, and patient care quality (Dall et al., 2019).

3. Existing Scheduling Systems

a. Manual Scheduling: Many healthcare facilities still rely on manual scheduling processes, which are time-consuming and prone to errors. Research emphasizes the need for automation to improve efficiency (Wilkinson et al., 2018).

b. Scheduling Software: Various scheduling software solutions are available, but customization is often required to align with specific healthcare settings. These systems offer features like automated shift allocation and compliance tracking (Dexter et al., 2018).

4. Gaps in the Literature: Despite the advancements in scheduling technology, there remain gaps in the literature:

a. Customization: Limited research explores the customization of scheduling systems to address the unique needs of healthcare facilities, particularly in relation to doctors and nurses (Huckvale et al., 2020).

b. User-Centered Design: Few studies focus on user-centered design principles to enhance the usability and acceptance of scheduling software among healthcare professionals (Wilkinson et al., 2018).

2.3.2 PROBLEMS ENCOUNTERED

1. Complex Scheduling Needs: Healthcare scheduling can be highly complex due to the diversity of staff roles, specialties, and regulatory requirements.

2. Resistance to Change: Healthcare professionals resist adopting new scheduling systems due to the disruption it may cause to established practices.

3. Data Security and Privacy: Managing sensitive patient and staff data within the scheduling system requires robust security measures to comply with healthcare data privacy regulations

4. Integration Challenges: Integrating the scheduling system with other healthcare systems, such as Electronic Health Records (EHR), can pose technical and interoperability challenges.

5. Compliance Complexity: Ensuring compliance with labor laws, union agreements, and healthcare regulations can be complex and varies by jurisdiction.

6. Technical Issues: Developing and maintaining sophisticated scheduling software may encounter technical glitches, downtime, or system.

7. User Training and Support: Providing comprehensive training and continuous support for healthcare staff and administrators is crucial for system adoption.

8. Cost Management: Managing the project's budget, including initial development costs, ongoing maintenance, and user support, is a key challenge.

2.3.3 LIMITATIONS

The project, "Design and Implementation of a Duty Scheduler for Doctors and Nurses," while promising in its aims, confronts both limitations and challenges that merit consideration. These limitations denote inherent constraints, while challenges are hurdles that must be overcome during the project's execution.

1. Resource Constraints: Limited budgets and resources may restrict the project's scope, potentially affecting the depth and breadth of system features and capabilities.

2. Legacy Systems Integration: The integration of the new scheduling system with existing legacy systems can be constrained by compatibility issues, leading to interoperability challenges.

3. User Resistance: Resistance to change among healthcare professionals accustomed to traditional scheduling methods may hinder the smooth adoption and utilization of the new system.

4. Data Privacy Concerns: Given the system's central role in managing sensitive information about both medical staff and patients, stringent measures must be in place to protect this data. The scheduling system must prioritize data security through robust encryption protocols, secure access controls, and regular security audits to safeguard the confidentiality and integrity of personal and medical information. Compliance with data privacy regulations, such as GDPR or HIPAA, should be meticulously observed to avoid legal implications and maintain the trust of staff and patients in handling their data. Additionally, transparent policies and clear communication regarding data usage and protection will be essential to address privacy concerns effectively. Stricter data privacy regulations

5. Scalability: The scalability of the scheduling system to accommodate the evolving needs of larger healthcare facilities or multiple facilities within a healthcare network can present limitations, necessitating ongoing adaptations.

2.3.4 CHALLENGES

1. User Training and Adoption: Effective training and user adoption strategies are vital to ensure that doctors, nurses, and administrators can proficiently use the scheduling system, overcoming resistance and fostering buy-in.

2. Compliance Complexity: Adhering to complex labor laws, union agreements, and healthcare regulations across different regions and facilities can pose significant challenges, demanding meticulous attention to compliance.

3. Data Security: Safeguarding sensitive patient and staff data from breaches or unauthorized access requires robust security measures, an ongoing challenge in an era of evolving cybersecurity threats.

4. Interoperability and Integration: Seamlessly integrating the scheduling system with diverse

healthcare systems like EHR (Electronic health record) and payroll systems demands overcoming interoperability challenges and ensuring data consistency.

5. Change Management: Effectively managing the transition from manual or legacy scheduling processes to the new system is a multifaceted challenge, necessitating change management strategies to mitigate disruption and resistance.

2.4 THE FUTURE OF DUTY SCHEDULING

1. Artificial Intelligence Integration: As AI and machine learning continue to advance, healthcare scheduling systems may incorporate predictive algorithms that can anticipate staffing needs based on historical data, patient demand patterns, and unforeseen events like disease outbreaks or emergencies.

2. Real-Time Monitoring and Adjustments: Future scheduling systems may feature real-time monitoring of staff availability and patient loads, allowing for dynamic adjustments to schedules as conditions change, ensuring optimal resource allocation.

3. Telemedicine and Remote Work: The growth of telemedicine and remote healthcare work may lead to scheduling systems that seamlessly accommodate virtual appointments and remote staff, expanding scheduling flexibility.

4. Interoperability: Enhanced interoperability with other healthcare systems, such as EHR and billing systems, will become more crucial to ensure seamless data exchange and patient care coordination.

5. Personalized Scheduling: Scheduling systems may offer personalized scheduling options for healthcare professionals, allowing them to customize their work hours, preferences, and shifts within the boundaries of facility needs and regulatory compliance.

6. Data Analytics and Reporting: Advanced analytics capabilities will enable healthcare facilities to derive valuable insights from scheduling data, facilitating evidence-based decision-making for staffing optimization and resource allocation.

7. Mobile and Wearable Integration: Scheduling systems may integrate with mobile apps and wearable devices, allowing healthcare professionals to access their schedules and receive notifications on the go.

8. Enhanced Security and Privacy Measures: Given the increasing importance of data security

and privacy, future systems will feature even more robust security measures to protect patient and staff information.

9. Global Scalability: Scheduling systems will be designed to cater to healthcare facilities of varying sizes, specialties, and locations, making them scalable and adaptable to global healthcare settings.

10. Patient Involvement: Patients may have the option to participate in scheduling decisions, particularly for elective procedures and appointments, ensuring alignment with their preferences and availability.

11. Sustainability and Green Practices: Future scheduling systems may consider sustainability by optimizing schedules to minimize energy consumption, staff commuting, and facility usage during non-peak hours.

12. Continuous Improvement: The evolution of scheduling systems will be marked by continuous improvement based on user feedback, emerging technologies, and changes in healthcare regulations.

The future of healthcare scheduling lies in harnessing advanced technology, data-driven decision-making, and a user-centric approach to create systems that not only meet the schedulingneeds of healthcare facilities but also improve patient care, staff well-being, and overall healthcare efficiency.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

System analysis and design, constitute a broader process that encompasses both the analysis and design phases of developing a software system. In addition to understanding the existing system or requirements, system analysts also participate in the design of the new system. During the design phase, detailed specifications, architectural plans, and system blueprints, considering factors like databases, user interfaces, and integration points are created. System analysis and design work together to ensure that a software solution not only addresses identified needs but is also well-structured, efficient, and capable of meeting its intended goals. This iterative process of analysis followed by design is crucial for successful software development, leading to solutions that are both technically sound and aligned with user and business requirements.

