HealthSpace – A Donation Platform

Name: M.A.M. Perera

London Met ID No: 23041950

London Metropolitan University

Declaration

Module: CS6P05ES **Deadline:**

Module Leader: **Student ID: 23041950 / E220375**

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

(signature:)________________________ Date: 05.05.2024

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations* on Cheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples…

(i) Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.

(ii) Falsifying data in experimental results.

(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.

(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.

(v) Collusion to present joint work as the work solely of one individual.

(vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.

(vii) Other conduct calculated to secure an advantage on assessment.

(viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.

2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. "$e = mc^2$ (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Dedication

I would like to extend my heartfelt gratitude to everyone who has stood by me throughout the development of HealthSpace. Firstly, I owe immense appreciation to my family for their unwavering support and encouragement. They have been my rock, providing boundless love and motivation.

A special acknowledgment goes to my supervisor, whose guidance and mentorship have played a pivotal role in shaping the success of HealthSpace. Her expertise and dedication have challenged me to excel, pushing the boundaries of my capabilities in crafting a robust system tailored to the needs of the medical industry.

I am also indebted to my friends and colleagues for their continuous inspiration and motivation. Their invaluable feedback, ideas, and collaborative efforts have enriched my understanding of the project and facilitated its effective implementation.

Furthermore, I extend my gratitude to North Colombo Teaching Hospital for generously providing data relevant to the medical industry, despite the challenges involved in sharing donation-related information. Their flexibility and willingness to support an external project like HealthSpace are truly commendable.

Lastly, I am profoundly thankful to all those who have believed in me and fueled my passion for making a positive impact in the world. Their unwavering support has been a driving force behind HealthSpace. I sincerely hope that this project will contribute significantly to the medical industry and make all my supporters proud.

Acknowledgements

I would like to extend my sincere gratitude and appreciation to my project supervisor for her invaluable guidance, support, and encouragement throughout the development of my individual project on the HealthSpace – A donation platform. Her insightful feedback, constructive criticism, and unwavering commitment to mentoring me have been instrumental in shaping my work and improving its quality. I am grateful for her expertise and for sharing his vast knowledge in the field of Artificial Intelligence and Software Development.

I would also like to acknowledge the efforts and contributions of all the other lecturers who have played a pivotal role in my learning experience during this course. Their lectures, discussions, and assignments have provided me with a comprehensive understanding of the subject matter and have honed my technical and analytical skills. I am thankful for their efforts and the valuable insights they have shared with me.

Lastly, I would like to express my heartfelt appreciation to my family and friends for their unwavering support and encouragement throughout this project. Their belief in me and understanding have been a constant source of motivation and inspiration, and I am truly grateful for their presence in my life.

Abstract

In this report, we present the comprehensive development process of HealthSpace, a platform designed to streamline medical donations. Utilizing a blend of advanced technologies including ReactJs, Bootstrap, CSS, NodeJS, and Python, HealthSpace offers a user-friendly interface and robust functionality tailored to the medical industry's unique needs.

The focal point of HealthSpace is its web-based dashboard, meticulously crafted using ReactJs and Bootstrap, complemented by a sleek CSS styling system. This dashboard provides users with intuitive navigation and seamless access to critical features and information.

Behind the scenes, the system's backend architecture is powered by NodeJS, ensuring optimal performance and scalability. The logical, functional, and API components are intricately woven together to facilitate smooth data flow and efficient processing of requests. Additionally, the system's database management is entrusted to MongoDB, offering robust data handling capabilities and ensuring data integrity.

HealthSpace is designed to be accessed primarily via desktop or laptop devices through the internet, ensuring flexibility and convenience for users across various platforms. With its sophisticated technology stack and user-centric design, HealthSpace is poised to revolutionize the landscape of medical donations, facilitating seamless contributions, and enhancing accessibility to essential healthcare resources.

Table of Contents

Table of figures

Contents

Table of Figures

Abbreviations

- IT – Information Technology
- QA – Quality Assurance
- SDLC – Software Development Life Cycle
- GUI - Graphical user interface
- DB – Database

1 Chapter 1 Introduction

The proposed HealthSpace System integrates ReactJS, Bootstrap, CSS, node.js, and MongoDB to create an intuitive platform tailored for managing healthcare donations and facilitating interactions between donors and recipients. One of its primary objectives is to establish a secure digital interface for tracking and analyzing donation data. Healthcare professionals can seamlessly input donation details into the system, ensuring efficient management and providing transparency for donors.

This comprehensive web-based system encompasses various functionalities, including donation management, communication between donors and recipients, and real-time updates through an online notice board. Leveraging Next.js, a React framework, the platform guarantees a dynamic user experience, with dedicated pages and API routes ensuring secure donation tracking and management.

Furthermore, the system introduces an innovative feature that predicts the most impactful allocation of donations based on healthcare needs, trends, and recipient requirements. Incorporating advanced data analytics and machine learning algorithms, this predictive modeling empowers healthcare organizations to optimize their resource allocation and maximize the impact of donations, ultimately enhancing healthcare delivery and outcomes.

1.1 Goals

1. Enhance Accessibility: Make healthcare donations more accessible by providing a user-friendly digital platform for both donors and recipients.

2. Transparency and Accountability: Ensure transparency in donation processes and accountability in resource allocation through detailed tracking and reporting features.

3. Streamline Donation Management: Simplify the process of managing healthcare donations by implementing efficient data entry and tracking systems.

4. Foster Communication: Facilitate seamless communication between donors and recipients, enabling real-time updates and feedback to enhance donor engagement and recipient satisfaction.

5. Optimize Resource Allocation: Utilize advanced analytics and predictive modeling to optimize the allocation of healthcare donations, maximizing their impact and effectiveness.

6. Empower Decision-Making: Provide healthcare organizations with actionable insights and data-driven recommendations to support informed decision-making in donation management and resource allocation.

7. Promote Collaboration: Foster collaboration between healthcare institutions, donors, and recipients to create a networked community dedicated to improving healthcare delivery and outcomes.

8. Ensure Security and Privacy: Implement robust security measures to safeguard sensitive donation data and ensure compliance with privacy regulations to maintain donor trust and confidence.

9. Drive Innovation: Continuously innovate and evolve the HealthSpace System to adapt to changing healthcare needs and technological advancements, ensuring its relevance and effectiveness in the long term.

10. Make a Positive Impact: Ultimately, strive to make a positive impact on healthcare delivery and outcomes by facilitating efficient donation management and optimizing resource allocation to address critical healthcare needs.

1.2 **Motivation**

The motivation behind the HealthSpace System stems from a fundamental need within the healthcare community for a streamlined and accessible platform for managing donations effectively. In today's digital age, individuals and organizations alike are increasingly turning to social media and word-of-mouth channels in search of avenues to contribute to healthcare causes. However, the lack of centralized platforms often results in fragmented efforts and missed opportunities for impactful donations.

Healthcare donation initiatives are often scattered across various platforms and channels, making it challenging for both donors and recipients to connect efficiently. Despite the widespread desire to contribute to meaningful causes within the healthcare sector, the absence of a dedicated platform hampers the ability to channel these intentions into tangible actions.

The inability to find reliable avenues for healthcare donations leads to frustration and disillusionment among potential donors who are eager to make a difference. Likewise, healthcare institutions and organizations in need of support find themselves grappling with the challenge of effectively communicating their needs and connecting with potential donors.

By addressing these critical gaps, the HealthSpace System aims to revolutionize the landscape of healthcare donations. By providing a centralized platform that brings together donors and recipients in a transparent and user-friendly manner, we aspire to empower individuals and organizations to make meaningful contributions to healthcare causes.

Through the HealthSpace System, donors will have the opportunity to discover and support a wide range of healthcare initiatives, knowing that their contributions are making a tangible impact where it's needed most. Simultaneously, healthcare institutions and organizations will gain access to a powerful tool for effectively managing and communicating their donation needs, fostering greater collaboration and support within the healthcare community.

Ultimately, our motivation lies in the belief that by harnessing the power of technology and community collaboration, we can drive positive change in the healthcare sector and make a meaningful difference in the lives of those in need.

1.3 **Method**

The execution of the HealthSpace System project was characterized by a methodical and iterative approach, commencing with a comprehensive analysis of healthcare donation management challenges. The team selected a technology stack comprising ReactJS, Bootstrap, CSS, node.js, and MongoDB to build a robust and user-friendly platform. Leveraging these technologies, the project aimed to address the complexities of healthcare donation management and foster seamless interaction between donors and recipients.

The development process kicked off with the establishment of the foundational architecture using ReactJS and Bootstrap, ensuring a responsive and intuitive user interface. Subsequently, the focus shifted to the implementation of donation management functionalities, including secure data storage and retrieval using MongoDB. Emphasis was placed on creating a secure digital interface for donors to input donation details, streamlining the process and enhancing transparency.

Furthermore, features facilitating communication between donors and recipients were integrated, such as real-time updates and feedback mechanisms. These enhancements aimed to foster a sense of community and engagement within the healthcare donation ecosystem.

A key highlight of the technical implementation was the incorporation of advanced analytics and predictive modeling to optimize donation allocation. Machine learning algorithms were deployed to analyze donation trends and recipient needs, enabling informed decision-making and maximizing the impact of donations.

Throughout the project lifecycle, an agile methodology was employed, allowing for continuous iteration, testing, and refinement. This iterative approach ensured that the HealthSpace System met the evolving needs of stakeholders and remained aligned with its overarching goal of enhancing healthcare donation management and fostering meaningful interaction within the healthcare community.

1.4 **Overview**

The technical framework of the HealthSpace System revolves around a combination of ReactJS, Bootstrap, CSS, node.js, and MongoDB, ensuring a robust and versatile platform for healthcare donation management. ReactJS provides the foundation for building dynamic and responsive user interfaces, while Bootstrap and CSS contribute to the platform's aesthetic appeal and user experience.

Node.js serves as the backend framework, enabling efficient data handling and processing, while MongoDB serves as the centralized database for storing and retrieving donation-related information securely.

The donation management system within HealthSpace employs interactive web forms to facilitate seamless input of donation details by donors, ensuring ease of use and accessibility. Real-time updates and notifications are integrated into the platform to enhance user engagement and provide timely information to both donors and recipients.

A standout feature of the technical landscape is the incorporation of advanced analytics and predictive modeling techniques to optimize donation allocation. Machine learning algorithms analyze donation trends and recipient needs, providing insights that empower healthcare organizations to make informed decisions about resource allocation.

The technical architecture of the HealthSpace System reflects a commitment to delivering a secure, responsive, and innovative platform for healthcare donation management, prioritizing scalability, performance, and user experience.

2 Chapter 2 Background and Problem Statement

2.1 Introduction

In recent years, the landscape of healthcare donation management has undergone significant transformation, fueled by advancements in technology and a growing emphasis on community engagement and support. However, despite the strides made in this field, several challenges persist, hindering the efficient and equitable distribution of healthcare donations. This introduction provides an overview of the background context and problem statement surrounding healthcare donation management, as well as an exploration of the HealthSpace System's potential to address these challenges.

The current situation in healthcare donation management reveals a series of complex issues that impact the effectiveness and accessibility of donation platforms. Firstly, there exists a lack of awareness among the general population regarding the existence and functionality of healthcare donation platforms. This limited awareness hampers the platform's reach and potential impact, as potential donors and recipients may not be aware of the opportunities available to them.

Furthermore, the digital divide exacerbates disparities in access to healthcare donation platforms, with certain demographics facing unequal access to digital technology and the internet. This digital exclusion not only limits participation but also perpetuates existing inequalities in healthcare access.

Data security concerns also loom large in the realm of healthcare donation management, with users expressing apprehensions about the security of sensitive health information and personal details stored within the platform. Building trust and credibility becomes paramount in fostering user confidence in the platform's authenticity and reliability.

Language and cultural barriers pose additional challenges, impeding effective communication between potential donors and recipients from diverse linguistic and cultural backgrounds. Overcoming these barriers requires tailored strategies that prioritize inclusivity and cultural sensitivity.

Moreover, resource disparities and logistical challenges further complicate the distribution and allocation of healthcare donations, particularly in regions with limited access to healthcare resources. Regulatory compliance, sustainability, user engagement, financial constraints, user education, emergency response mechanisms, and legal and ethical considerations add layers of complexity to the healthcare donation landscape, highlighting the multifaceted nature of the challenges at hand.

2.2 **Literature Review**

Research in the field of healthcare donation systems and organ transplantation has yielded valuable insights into the challenges and opportunities associated with facilitating organ donations and healthcare equipment transactions. Several studies have investigated various aspects related to organ donation, recipient-donor matching, ethical considerations, and the impact of technology on improving the donation process. Here is a summary of key findings from relevant literature:

<u>Organ Donation Rates and Challenges:</u>

- Numerous studies have examined the factors influencing organ donation rates, including cultural beliefs, religious practices, and legal frameworks. Research indicates that despite efforts to increase awareness and streamline the donation process, organ donation rates remain suboptimal in many regions. (Soqia, 2022)

<u>Technology in Organ Donation:</u>

- The integration of technology, particularly through online platforms and databases, has been explored as a means to improve organ procurement and allocation processes. Studies have highlighted the potential of digital platforms to enhance communication between donors, recipients, and healthcare professionals, leading to more efficient organ matching and allocation (Meena, 2022).

<u>Ethical and Legal Considerations:</u>

- Ethical and legal issues surrounding organ donation, including consent, equity, and allocation criteria, have been extensively examined. Research emphasizes the importance of transparent and equitable allocation systems, as well as robust ethical guidelines to ensure fairness and respect for donor autonomy (Bunnik, 2022).