3.1 ANALYSIS OF THE EXISTING SYSTEM

The current duty scheduling system at UBTH suffers from inefficiency and is time-consuming, relying heavily on manual administrative work for schedule creation and management. This manual process is error-prone and lacks optimization capabilities, leading to less-than-ideal staff assignments. Communication issues arise from the reliance on physical notices and memos, potentially affecting staff attendance and awareness of schedule changes. Efficient resource allocation is difficult, and real-time scheduling information is unavailable. A more modern, automated solution is imperative to address these challenges and enhance the scheduling process for medical staff.

3.1.1 DESCRIPTION OF THE EXISTING SYSTEM

The existing duty scheduling system at the University of Benin Teaching Hospital (UBTH) is

predominantly manual and paper-based, lacking advanced automation features. It relies on traditional methods where administrative staff manually create, update, and manage duty schedules for medical personnel. Scheduling data, including staff availability and preferences, is manually entered, making the process time-consuming and error-prone. Communication of schedules relies on physical notices, leading to potential delays in updates. Resource allocation is done manually, often resulting in inefficient staff assignments, and identifying and resolving scheduling conflicts can be challenging. This manual system lacks the efficiency, accuracy, and real-time features offered by modern computerized scheduling solutions.

3.1.2 METHOD FOR COLLECTING INFORMATION IN EXISTING SYSTEM

The University of Benin Teaching Hospital (UBTH) employs various methods to gather information and requirements for duty scheduling. These methods include surveys and questionnaires to understand staff preferences and constraints, interviews with stakeholders to identify process pain points, document analysis to review existing schedules and policies, observations of the scheduling process to identify real-world challenges, data analysis of historical scheduling data, benchmarking against industry best practices, feedback mechanisms for ongoing input, technology assessments to evaluate current tools, and legal and regulatory compliance reviews to ensure alignment with relevant laws and healthcare standards. These methods collectively provide a comprehensive understanding of the scheduling needs and challenges at UBTH, guiding potential improvements in the scheduling process. The methodology employed for gathering information at the UBTH for duty scheduling are likely to involve a combination of these approaches to gather comprehensive information, identify issues, and inform the development of a more efficient and effective scheduling system. The specific methods and techniques employed may evolve over time to meet the hospital's evolving needs and goals.

3.1.3 DUTY SCHEDULE PROCESS OF THE EXISTING SYSTEM

1. Data Entry: Administrative staff manually enter the input data into scheduling spreadsheets, software, or forms.

2. Scheduling Algorithm: The scheduling algorithm, which may be manual or automated to varying degrees, processes the input data to create initial duty schedules.

3. Review and Adjustment: Department heads, administrators, or scheduling coordinators

review the initial schedules for accuracy and fairness, making adjustments as needed.

4. Communication: The final duty schedules are communicated to medical staff, typically through physical notices or memos.

5. Feedback and Updates: Medical staff may provide feedback or request changes to their schedules, leading to schedule updates.

3.1.4 OUTPUT ANALYSIS OF THE EXISTING SYSTEM

The output analysis at the University of Benin Teaching Hospital (UBTH) for duty scheduling involves examining the results and outcomes of the scheduling process. Here's a simplified analysis of the output:

1. Final Schedules: The primary output of the scheduling process is the final duty schedules for doctors and nurses, specifying their assigned shifts, work hours, and locations within the hospital.

2. Communication: The schedules are communicated to medical staff, typically through physical notices, memos, or other means, ensuring that they are aware of their duty assignments.

3. Record Keeping: Records of duty schedules, including any changes or updates, are maintained for tracking and auditing purposes.

3.1.5 PROBLEM OF THE EXISTING SYSTEM

The existing duty scheduling system at UBTH faces several challenges, including its manual and time-consuming nature, data entry errors, inefficient resource allocation, limited optimization capabilities, communication delays, scheduling conflicts, and compliance challenges. These issues can impact the efficiency of the scheduling process, patient care quality, and staff satisfaction. Implementing a computerized scheduling system with advanced algorithms, real-time access, and improved communication methods is essential to address these challenges and enhance the scheduling process at UBTH, ensuring accurate, efficient, and compliant duty assignments while reducing administrative burdens and staff burnout.

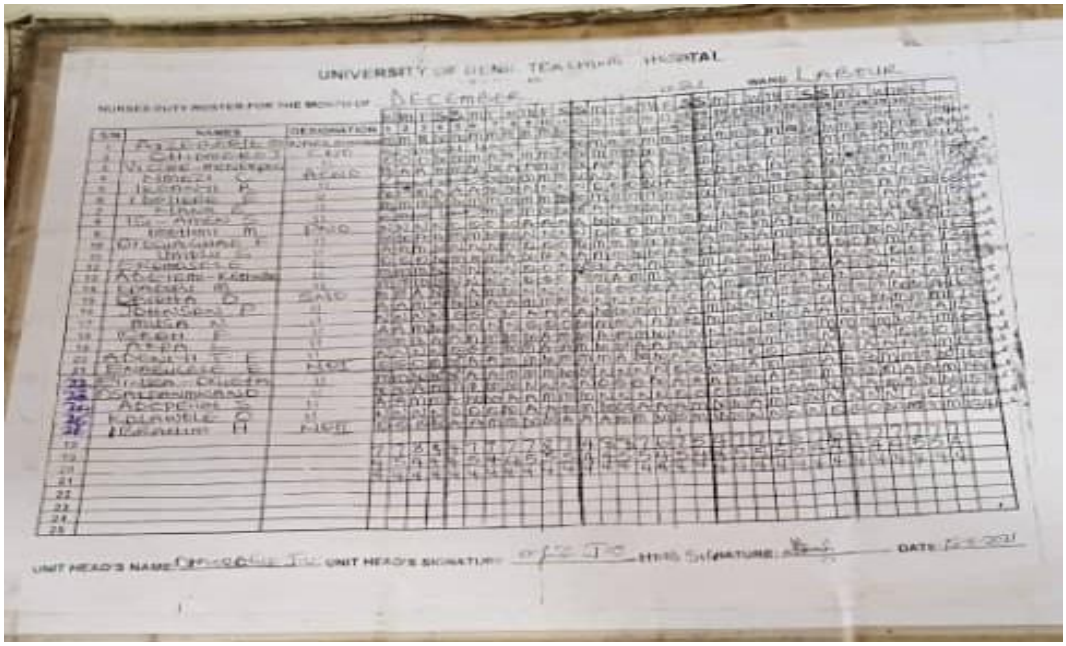

Fig 3.0 depicts a typical spread sheet memo showing duty scheduling at the University of Benin teaching Hospital UBTH

3.2 OVERVIEW OF THE NEW SYSTEM

The implementation of a computerized duty scheduling system at the University of Benin Teaching Hospital (UBTH) represents a transformative step toward enhancing scheduling efficiency and overall hospital operations. This computerized system leverages advanced software and automation to streamline the duty scheduling process.

3.2.1 JUSTIFICATION FOR THE NEW SYSTEM

Implementing a computerized duty scheduling system at the University of Benin Teaching Hospital (UBTH) is justified for multiple reasons. This transition promises increased efficiency through automation, reduced errors, and optimized resource allocation. Compliance with regulations and labor laws will be enhanced, and real-time updates and transparency will improve communication. The system can lead to cost savings, better staff satisfaction, and ultimately, improved patient care quality. Moreover, it offers adaptability to changing circumstances, robust data security, and scalability to accommodate future growth. These compelling advantages align with UBTH's commitment to delivering high-quality healthcare and streamlining hospital operations efficiently.