<u>Impact of Social Media and Online Communities:</u>

- The role of social media and online communities in facilitating organ donation and raising awareness has gained attention in recent years. Studies have explored the effectiveness of social media campaigns, peer support networks, and online platforms in encouraging individuals to register as organ donors and engage in discussions about donation-related topics (Pacheco, 2017).

Patient-centered Approaches:

- Patient-centered approaches to organ donation and transplantation, focusing on the needs and experiences of both donors and recipients, have emerged as a key area of research. Studies advocate for tailored support services, improved communication channels, and enhanced decision-making tools to empower patients and optimize outcomes throughout the donation process (Zheng, 2020).

By synthesizing findings from existing literature, this review provides valuable insights into the current state of knowledge regarding healthcare donation systems and organ transplantation. These insights will inform the development and implementation of the proposed web application, contributing to the advancement of organ donation practices and the enhancement of healthcare delivery in Sri Lanka.

Research about similar solutions

Dialog - NBTS SMS Blood Service.

Dialog Company introduced a SMS based blood donor management application together with the National blood transfusion service, Sri Lanka. Introducing volunteer donors and adding them to the national blood supply system is the main goal of this project. From that they hope to help the Sri Lankan peoples who need blood in an emergency situation. When someone registers to SMS blood donation system through SMS their details will be stored in Database which maintains national blood transfusion service. Then in a situation like needing emergency blood the national blood transfusion service can contact donors easily. To register for SMS blood; one should Type BLOOD and send to 7777 from your Dialog mobile (Anon., n.d.).

Online Blood and Organ Transplant Management System

Professor Alan Lopez reports from the Department of Information Technology at the University of Engineering, Wasai, India mentioned in his research that Blood is an essential component of all living organisms. In the event of an emergency, it proves to be a lifesaving component. Users can view all of the data in this section. The major goal of creating this software is to drastically minimize the amount of time spent looking for the suitable donor and ensuring that the blood needed is available. In addition, the user can register for organ donation after death to those who are in need. Online Blood Bank and Organ Transplant will be a website. The system's goal is to make the process of searching for blood in an emergency easier and more automated, as well as to keep track of blood donors, recipients, blood donation programs, and blood stocks in the bank. Blood seekers can use this website to look for blood donors and then call or message them using this Android app. This website may be used by both organ donors and organ seekers, and it allows people to register for organ donation. The proposed system will feature a directory that lists all blood banks in India. The user can search for a Blood Bank by entering their PIN code and the name of the state or city (Prajapati, 2017).

UNOS-Organ Donation Application and Website

Prof. Raghunathsingh Rajput, Javeriya Soudagar, Pavitra Viraktamatha, Snehal Jadhav, Srushti SG mentioned in his research that Organ donation is one among the many contributions that an individual can make to society. The organ donation system may be a combination of Android application and website that's made for such noble and great causes. The growing technology in android development has made this possible; the hospitals, organ banks, medical stores, ambulances and users can register through a web organ donation application provides how for seekers to look for donors through the app. This application is additionally employed by organ donors and seekers where an individual can register for interest in organ donation. The procedure to use the appliance is to first download the appliance on your smartphone, register on the app with basic details like name, address, contact details, email and medical record information of organ one could prefer to donate. All this information is often saved during a data server that's interconnected between user, donor, seeker and hospital with the multi-connectivity server. This Android application was created to make contacting the hospital as simple 3 as possible. Provisional organ allocation begins when a donor has been examined and consent has been gained. UNOS created a computer program that provides a donor specific match list for appropriate recipients based on the parameters that the patient provided. The proposed solution entails developing an Android application that makes organ donation simple. Donor cards are issued to all donors who have registered in this application. Family members must notify the medical committee of the donor's death within 6 hours of the donor's death by filling out this application in one step (R. Rajput, 2019).

Blood Bank PH: A Framework for an Android-based Application for the Facilitation of Blood Services in the Philippines

Abigail Casabuena et al developed a framework for an Android-based application to facilitate blood services in their article. Blood service institutions, blood donors, and persons in need of blood will all benefit from the proposed system. Blood services such as blood donation and blood request would be made easier and more convenient with the proposed Android mobile application. Blood donors and those in need of blood can use their present location to find the nearest blood bank. People who download the mobile application will be informed about the newest activities of the Philippine Red Cross blood service facilities, such as blood drives, which will boost their readiness to donate and raise awareness of the critical need for blood, resulting in more donations. The proposed online system will assist blood service institutions in managing blood services and keeping a current database of blood donors and requesters. The reports that will be generated will be crucial since they will serve as the foundation for events such as blood drives. The suggested system also displays an up-to-date inventory of all blood service institutions' blood supplies. If the blood supply falls below a critical level, an automatic broadcast will be made to encourage blood donors to donate. The suggested technology will be implemented in all Philippine Red Cross blood service centres in the future, according to the proponents (Casabuena, 2019).

An Android Application for Blood and Organ Donation Management

Prithvi Veenu, Niranjan Kumara B, Venkata Ramana G, Dr. Sethukarasi T.A has mentioned in their research paper that a mobile application that might bridge the gap is that the need of the hour, life siren is not any different and it's a touch different from the prevailing system, our application makes use of a mobile app at both ends, 4 anyone can register and switch bent be a user, the hospital when in need raises an invitation for blood/organ of a selected type with all the small print , and this request will notify all the users and therefore the nearest medical bank centers of it, the request will have a licensed e-signature of the handling doctor with their contact details, and therefore the application will let the recipient choose the foremost feasible and nearby donor for help. The app is about to start out on Android and can make use of a SQLite database at the application's end. The application is called “LIFE SIREN”. It aims to close the gap between patients and donors in terms of coordination and communication. The application begins with a Firebase Authentication Mechanism-based authentication check. The authentication technique is divided into two types: organization authentication and user authentication. They are granted access to the application after properly authenticating. The application keeps track of all of the big hospitals and organ banks in each city, assigning them a verified status, and allowing those hospitals to verify other small hospitals to join the service. When an organ or blood donation is required, the user can submit a request using the app with the help of a practicing doctor's ID. When more than one hospital or organ bank is ready to give, the request is updated in Cloud SQL and broadcasted to all hospitals and organ banks that can react to requests through the Application. This program features a separate viewable panel where you can specify all of the requests made to you, and your response will be saved in the cloud database. Once a response is received on the cloud database, we can view the responses to our request later (P. Venu, 2019).

Applications of simulation within the healthcare context

One of the researchers was employed as a research fellow in Warwick Business School while working on this research, and wishes to thank the School for supporting this research. They also thank Dr. Simon J. E. Taylor for his comments, which have improved the paper. A large number of studies have applied simulation to a multitude of issues relating to healthcare. These studies have been published in a number of unrelated publishing outlets, which may hamper the widespread reference and use of such resources. In this paper, we analyze existing research in healthcare simulation in order to categorize and synthesize it in a meaningful manner. Hence, the aim of this paper is 5 to conduct a review of the literature pertaining to simulation research within healthcare in order to ascertain its current development. A review of approximately 250 high quality journal papers published between 1970 and 2007 on healthcare-related simulation research was conducted. The results present a classification of the healthcare publications according to the simulation techniques they employ; the impact of published literature in healthcare simulation; a report on demonstration and implementation of the studies' results; the sources of funding; and the software used. Healthcare planners and researchers will benefit from this study by having ready access to an indicative article collection of simulation techniques applied to healthcare problems that are clustered under meaningful headings. This study facilitates the understanding of the potential of different simulation techniques in solving diverse healthcare problems (K. Katsaliaki and N. Mustafee, 2017).