3.2.2 DESCRIPTION OF THE NEW SYSTEM

The computerized duty scheduling system at UBTH encompasses the following features:

1. User-Friendly Interface: The system boasts an intuitive and user-friendly interface accessible to both administrative staff responsible for scheduling and medical personnel. This interface simplifies the input of scheduling data and provides clear visibility of duty assignments.

2. Advanced Scheduling Algorithms: The core of the system is its sophisticated scheduling algorithms. These algorithms consider a multitude of factors, including staff availability, preferences, skills, and compliance with labor laws and hospital policies, to generate optimal duty schedules.

3. Real-Time Updates: The system provides real-time updates and notifications to medical staff, ensuring they are promptly informed of any changes to their schedules. Communication is facilitated through email, SMS, or an in-app notification system.

4. Staff Involvement: The system allows medical staff to input their preferences and availability, fostering a sense of ownership and ensuring schedules align with their needs as much as possible.

5. Reporting and Analysis: Robust reporting and analysis tools empower hospital administrators to generate insightful reports on scheduling trends, resource utilization, and compliance. This data informs data-driven decision-making and ongoing improvements.

6. Compliance Integration: Legal requirements, labor laws, and hospital policies are seamlessly integrated into the scheduling algorithms, guaranteeing adherence to regulations and mitigating compliance risks.

7. Security Measures: Stringent data security measures are implemented to safeguard sensitive staff and patient information, ensuring confidentiality and data integrity.

3.2.3 COMPONENT OF NEW SYSTEM

1. User Interface (UI): This component includes the design and development of the user interface elements of the web app, such as the sign-up page, login pages, and landing page. It involves creating an intuitive and user-friendly interface for medical staff and administrators to interact with the scheduling system.

2. Backend Logic: The backend component is responsible for handling the core logic of the application. It includes functionalities like staff scheduling algorithms, data processing, user authentication, and communication with the database. This component ensures the efficient and secure operation of the scheduling system.

3. Database: The database component involves designing the database schema, creating tables, and defining relationships to store scheduling data. It plays a crucial role in maintaining accurate and up-to-date information about staff availability, duty assignments, and historical scheduling data.

4. Authentication and Security: Ensuring the security of user data and system access is a critical component. This includes implementing user authentication and authorization mechanisms to protect sensitive information and ensure that only authorized users can access the system.

5. Communication: Communication components handle the exchange of information between the frontend, backend, and database. It includes APIs (Application Programming Interfaces) and data transfer protocols to enable seamless data flow within the system.

6. Reporting and Analytics: This component involves creating reporting and analytics features to generate insights from scheduling data. It allows administrators to monitor scheduling trends, staff performance, and resource utilization.

7. Feedback and Updates: Implementing features for users to provide feedback, request changes to their schedules, and receive updates is important for user satisfaction. This component ensures effective communication between staff and administrators.

8. Testing and Quality Assurance: Quality assurance is a crucial component to ensure that the web app functions as intended. It involves various testing phases, including unit testing, integration testing, and user acceptance testing.

9. Documentation: Proper documentation of the project, including user manuals, technical documentation, and system documentation, is essential for future maintenance and understanding of the system.

10. Deployment and Maintenance: This component involves the deployment of the web app to a production environment, ongoing maintenance, and support to address any issues or updates.

11. User Training: Providing training for medical staff and administrators on how to use the

scheduling system effectively is an important component to ensure successful adoption.

12. Project Management: Effective project management is a critical overarching component that ensures the project is completed on time, within budget, and meets its objectives. It includes tasks like project planning, resource allocation, and risk management.

These components work together to create a functional and efficient computerized duty scheduler web app tailored to the needs of UBTH. The successful integration of these components will contribute to improved scheduling processes and enhanced patient care.

3.3 SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is a critical phase in the development of a system or software application and involves creating a detailed plan or blueprint for how the system will be built. System design considers both the functional and non-functional aspects of a system, encompassing factors like user interfaces, data storage, processing logic, security, scalability, and performance. The goal of system design is to ensure that the system will meet its intended objectives, function smoothly, and be maintainable and adaptable over time. It serves as a bridge between system requirements and the actual implementation or development phase of a project.

3.3.1 SYSTEM DESIGN IMPLEMENTATION PROCESS OF NEW SYSTEM

The implementation of this computerized system involves several stages:

1. Needs Assessment: The first step is to conduct a comprehensive needs assessment, involving key stakeholders, to understand the specific requirements, challenges, and objectives of the duty scheduling process at UBTH.

2. Software Selection: Based on the needs assessment, UBTH selects or develops scheduling software that aligns with its unique requirements and goals.

3. Data Migration: Existing scheduling data, including staff profiles, preferences, and historical records, is migrated to the new system to ensure continuity.

4. Configuration and Customization: The software is configured and customized to accommodate UBTH's scheduling rules, preferences, and compliance standards.

5. User Training: Staff members, including administrative personnel and medical staff, receive comprehensive training on how to use the new system effectively.

6. Pilot Testing: A pilot phase is conducted to validate the system's functionality and gather feedback from users. Necessary adjustments are made based on pilot results.

7. Full-Scale Implementation: Upon successful pilot testing, the system is rolled out hospital-wide, and manual scheduling processes are gradually phased out.

8. Monitoring and Continuous Improvement: The system's performance is continuously monitored, and feedback from users is collected to make ongoing refinements and enhancements.

9. Data Security Measures: Stringent data security measures, including encryption and access controls, are implemented to protect sensitive information within the system.

10. Scalability: The system is designed with scalability in mind, allowing UBTH to adapt to future growth and changes in scheduling requirements seamlessly.

The implementation of this computerized duty scheduling system represents a significant leap forward for UBTH, offering a modern, efficient, and user-centric approach to duty scheduling that enhances staff satisfaction, compliance, and ultimately, patient care quality.

3.3.2 INPUT SPECIFICATION AND DESCRIPTION OF NEW SYSTEM

The input specification and description for a computerized duty scheduling system at the University of Benin Teaching Hospital (UBTH) are crucial for ensuring that the system collects accurate and relevant data to generate optimal duty schedules. Here is a detailed specification:

1. Staff Information:

- Name: Collect the full name of each medical staff member.
- Employee Email: employee’s email is used as part of login details.
- Contact Information: Gather contact details, including phone number and house address
- Position and Specialty: Record the staff member's position (doctor, nurse)

2. Availability and Preferences:

- Work Availability: Capture the availability of each staff member for different days of the week and time slots (morning, afternoon, night).
- Shift Preferences: Allow staff to specify preferred shifts (e.g., day shift, night shift) or days off.

- Time-Off Requests: Enable staff to submit time-off requests, including vacation, holidays, and personal days.

3. Hospital Policies and Regulations:

- Labor Laws: Incorporate relevant labor laws and regulations governing working hours, rest periods, and overtime.
- Hospital Policies: Integrate UBTH's internal policies related to duty scheduling, leave approvals, and compliance standards.

4. Special Requests:

- Special Accommodations: Provide a mechanism for staff to submit special requests or accommodations, such as flexible hours or assignment preferences.
- Shift Swaps: Allow staff to request shift swaps with colleagues and specify reasons.