IT support for healthcare processes – premises, challenges, perspectives

Richard Lenz holds a Ph.D. in Computer Science and is currently temporary Director of the Institute of Medical Informatics at the University of Marburg, Germany has finished his habilitation thesis on "Evolutionary Information Systems", where he presents a generally applicable reference model for layered system architectures as well as an adapted strategic approach for the Hospital Information System in Marburg in 2005. In 1997 he earned his Ph.D. from the University of Erlangen with his thesis on "Adaptive Data Replication in Distributed Systems". Healthcare processes require the cooperation of different organizational units and medical disciplines. In such an environment optimal process support becomes crucial. Though healthcare processes frequently change, and therefore the separation of the flow logic from the application code seems to be promising, workflow technology has not yet been broadly used in healthcare environments. In this paper we elaborate both the potential and the essential limitations of IT support for healthcare processes. We identify different levels of process support in healthcare, and distinguish between organizational processes and the medical treatment process. To recognize the limitations of IT support we adopt a broad socio-technical perspective based on scientific literature and personal experience. Despite the limitations we identified, undeniably, IT has a huge potential to improve healthcare quality which has not been 6 explored by current IT solutions. In particular, we indicate how advanced process management technology can improve IT support for healthcare processes (R. Lenz and M. Reichert, 2007).

Android application for blood donation and organ donation and awareness

Introduced an android application named as LIFESAVER to blood donation and organ donation and awareness. This system for reducing the gap between seeker and donor. This system is for all seeker, donor or hospital whenever there is need as for hospitals they can raise a request for demand which will be notified with all the details to all donors, blood banks and organ banks.

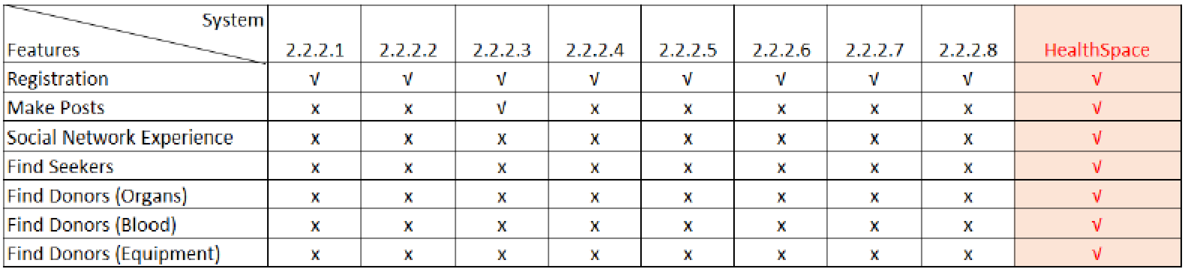
Kshipra B.Panaskar, Akansha N.Nakate, Siddhi R.Mhatre are students of department of computer engineering, M.G.M. College of engineering and Technology, Kamothe, Maharashtra in India and Prof. Sachin Chavan professor department of computer engineering, M.G.M. College of engineering and Technology, Kamothe, Maharashtra in India were studying and researching the healthcare domain and identify lack of blood, organ donation problems and wrote this article in 2021. As the main findings in this source, the problem is due to manual work, awareness among people was less about blood and organ donation or transplantation. In India, many people are losing their lives because they are suffering from lack of blood & organs in time. So that as a solution, use the latest technologies and the tools to find a system that fills the gap and provides an organized solution. So the proposed system is an android application that helps to bridge the gap between seeker and donor. An application which supports current technology and is user friendly. Consumes less time and gives faster results as compared to other previous existing systems. Another author's contribution in this source is seven references and other resources. Our project title is blood and organ donation mobile application in the healthcare domain. This is a source specified about android application for blood and organ donation and awareness. Also it is in the healthcare domain. LIFESAVER solves the problem closest to our project problem. So this source is relevant to our topic (K. B. Panaskar, 2021).

Summary

- Similar Approach 1 - Dialog - NBTS SMS Blood Service.
- Similar Approach 2 - Online Blood And Organ Transplant Management System
- Similar Approach 3 - An Android Application For Blood And Organ Donation Management
- Similar Approach 4 - UNOS-Organ Donation Application And Website
- Similar Approach 5 - Blood Bank PH: A Framework for an Android based Application

for the Facilitation of Blood Services in the Philippines

- Similar Approach 6 - Applications of simulation within the healthcare context
- Similar Approach 7 - IT support for healthcare processes – premises, challenges, perspectives
- Similar Approach 8 - Android application for blood donation and organ donation and awareness
- Author's approach (A mobile application - HealthSpace)



System Features	2.2.2.1	2.2.2.2	2.2.2.3	2.2.2.4	2.2.2.5	2.2.2.6	2.2.2.7	2.2.2.8	HealthSpace
Registration	√	√	√	√	√	√	√	√	√
Make Posts	x	x	√	x	x	x	x	x	√
Social Network Experience	x	x	x	x	x	x	x	x	√
Find Seekers	x	x	x	x	x	x	x	x	√
Find Donors (Organs)	x	x	x	x	x	x	x	x	√
Find Donors (Blood)	x	x	x	x	x	x	x	x	√
Find Donors (Equipment)	x	x	x	x	x	x	x	x	√

Figure 1: Feature Checklist

2.3 Problem Statement

Against this backdrop, the HealthSpace System aims to address these challenges by providing a centralized, user-friendly platform for healthcare donation management. By leveraging innovative technologies and adopting a comprehensive approach to addressing the identified issues, HealthSpace seeks to enhance the efficiency, transparency, and accessibility of healthcare donation processes, ultimately contributing to improved healthcare outcomes and community well-being.

3 Chapter 3 project management

The management of the HealthSpace project adopts an Agile Scrum methodology, which emphasizes flexibility, collaboration, and iterative development. The project is organized into sprints, each typically lasting two to four weeks, during which specific features or functionalities are planned, developed, tested, and delivered.

At the outset of each sprint, a sprint planning meeting is held to prioritize tasks from the product backlog, ensuring that the most valuable and impactful features are addressed first. The product backlog is maintained and managed throughout the project, with new items added, refined, and prioritized based on changing requirements and stakeholder feedback.

The project team is cross-functional, comprising members with diverse skills and expertise, including developers, designers, testers, and domain experts. Daily stand-up meetings are conducted to facilitate communication, identify any impediments or challenges, and ensure alignment on project goals and progress.

Throughout the sprint, the project manager, acting as the Scrum Master, oversees the sprint progress, removes any obstacles hindering the team's productivity, and ensures adherence to Agile principles and practices. Regular sprint reviews and retrospectives are conducted to evaluate the completed work, gather feedback from stakeholders, and identify areas for improvement in future sprints.

The Agile Scrum approach allows for flexibility and adaptability in response to changing requirements or priorities, enabling the project team to deliver value incrementally and iteratively. By fostering collaboration, transparency, and continuous improvement, the project management framework ensures the successful development and delivery of the HealthSpace System, meeting the needs and expectations of stakeholders effectively.

3.1 Approach

The development approach for the HealthSpace System revolves around iterative and incremental development, focusing on delivering value to users early and often. The project follows a structured Agile Scrum methodology, which enables adaptability, collaboration, and continuous improvement throughout the development lifecycle.

Requirements Gathering: The project begins with a thorough analysis of stakeholder requirements and user needs. This involves conducting interviews, surveys, and workshops to gather insights and prioritize features for implementation.

Sprint Planning: The project is divided into time-boxed iterations called sprints, typically lasting two to four weeks. At the start of each sprint, the project team collaborates to select and prioritize tasks from the product backlog, ensuring that the most valuable features are addressed first.

Development: During the sprint, the development team works collaboratively to implement the selected features. Daily stand-up meetings are held to provide updates on progress, discuss any obstacles, and ensure alignment with project goals.

Testing: Quality assurance and testing are integral parts of the development process. Automated and manual testing techniques are employed to verify the functionality, usability, and performance of the system, ensuring that it meets the specified requirements.

Review and Feedback: At the end of each sprint, a sprint review meeting is held to demonstrate the completed work to stakeholders and gather feedback. This feedback is used to refine and prioritize the product backlog for future sprints.

Retrospective: Following the sprint review, a retrospective meeting is conducted to reflect on the sprint process and identify areas for improvement. Lessons learned are captured and incorporated into the project management practices to enhance team performance and productivity.

Incremental Delivery: The project follows a strategy of incremental delivery, with new features and functionalities released to users at the end of each sprint. This allows for early validation of the system and enables stakeholders to provide feedback for further refinement.

Continuous Improvement: The Agile Scrum approach fosters a culture of continuous improvement, with the project team regularly reflecting on their processes and practices to identify opportunities for enhancement. By embracing change and adapting to evolving requirements, the project ensures the successful delivery of the HealthSpace System while maximizing value for stakeholders.

3.2 Initial Project Plan

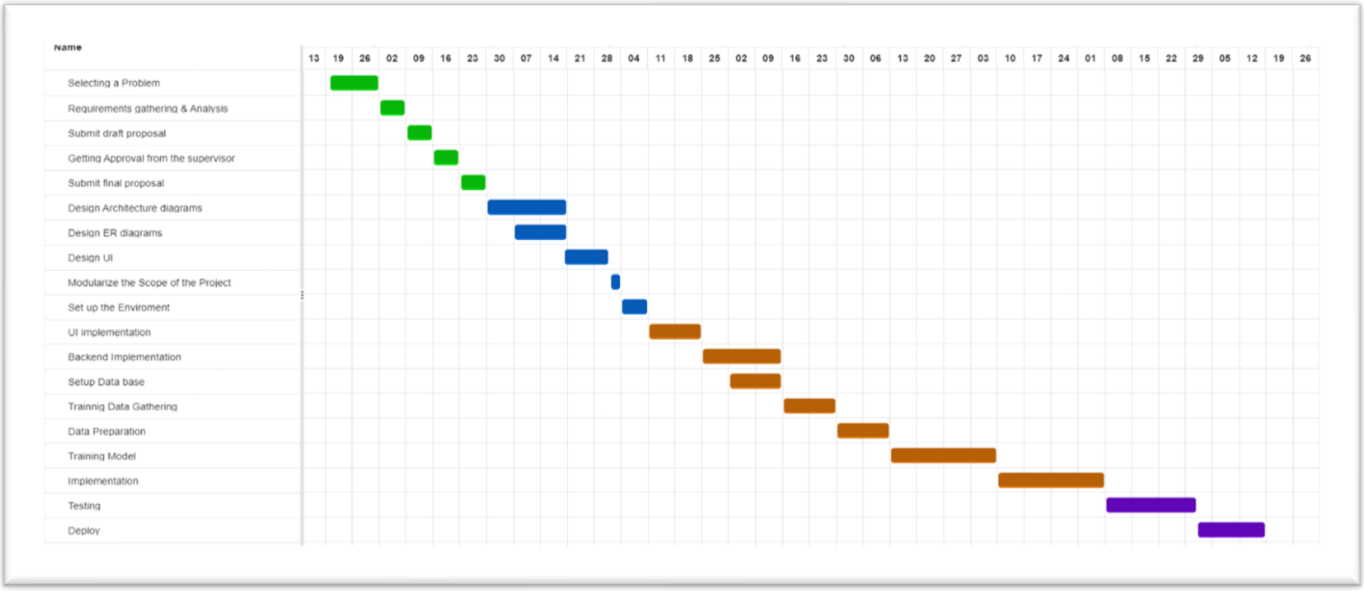


Figure 2: Project Plan

The initial project plan for the HealthSpace System aimed to develop a comprehensive ReactJS web-based platform to address critical challenges in healthcare donation management. The plan outlined the creation of a Donation Management System with a MongoDB centralized database, providing efficient storage and retrieval of donation-related information. Additionally, features such as a Donation Input Form for streamlined donation entry and an Online Notice Board for real-time updates were included to enhance user engagement and communication.

Furthermore, the plan incorporated an innovative predictive feature leveraging data analytics and machine learning algorithms to optimize donation allocation. Milestones and timelines were established to guide the project's progress, with a focus on iterative development and continuous improvement. The technology stack initially included ReactJS for frontend development and MongoDB for database management, chosen based on their suitability for the project requirements.

Throughout the project lifecycle, an Agile Kanban development approach was employed, allowing for flexibility and adaptability to evolving project needs. Iterative feedback loops and adjustments were emphasized to ensure alignment with stakeholder expectations and project goals. As the project progressed, refinements and optimizations were made to the technology stack and development processes to support the successful delivery of the HealthSpace System.

3.3 Problems and Changes to the Plan

Several potential problems and changes to the project plan for the HealthSpace System may arise throughout its development lifecycle. These could include:

1. Technical Challenges: Unexpected technical difficulties or limitations may emerge, such as compatibility issues with certain browsers or platforms, performance bottlenecks, or scalability concerns. These challenges may necessitate adjustments to the technology stack, architecture, or development approach to address and mitigate these issues effectively.

2. Scope Creep: As the project progresses, stakeholders may request additional features or functionalities beyond the initial project scope. Managing scope creep requires careful prioritization and negotiation to ensure that project goals remain achievable within the allocated resources and timeline.

3. Resource Constraints: Limited availability of skilled team members, budgetary constraints, or time limitations may impact the project's progress and ability to meet its objectives. Adjustments to the project plan may be necessary to optimize resource allocation, streamline workflows, or seek additional support as needed.