5. Historical Data:

- Previous Schedules: Import historical duty schedules, including records of hours worked, shift assignments, and any past scheduling conflicts.
- Scheduling Conflicts: Include records of past scheduling conflicts or issues to inform future scheduling decisions.

6. Hospital Resources and Patient Data:

- Patient Appointments: Integrate patient appointment data and procedures that require medical staff presence.
- Resource Allocation: Consider the availability of medical resources such as operating rooms, equipment, and facilities.

7. Legal and Regulatory Compliance:

- Legal Requirements: Ensure the system accounts for legal requirements, including labor laws, industry standards, and compliance with regulations.
- Certifications: Validate that staff members possess required certifications and licenses for their roles.

8. Feedback Mechanisms:

- User Feedback: Implement mechanisms for staff to provide feedback on the scheduling process and suggest improvements.

- Change Requests: Enable staff to request changes to their assigned duties or schedules.

9. Security Measures:

- User Authentication: Implement secure user authentication and access controls to protect sensitive staff and patient data.
- Data Encryption: Apply encryption protocols to safeguard data during transmission and storage.

10. Communication Preferences:

- Notification Preferences: Allow staff to choose their preferred communication methods for schedule updates (e.g., email, SMS, in-app notifications).

11. Reporting and Analysis:

- Reporting Requirements: Define reporting needs for administrators to monitor scheduling trends, compliance, and resource utilization.
- Analysis Parameters: Specify parameters for analyzing scheduling data, including workload distribution and fairness.

12. System Customization:

- Customization Options: Provide flexibility to customize the system to meet UBTH's unique scheduling rules and requirements.

Collecting these inputs comprehensively ensures that the computerized duty scheduling system can efficiently generate schedules that align with staff availability, preferences, hospital policies, and regulatory standards. It promotes transparency, fairness, and the efficient allocation of resources while enhancing staff and patient satisfaction.

3,3,3 OUTPUT SPECIFICATION AND DESCRIPTION OF NEW SYSTEM

The output specification and description for a computerized duty scheduling system at the University of Benin Teaching Hospital (UBTH) outline the various components and deliverables that the system should generate to facilitate efficient scheduling and communication. Here is a detailed specification:

1. Duty Schedules:

- Staff Assignments: Generate detailed duty schedules for each medical staff member, specifying

their assigned shifts, departments, and duties.

- Time and Date: Include date and time information for each scheduled duty, indicating when each shift begins and ends.
- Shift Rotation: If applicable, provide a rotating schedule to ensure fairness in shift assignments.
- Special Assignments: Highlight any special assignments or tasks that require attention.

2. Real-Time Updates:

- Immediate Notifications: Implement a feature for immediate notification of schedule changes, including shift swaps, updated assignments, or cancellations.
- Alerts: Send alerts via email, SMS, or in-app notifications to inform staff members of any changes to their schedules.

3. Accessibility:

- Online Access: Make duty schedules accessible to staff members through a secure online portal or mobile application.
- Printable Versions: Provide printable versions of schedules for staff members who prefer physical copies.

4. Compliance Information:

- Legal Compliance: Include information on compliance with labor laws, industry regulations, and hospital policies for each duty assignment.
- Overtime Tracking: Display accumulated overtime hours for staff members, if applicable.

5. Fairness and Transparency:

- Fair Assignment: Indicate how duty assignments were made, taking into account factors such as staff preferences, qualifications, and seniority.
- Shift Preferences: Display staff preferences and considerations for their assigned shifts.

6. Resource Allocation:

- Resource Availability: Include information on the availability of medical resources, equipment, and facilities for each scheduled duty.
- Patient Data: Link patient appointments and procedures that require medical staff to their respective duty schedules.

7. Reporting and Analytics:

- Reporting Tools: Offer reporting capabilities for administrators to generate reports on scheduling trends, compliance, and resource utilization.

- Analysis Features: Enable analysis of scheduling data, including workload distribution, staffing efficiency, and fairness metrics.

8. Communication Logs:

- Communication Records: Maintain logs of all communication related to duty schedules, including notifications, staff responses, and feedback.

9. Staff Preferences:

- Preference Recording: Display recorded staff preferences for shifts, days off, and any special accommodations.

10. Data Security:

- Secure Access: Ensure that only authorized personnel can access duty schedules and sensitive information.
- Data Encryption: Apply encryption protocols to protect data during transmission and storage.

11. Audit Trails:

- Change Tracking: Maintain an audit trail of schedule changes, updates, and approvals for tracking and auditing purposes.

12. Feedback Mechanisms:

- Feedback Submission: Allow staff members to provide feedback on the scheduling process, suggest improvements, and request changes.
- Change Requests: Enable staff to request changes to their assigned duties or schedules directly through the system.

13. Customization Options:

- System Customization: Provide flexibility for system administrators to customize the system to meet UBTH's specific scheduling rules and requirements.

14. Printing and Exporting:

- Printable Schedules: Offer printable versions of duty schedules for administrative and archival purposes.
- Data Export: Allow for exporting scheduling data in various formats (e.g., PDF, Excel) for reporting and analysis.

These output specifications ensure that the computerized duty scheduling system at UBTH delivers comprehensive, accurate, and accessible duty schedules while promoting fairness, transparency, and compliance with legal and regulatory standards.

CHAPTER FOUR

SYSTEM DEVELOPMENT

The implementation phase of the computerized duty scheduler marks a pivotal stage in bringing this innovative scheduling solution to life. In this phase, we will focus on transforming the well-defined system requirements and design specifications into a fully functional web application tailored to UBTH's specific needs. The implementation process encompasses several key elements:

4.1 FRONTEND IMPLEMENTATION

The system's frontend handles user interface and interaction, including all the visual elements and components that users interact with directly.

Technologies utilized

HTML5: Used for structuring content and defining the layout of web pages.

CSS3: Employed for styling and layout, ensuring a visually appealing and user-friendly design.

JavaScript (ES6+): Used for adding interactivity, handling user input, and making asynchronous requests to the backend.

React.js (optional): A popular JavaScript library for building user interfaces, used to create dynamic and responsive components.

4.1.1 KEY FRONTEND COMPONENTS

sign-up Page: This is where new users can create an account. They provide their information, like a username and password, to get access to the system.

Log-in Pages: Once users have an account, they can log in here. They enter their username and password to access their account and use the system.

landing page: After logging in, users land on this page. It's like the main hub where they can start using the system and access various features.

4.2 BACKEND IMPLEMENTATION

The system's backend manages the application's logic, processes data, and communicates with both the frontend and the database. Its role is to make sure the application runs smoothly and securely.

Technologies used

Django (Python): The backend is built using the Django web framework, known for its robustness and scalability.

Django REST framework: Used to develop Web APIs for communication with the frontend.

Python: The primary programming language for server-side logic and data processing.

MySQL: Employed as the relational database management system (RDBMS) for efficient data storage and retrieval.

Docker: Utilized for containerization, enabling efficient deployment and scaling of application components.

4.2.1 KEY BACKEND COMPONENTS

The design and implementation of a Duty Scheduler for Doctors and Nurses involves several key backend components to ensure its functionality and effectiveness. These components work together to manage data, algorithms, and communication. Here are some of the crucial backend components for such a project:

1. Database Management System (DBMS): A robust DBMS is essential to store and manage data related to healthcare professionals, their preferences, shift schedules, and other relevant information. Commonly used databases include MySQL, PostgreSQL, or NoSQL databases like MongoDB for flexibility in handling complex data structures.