4. Stakeholder Expectations: Changes in stakeholder requirements, preferences, or priorities may necessitate revisions to the project plan to ensure alignment with their expectations and objectives. Effective communication and stakeholder management are essential to address any discrepancies and maintain project momentum.

5. Regulatory Compliance: Evolving regulatory requirements or compliance standards related to healthcare data privacy, security, or donation management may necessitate updates to the project plan to ensure adherence and mitigate associated risks.

6. Integration Challenges: Integrating the HealthSpace System with existing healthcare systems, databases, or third-party platforms may pose integration challenges, such as data synchronization issues or compatibility issues. These challenges may require adjustments to the integration strategy or collaboration with external stakeholders to overcome.

7. User Feedback: Feedback from users during testing or pilot phases may uncover usability issues, user experience concerns, or feature requests that require modifications to the project plan. Iterative user testing and feedback loops are essential to identify and address these issues promptly.

8. External Dependencies: Dependencies on external vendors, partners, or stakeholders may introduce delays or uncertainties into the project timeline. Proactive management of external dependencies and communication with relevant parties are crucial to minimize disruptions and maintain project progress.

3.4 **Final Project Record**

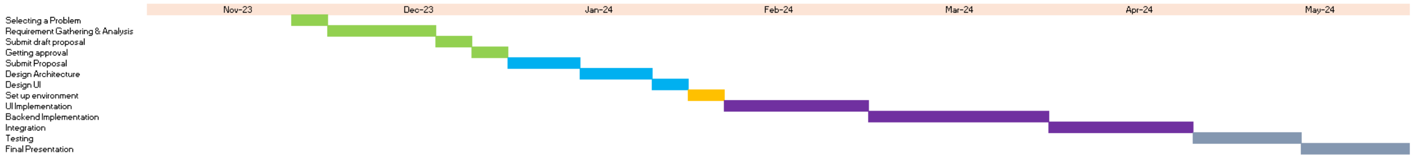


Figure 3: Final plan

4 Chapter 4 feasibility study

A feasibility study is a crucial step in determining the viability of a project or business venture. It involves analyzing and evaluating various factors, including financial, technical, operational, legal, and environmental aspects, to determine if the project is feasible or not. The study is conducted to identify potential challenges, risks, and opportunities associated with the project, and to assess the project's potential to succeed in the market. The feasibility study helps stakeholders make informed decisions about whether to proceed with the project or not, based on the data and insights gathered from the study. The feasibility study includes a comprehensive analysis of the project's objectives, scope, and requirements. The analysis also includes a review of the current market and industry trends, competitor analysis, and a review of regulatory requirements and constraints. The study evaluates the technical feasibility of the project, including the availability of the necessary resources, equipment, and expertise, and the potential risks associated with the technology used. Additionally, the financial feasibility of the project is assessed by analyzing the cost-benefit analysis, revenue projections, and the overall return on investment. The study also evaluates the project's operational feasibility, including the availability of necessary infrastructure, human resources, and the project's impact on the environment. In summary, a feasibility study provides critical insights into the viability of a project, which helps stakeholders make informed decisions about whether to proceed with the project or not. The study assesses various aspects of the project, including financial, technical, operational, legal, and environmental aspects, and provides stakeholders with a comprehensive understanding of the project's potential for success.

4.1 Time feasibility

Time Management: Strategies for effective time management, such as prioritizing tasks, setting realistic goals, and managing work-life balance, have been considered to ensure that project work can be accomplished alongside full-time employment responsibilities.

4.2 Cost feasibility.

Cost Analysis: A comprehensive cost-benefit analysis has been conducted to evaluate the financial feasibility of the project, considering factors such as development costs, infrastructure expenses, and potential revenue streams or cost savings generated by the platform.
Return on Investment (ROI): The projected ROI of implementing the HealthSpace System has been estimated based on anticipated benefits such as increased efficiency in donation management, improved access to healthcare resources, and enhanced donor engagement.

4.3 **Operational feasibility**

Stakeholder Analysis: The needs and expectations of key stakeholders have been analyzed to ensure alignment with project objectives, considering the constraints of balancing project work with full-time employment commitments.
Operational Requirements: The operational requirements for implementing and maintaining the HealthSpace System, including staffing, training, and ongoing support, have been evaluated to determine feasibility while managing time effectively.

4.4 **Technical feasibility.**

Technology Stack: The chosen technology stack, comprising ReactJS for frontend development and MongoDB for database management, has been evaluated for its suitability and compatibility with project requirements.
Development Resources: The availability of skilled developers, access to necessary tools and frameworks, and the feasibility of implementing required features within the given technical constraints have been assessed.

4.5 **Scheduling feasibility.**

Project Timeline: A detailed project timeline has been developed, considering the constraints of working as a full-time employee, to assess the feasibility of meeting project deadlines and objectives.
Resource Allocation: The availability of resources, including time and energy, has been evaluated to determine the feasibility of executing the project within the specified timeframe

5 Chapter 5 design

In the system design phase of the student performance analysis system, meticulous planning and structuring were undertaken to translate the identified requirements into a tangible and functional solution. This phase involved specifying the architecture, components, modules, data management strategies, and the overall interaction flow of the Next.js web-based platform. Notable elements included the design of the Student Mark Management System, encompassing the transition from MongoDB to PostgreSQL for improved data management and the establishment of robust entity relationships. The system design also accounted for the implementation of the Student Question Form and Online Notice Board, ensuring user- friendly interfaces and seamless communication channels. Furthermore, the design phase incorporated the innovative predictive feature, requiring a thoughtful integration of data analytics and machine learning components. Overall, the system design phase laid the foundation for a cohesive, scalable, and feature-rich educational management platform.