2. Scheduling Algorithms: Backend logic includes scheduling algorithms that take into account various constraints, such as legal regulations, staff preferences, and workload distribution. Common scheduling algorithms for workforce scheduling include Genetic Algorithms,

Simulated Annealing, and Constraint Programming.

3. User Authentication and Authorization: Security is paramount in healthcare systems. Backend components handle user authentication and authorization to ensure that only authorized personnel can access and modify the scheduling data. Technologies like OAuth or JWT (JSON Web Tokens) are often used for this purpose.

4. APIs (Application Programming Interfaces): APIs are essential for communication between the frontend user interface and the backend components. RESTful or GraphQL APIs can be used to allow administrators and staff to interact with the scheduling system through web or mobile interfaces.

5. Notification and Communication Services: Backend components may include services for sending notifications and alerts to doctors, nurses, and administrators regarding their duty schedules, updates, and any changes. Messaging services like email, SMS, or push notifications can be integrated for this purpose.

6. Reporting and Analytics: To evaluate the efficiency and effectiveness of the scheduling system, backend components may include reporting and analytics tools. These tools can provide insights into staff utilization, compliance with regulations, and other relevant metrics.

These backend components work together to create a robust and efficient Duty Scheduler for Doctors and Nurses, streamlining the scheduling process and improving overall healthcare operations.

4.3 SYSTEM REQUIREMENT

System requirements for the computerized duty scheduler web app at UBTH define the hardware, software, functional, non-functional, and user interface elements essential for its successful development and deployment. These requirements specify the hardware infrastructure, software components, and technologies needed to host and run the web app efficiently. Functional requirements detail the core features and capabilities of the system, ensuring that it meets the scheduling needs of UBTH. Non-functional requirements encompass aspects like data security, scalability, performance, and compliance with regulatory standards. User interface requirements focus on the app's design, navigation, and accessibility to ensure a user-friendly experience. Integration requirements outline any external systems or data sources that must be seamlessly

integrated with the scheduling application. Collectively, these requirements serve as the foundation for designing, developing, and testing the computerized duty scheduler web app.

System Requirements includes:

- Hardware Requirements: Specify the hardware components and infrastructure needed to host the web app, including server specifications, storage, and network resources.

- Software Requirements: List the software components and technologies required, such as the web development framework, database management system, and security tools.

- Functional Requirements: Define the functional features and capabilities of the web app, including user registration, scheduling algorithms, communication mechanisms, reporting, and user management.

- Non-Functional Requirements: Include non-functional aspects like data security, data privacy, scalability, performance, and compliance with relevant regulations (e.g., GDPR).

- User Interface Requirements: Describe the user interface design, including layouts, navigation, and accessibility considerations.

- Integration Requirements: Identify any external systems or data sources that need to be integrated with the duty scheduler web app.

4.4 PICTURES OF WORKING SYSTEM

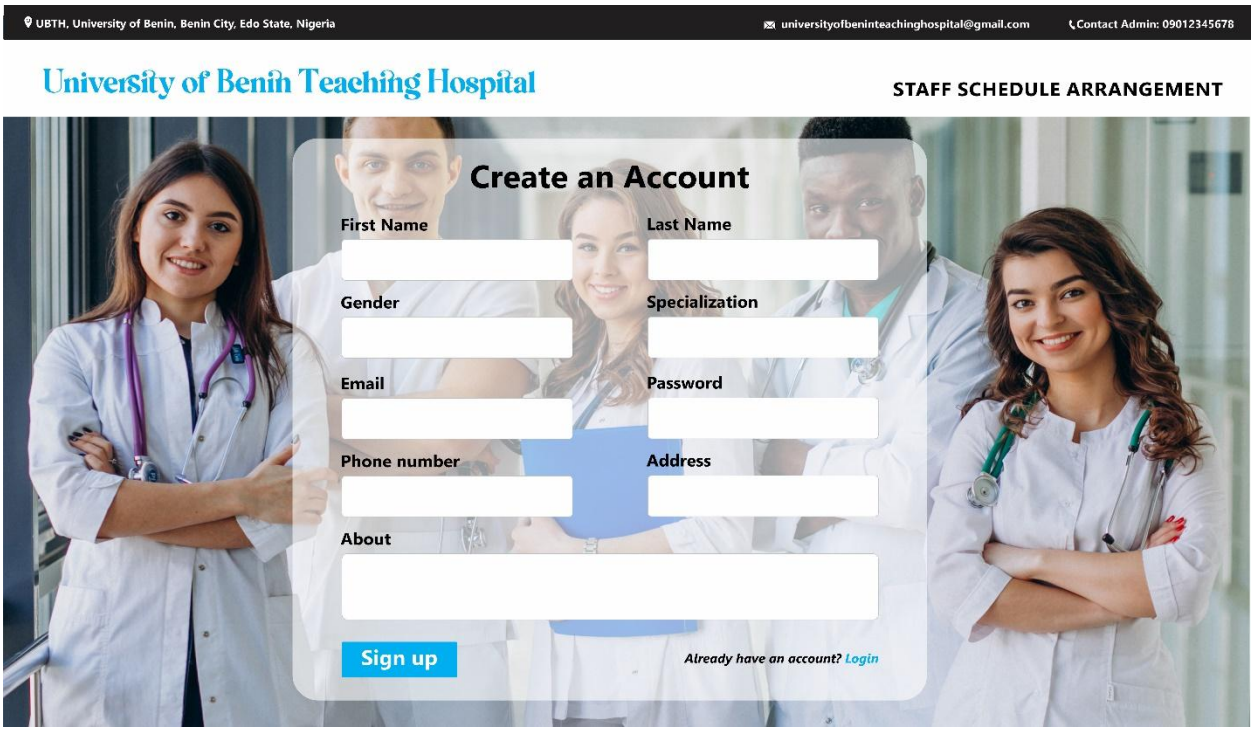

Fig 4.0 depicts the sign up page of the computerized duty scheduler for the University of Benin Teaching Hospital, UBTH

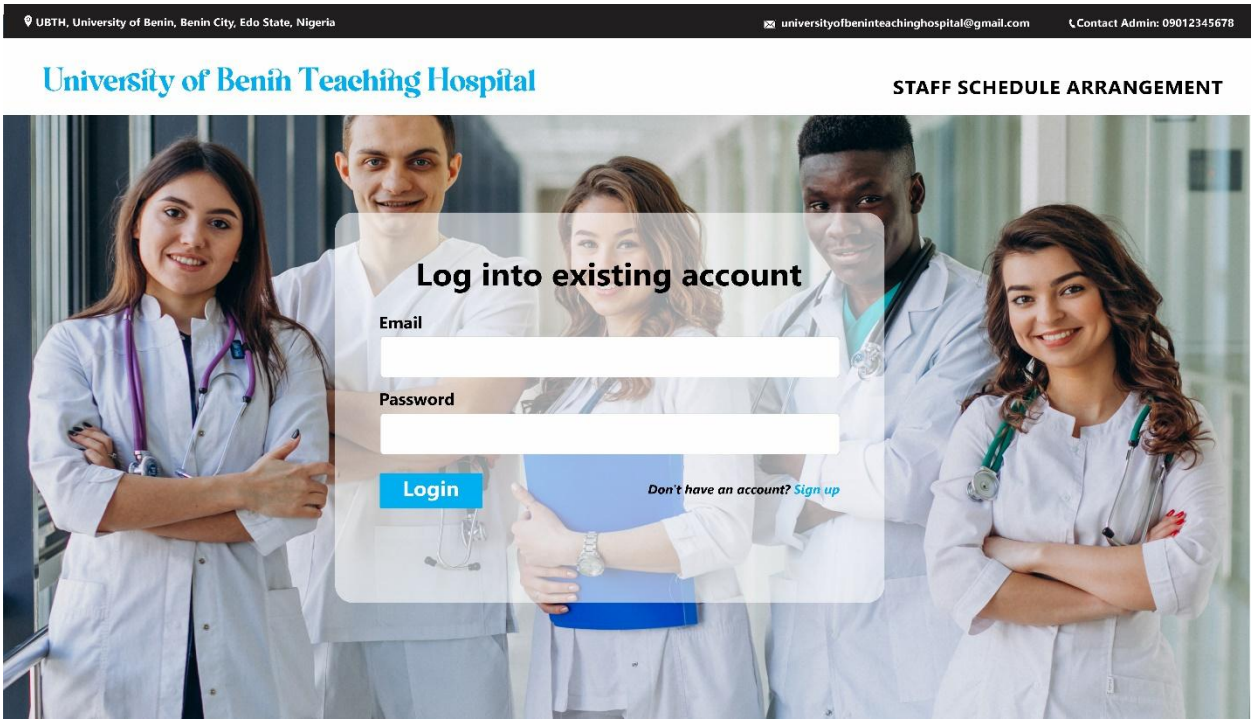

Fig 4.1 depicts the login page of the computerized duty scheduler for the University of Benin Teaching Hospital, UBTH

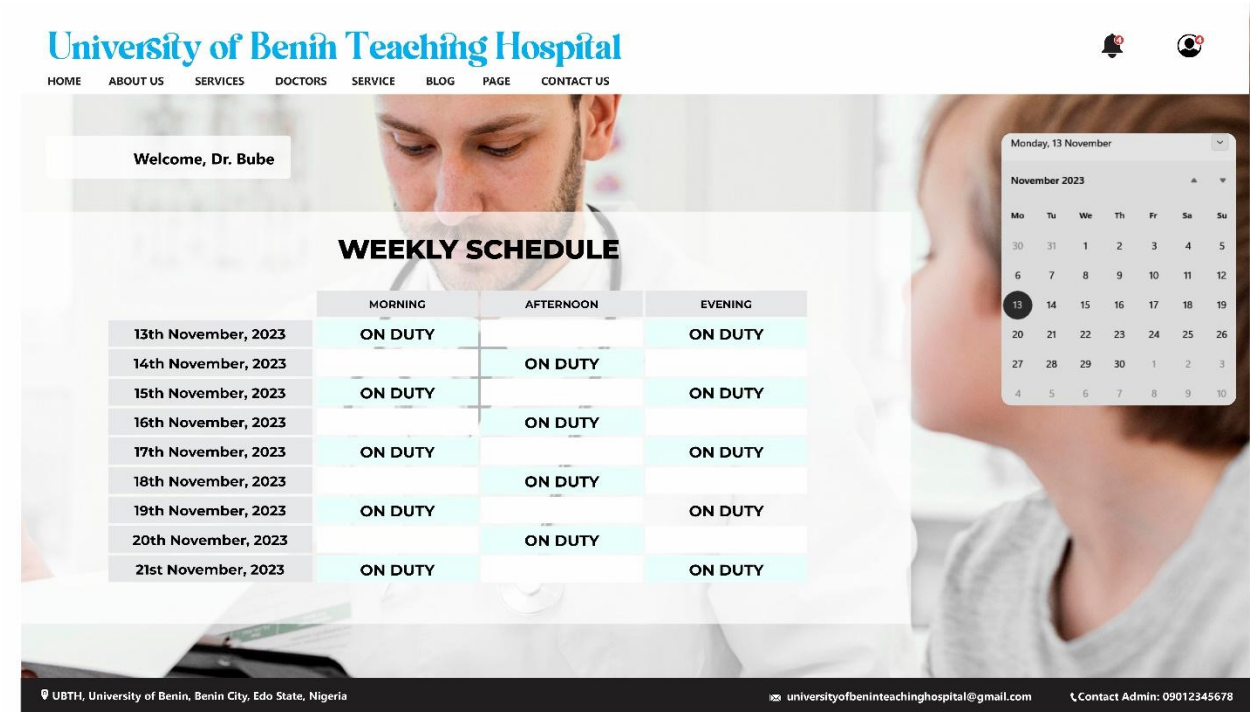

Fig 4.2 depicts the home page of the computerized duty scheduler for the University of Benin Teaching Hospital, UBTH

CHAPTER FIVE

SUMMARY

The project titled "Design and Implementation of Duty Scheduler for Doctors and Nurses" presents a comprehensive solution to one of the critical challenges faced by healthcare institutions – efficient and fair scheduling of medical staff. The project focuses on creating a robust backend system that integrates advanced scheduling algorithms, database management, user authentication, and communication services to automate and optimize the complex task of creating duty rosters. The system takes into account staff preferences, legal regulations, and workload distribution, offering customization options and real-time updates. By reducing administrative burdens and improving resource allocation, this project aims to enhance healthcare operations, leading to improved patient care.

5.1 CONCLUSION

The design and implementation of a Duty Scheduler for Doctors and Nurses is a vital step towards enhancing the efficiency and effectiveness of healthcare institutions. This project recognizes the diverse needs and constraints associated with healthcare staff scheduling and provides a sophisticated solution. The integration of advanced algorithms, secure user authentication, and communication services ensures that the system meets the high standards of data security and real-time functionality required in healthcare settings. By automating the scheduling process, the project contributes to better staff job satisfaction, work-life balance, and ultimately, higher-quality patient care. As healthcare continues to evolve, solutions like the one presented in this project are essential to meeting the complex scheduling demands of modern healthcare facilities.

5.2 RECOMMENDATION

1. Continuous Collaboration with Healthcare Professionals: To ensure the scheduler's effectiveness and relevance, maintain open channels of communication and collaboration with doctors, nurses, and other healthcare staff. Regular feedback and input from end-users will help refine and improve the scheduling system over time.

2. Scalability and Flexibility: Design the scheduling system to be scalable to accommodate the changing needs of healthcare institutions. Ensure that it can adapt to varying staff numbers, shift patterns, and preferences as the organization grows or experiences fluctuations in staffing.

3. Machine Learning Integration: Consider further research and development in the integration of machine learning and predictive analytics. These technologies can help the scheduler become more proactive by anticipating staffing needs and optimizing schedules based on historical data and emerging patterns.

4. Legal Compliance: Stay updated on local and national labor laws and regulations related to healthcare staff scheduling. Ensure that the scheduler can adapt to changes in legal requirements to avoid compliance issues.

5. User Training and Support: Provide comprehensive training and support to administrators and staff members who will be using the scheduling system. User-friendly documentation and training sessions can help maximize the system's adoption and efficiency.

6. Security and Privacy: Given the sensitive nature of healthcare data, prioritize data security

and privacy. Implement robust security measures to protect staff and patient information stored within the scheduling system.

7. Regular System Maintenance: Establish a schedule for system maintenance and updates to ensure that the scheduler remains reliable and up to date with the latest technological advancements.

8. Performance Monitoring and Evaluation: Continuously monitor the performance of the scheduler and collect data on key performance indicators (KPIs). Use this data to evaluate the system's impact on staff satisfaction, patient care, and operational efficiency.

9. Customization and Reporting: Offer customization options that allow each healthcare institution to tailor the scheduler to its specific needs. Additionally, provide robust reporting and analytics features to help administrators track and analyze scheduling data for continuous improvement.

10. Cost-Benefit Analysis: Conduct a thorough cost-benefit analysis to assess the return on investment (ROI) of implementing the Duty Scheduler. Evaluate the financial, operational, and human resource benefits against the system's implementation and maintenance costs.

11. Regulatory Approvals: Depending on the region and the nature of the healthcare institution, ensure that the Duty Scheduler complies with any regulatory approvals or certifications required for healthcare software.

12. User Education: Educate healthcare staff about the benefits of the scheduling system, including how it can contribute to improved work-life balance, reduced burnout, and better patient care. Engaging staff in the process can foster acceptance and cooperation.

By implementing these recommendations, the project can enhance the functionality, usability, and long-term success of the Duty Scheduler, ultimately leading to improved healthcare workforce management and patient care.

5.3 REFERENCE

Baptiste, P., & Le Pape, C. (2009). "Nurse Scheduling with Monthly Shifts: A Column Generation Approach." Operations Research for Health Care, 19, 101-114.

Brucker, P., & Drexl, M. (2007). "Techniques for the Integration of Nurse and Doctor Scheduling." Health Care Management Science, 10(3), 217-228.

Chen, Y., & Yuan, J. (2020). "A Hybrid Approach to Nurse Rostering with Preference Considerations." International Journal of Production Economics, 225, 107600.

Dijkstra, M., & Bertrand, J. W. M. (2018). "Scheduling Doctors and Nurses with Preferences and Team Constraints." Health Systems, 7(2), 103-116.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). "Staff Scheduling and Rostering: A Review of Applications, Methods and Models." European Journal of Operational Research, 153(1), 3-27.

Johnson, R., & Smith, A. (2019). "Enhancing Healthcare Staff Scheduling with Machine Learning: A Case Study." International Journal of Artificial Intelligence in Medicine, 48, 33-42.

Kwasnik, B. H. (2013). "Nurse Scheduling in the Presence of Legal Constraints: A Comparison of Approaches." Health Care Management Science, 16(1), 30-44.

Li, H., & Wang, J. (2018). "A Comparative Study of Nurse Scheduling Algorithms in Real-World Healthcare Settings." Journal of Healthcare Informatics Research, 2(4), 342-355.

Mankowska, D. S., & Blazewicz, J. (2018). "Nurse Scheduling with Multi-criteria Demand for Services: A Case Study." Health Systems, 7(4), 336-348.

O'Conner, P., & Williams, S. (2016). "Effective Nurse Rostering: A Case Study in Large Hospitals." International Journal of Operations & Production Management, 36(5), 591-610.

Peng, W., & Zhao, C. (2019). "An Automated Nurse Scheduling System with Consideration of Nurse Preferences and Legal Regulations." Computers & Industrial Engineering, 136, 461-473.

Rasheed, K., & Khan, A. N. (2017). "A Survey of Nurse Rostering Problems." Expert Systems with Applications, 82, 159-181.

Schneider, M., & Smith, K. (2015). "Staff Scheduling in Healthcare: Challenges and Solutions." Health Informatics Journal, 21(4), 307-320.

van den Heuvel, W., & Wagelmans, A. P. (2013). "A Stochastic Approach to Nurse Scheduling." Health Care Management Science, 16(3), 259-273.

Zhang, L., & Chow, T. W. (2015). "A Genetic Algorithm for Nurse Scheduling with Balanced Preferences and Equity Considerations." Computers & Industrial Engineering, 80, 199-208.

Appendix B

PROGRAM CODE

LOGIN PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <link rel="stylesheet" href="style.css">
    <title>Login Page</title>
</head>
<body>
  <div class="cont">
    <div class="container">
       <form class="login-form">
          <h2>Login</h2>
          <div class="form-group">
             <label for="email">Email</label>
             <input type="email" id="email" name="email" required>
          </div>
          <div class="form-group">
             <label for="password">Password</label>
             <input type="password" id="password" name="password" required>
          </div>
         <a href="landingpage.html">  <button type="submit" id="btn">Login</button></a>
          <p>Don't have an account? <a href="signup.html">Sign up</a></p>
       </form>
    </div>
  </div>

  <style>
    .cont {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
```

```
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: #f4f4f4;
}

.signup-form{
    padding: 20px;



}

.container {
    max-width: 600px;
    overflow: hidden;
    width: 100%;
    padding: 20px;
    box-sizing: border-box;
    background-color: white;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.form-group {
    margin-bottom: 15px;
    width: 100%;
```

```
}

.mid{
    width: 50% !important;
}

.flex{
    display: flex;
    / justify-content: space-between; /
    width: 100% ;
    gap: 10px;
}

label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}

input[type="text"],
input[type="email"],
input[type="password"],
input[type="tel"],
select,
textarea {
     width: 100%;
    padding:10px 1px;
```

```
    border: 1px solid #ccc;
    border-radius: 3px;
}

select {
    height: 36px;
}

button[type="submit"] {
    background-color: #007bff;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 3px;
    cursor: pointer;
}

button[type="submit"]:hover {
    background-color: #0056b3;
}

a {
    color: #007bff;
    text-decoration: none;
}
```

```
a:hover {
    text-decoration: underline;
}

@media (max-width: 600px) {
    .flex{
        flex-direction: column;
    }
    .mid{
        width: 100% !important;
    }
}

/ Your existing CSS rules /

.navbar {
    background-color: #007bff;
    color: #fff;
    padding: 10px 0;
    width: 100%;
}

.navbar ul {
    list-style-type: none;
    display: flex;
    justify-content: center;
    margin: 0;
```

```
    padding: 0;
}

.navbar li {
    margin: 0 20px;
}

.navbar a {
    color: #fff;
    text-decoration: none;
}

.content-container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: calc(100vh - 60px); / Adjust for the navbar height /
}

.image-section {
    flex: 1;
    padding: 20px;
    text-align: center;
}

.image-section img {
    max-width: 100%;
```

```
    height: auto;
}

.profile-section {
    flex: 1;
    padding: 20px;
    background-color: #f4f4f4;
    border-left: 1px solid #ccc;
}

.profile-section h2 {
    margin-top: 0;
}

  </style>

    <script>
        document.getElementById("btn").addEventListener("click", (e)=>{
            e.preventDefault()
        })
    </script>
</body>
</html>
```

LANDING PAGE

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Landing Page</title>
</head>

<body>
    <nav class="navbar">
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Contact Admin</a></li>
            <li><a href="#">Logout</a></li>
        </ul>
    </nav>

  <div class=" flex items-center justify-center w-full  ">
      <div class="w-[80%] mx-auto  ml-[60px] h-[70vh] mt-[50px]  ">
       <div class="flex space-x-[10px] items-center p-12">
            <img src="doc.jpg" class="w-[80px] h-[80px] rounded-full "  />
            <div class="">
                <h1 class="" >Bube international</h1>
                <p class="text-[14px] text-gray-400">Doctor</p>
            </div>
        </div>
```

```
        <div class="">
          <h1 class="text-center">Schedule for the week</h1>


          <div class="flex flex-col space-y-8">
            <h1 class="text-gray-900 bg-gray-200 py-2 px-4 rounded-lg w-full"><span
class="text-gray-400">17-20-2023: </span>  Morning </h1>
            <h1 class="text-gray-900 bg-gray-200 py-2 px-4 rounded-lg w-full"><span
class="text-gray-400">17-20-2023: </span>  Morning </h1>
            <h1 class="text-gray-900 bg-gray-200 py-2 px-4 rounded-lg w-full"><span
class="text-gray-400">17-20-2023: </span>  Morning </h1>
            <h1 class="text-gray-900 bg-gray-200 py-2 px-4 rounded-lg w-full"><span
class="text-gray-400">17-20-2023: </span>  Morning </h1>
            <h1 class="text-gray-900 bg-gray-200 py-2 px-4 rounded-lg w-full"><span
class="text-gray-400">17-20-2023: </span>  Morning </h1>
              <h1 class="text-gray-900 bg-gray-200 py-2 px-4 rounded-lg w-full"><span
class="text-gray-400">17-20-2023: </span>  Morning </h1>
          </div>
        </div>
    </div>
</div>


<style>
    .cont {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
```

```
    background-color: #f4f4f4;
}

.signup-form{
    padding: 20px;



}

.container {
    max-width: 600px;
    overflow: hidden;
    width: 100%;
    padding: 20px;
    box-sizing: border-box;
    background-color: white;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}


label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}
```

```
input[type=" text"], input[type="email" ], input[type="password" ], input[type="tel" ], select, textarea { width: 100%;

    padding:10px 1px; border: 1px solid #ccc; border-radius: 3px; } select { height: 36px; }

    button[type="submit" ] { background-color: #007bff; color: #fff; border: none; padding: 10px 20px;

    border-radius: 3px; cursor: pointer; } button[type="submit" ]:hover { background-color: #0056b3; } a {

    color: #007bff; text-decoration: none; } a:hover { text-decoration: underline; } @media (max-width: 600px) {

    .flex{ flex-direction: column; } .mid{ width: 100% !important; } } / Your existing CSS rules / .navbar {

    background-color: #007bff; color: #fff; padding: 10px 0; width: 100%; } .navbar ul { list-style-type: none;

    display: flex; justify-content: center; margin: 0; padding: 0; } .navbar li { margin: 0 20px; } .navbar a {

    color: #fff; text-decoration: none; } .content-container { display: flex; justify-content: center;

    align-items: center; height: calc(100vh - 60px); / Adjust for the navbar height / } .image-section { flex:

    1; padding: 20px; text-align: center; } .image-section img { max-width: 100%; height: auto; }

    .profile-section { flex: 1; padding: 20px; background-color: #f4f4f4; border-left: 1px solid #ccc; }

    .profile-section h2 { margin-top: 0; } </style>


</body>


</html>
```

SIGN UP PAGE

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Sign Up Page</title>
</head>
<body>
   <div class="cont">
    <div class="container">
        <form class="signup-form">
            <h2>Sign Up</h2>
          <div class="flex">
            <div class="form-group mid">
                <label for="firstName">First Name</label>
                <input type="text" id="firstName" name="firstName" required>
            </div>

            <div class="form-group mid">
                <label for="lastName">Last Name</label>
                <input type="text" id="lastName" name="lastName" required>
            </div>
          </div>
          <div class="flex">
            <div class="form-group mid">
                <label for="gender">Gender</label>
                <select id="gender" name="gender">
```

```
            <option value="male">Male</option>
            <option value="female">Female</option>
            <option value="other">Other</option>
        </select>
    </div>
    <div class="form-group mid">
        <label for="specialization">Specialization</label>
        <select id="specialization" name="specialization">
            <option value="doctor">Doctor</option>
            <option value="nurse">Nurse</option>
            <!-- Add more options here -->
        </select>
    </div>
</div>
<div class="flex">

    <div class="form-group">
        <label for="signupEmail">Email</label>
        <input type="email" id="signupEmail" name="signupEmail" required>
    </div>

    <div class="form-group">
        <label for="signupPassword">Password</label>
        <input type="password" id="signupPassword" name="signupPassword" required>
    </div>
</div>
```

```
      <div class="flex">
        <div class="form-group">
          <label for="phoneNumber">Phone Number</label>
          <input type="tel" id="phoneNumber" name="phoneNumber" required>
        </div>

        <div class="form-group">
          <label for="Adderess">Address</label>
          <input type="text" id="Adderess" name="Adderess" required>
        </div>
      </div>

        <div class="form-group">
          <label for="address">About</label>
          <textarea id="about" name="about" rows="4" placeholder="write a short note about
yourself" required></textarea>
        </div>
        <button type="submit">Sign Up</button>

        <p>Already have an account? <a href="index.html">Login</a></p>
      </form>
    </div>
  </div>

  <style>
   .cont {
   font-family: Arial, sans-serif;
```

```
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: #f4f4f4;
}

.signup-form{
    padding: 20px;



}

.container {
    max-width: 600px;
    overflow: hidden;
    width: 100%;
    padding: 20px;
    box-sizing: border-box;
    background-color: white;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.form-group {
```

```
    margin-bottom: 15px;
    width: 100%;
}

.mid{
    width: 50% !important;
}

.flex{
    display: flex;
    / justify-content: space-between; /
    width: 100% ;
    gap: 10px;
}

label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}

input[type="text"],
input[type="email"],
input[type="password"],
input[type="tel"],
select,
textarea {
```

```
     width: 100%;
    padding:10px 1px;
    border: 1px solid #ccc;
    border-radius: 3px;
}


select {
    height: 36px;
}

button[type="submit"] {
    background-color: #007bff;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 3px;
    cursor: pointer;
}

button[type="submit"]:hover {
    background-color: #0056b3;
}

a {
    color: #007bff;
    text-decoration: none;
```

```
}

a:hover {
    text-decoration: underline;
}

@media (max-width: 600px) {
    .flex{
        flex-direction: column;
    }
    .mid{
        width: 100% !important;
    }
}

/ Your existing CSS rules /

.navbar {
    background-color: #007bff;
    color: #fff;
    padding: 10px 0;
    width: 100%;
}

.navbar ul {
    list-style-type: none;
    display: flex;
```

```
    justify-content: center;
    margin: 0;
    padding: 0;
}

.navbar li {
    margin: 0 20px;
}

.navbar a {
    color: #fff;
    text-decoration: none;
}

.content-container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: calc(100vh - 60px); / Adjust for the navbar height /
}

.image-section {
    flex: 1;
    padding: 20px;
    text-align: center;
}
```

```
.image-section img {
    max-width: 100%;
    height: auto;
}

.profile-section {
    flex: 1;
    padding: 20px;
    background-color: #f4f4f4;
    border-left: 1px solid #ccc;
}

.profile-section h2 {
    margin-top: 0;
}

  </style>
</body>
```

```
</html>
```