5.1.1 ER diagram.

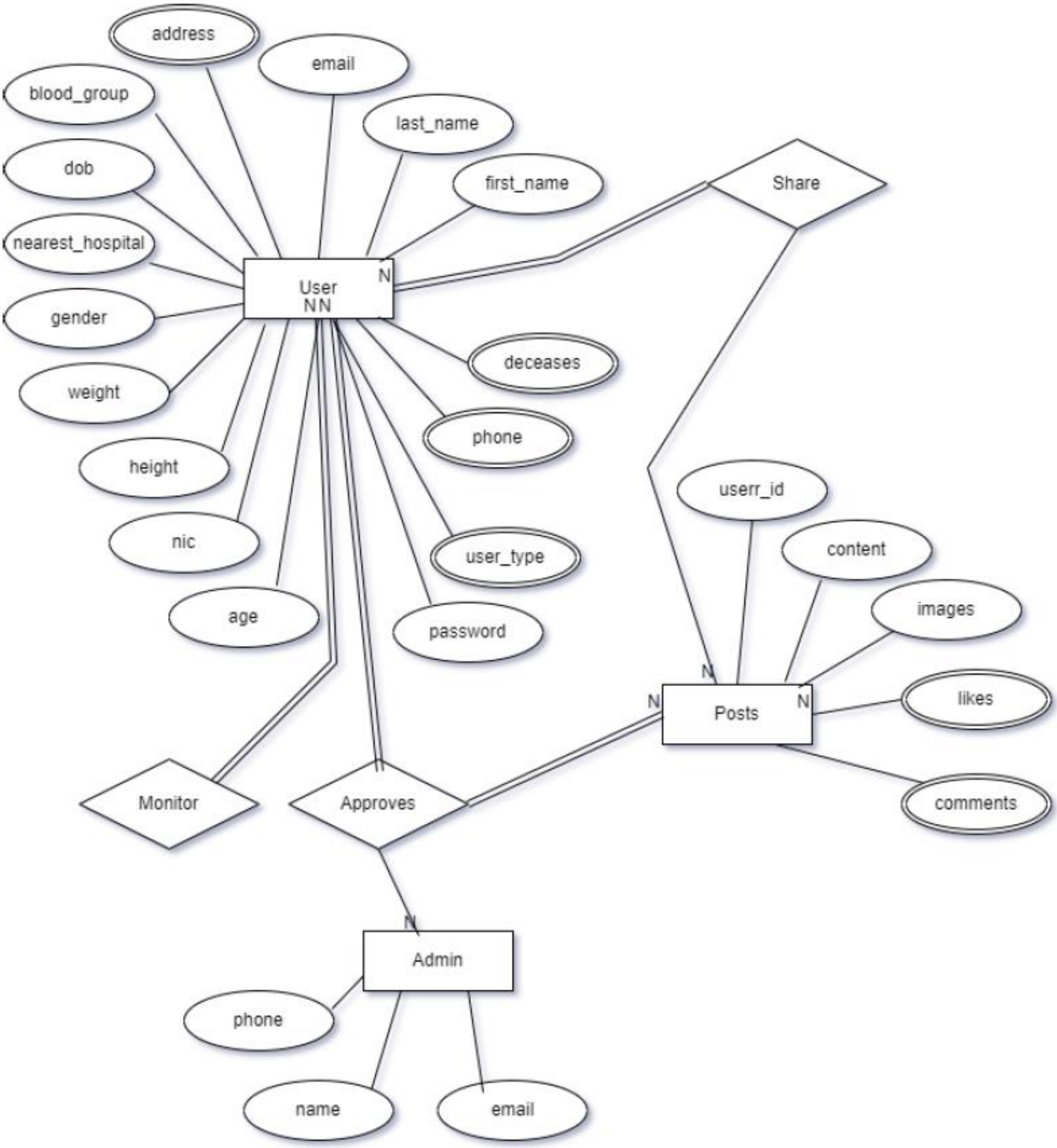


Figure 4: ER Diagram

5.1.2 User Case Diagram

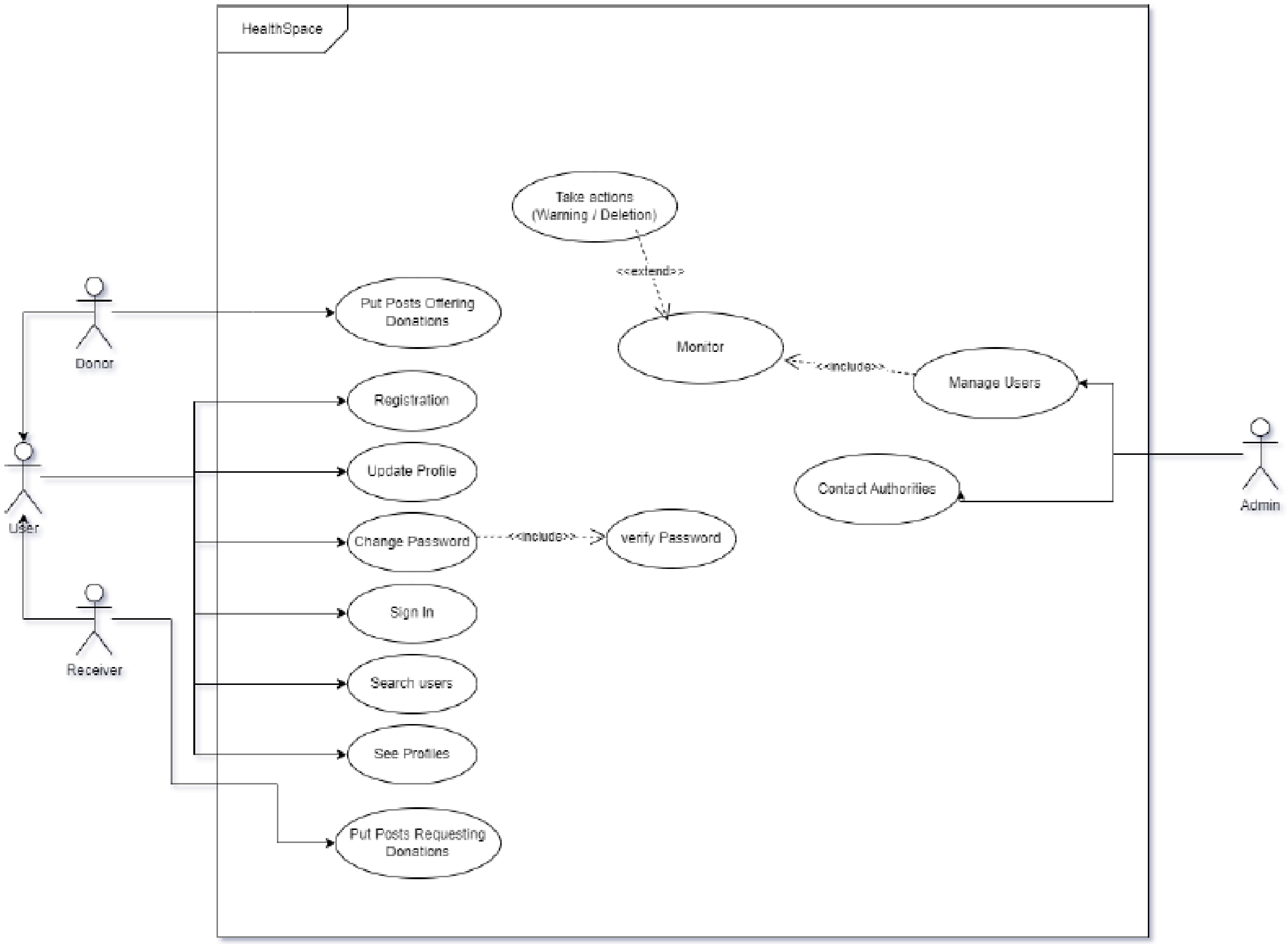


Figure 5: Use Case Diagram

5.1.3 Class Diagram

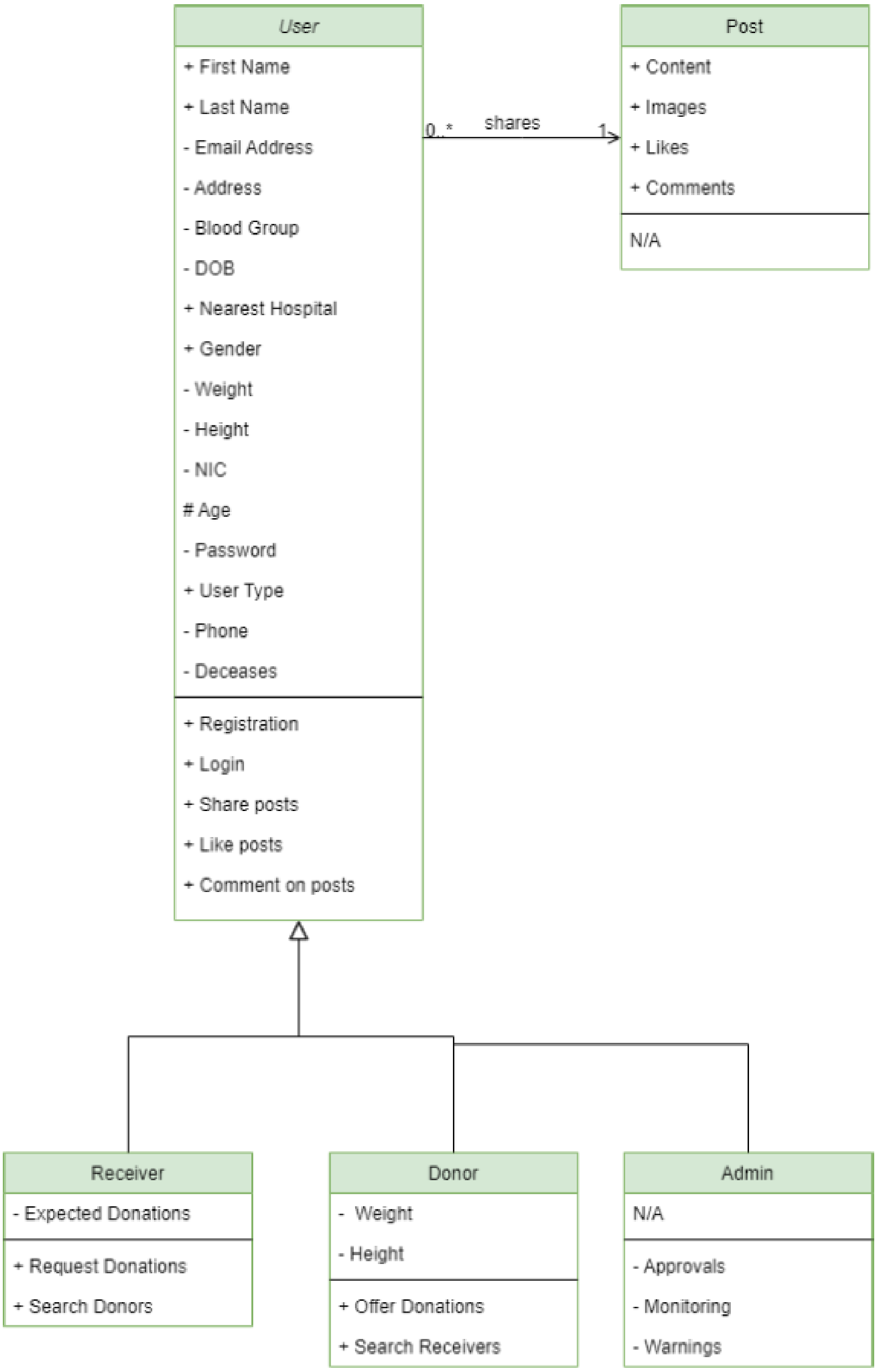


Figure 6: Class Diagram

5.1.4 System Architecture

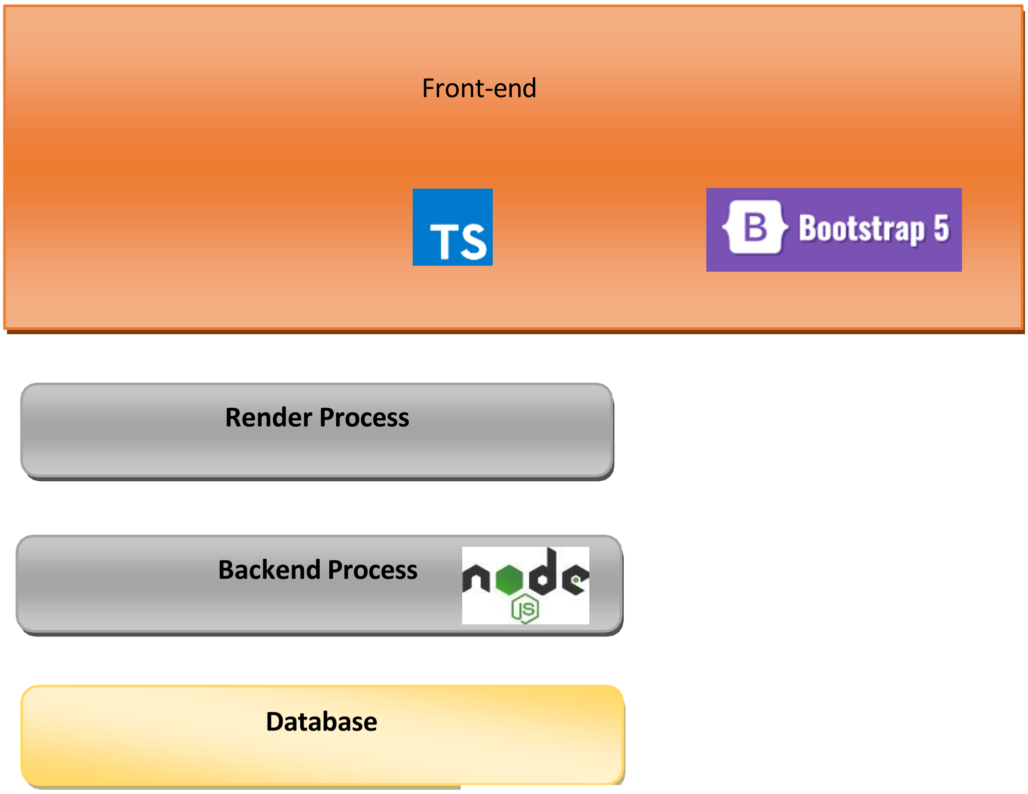


Figure 7: System Diagram

5.2 Hardware and software requirements

5.2.1 Hardware requirements

Computing Devices:

Desktop Computers/Laptops: The development and testing of the HealthSpace System can be conducted using standard desktop computers or laptops with modern specifications, including sufficient processing power, RAM, and storage capacity to support development environments and testing suites.

Server Infrastructure:

Hosting Environment: A cloud-based hosting environment, such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP), is recommended for deploying the HealthSpace System. This hosting environment should offer scalable infrastructure resources, including virtual machines, containers, or serverless computing options, to accommodate varying levels of system usage and traffic.

Virtual Private Servers (VPS) or Dedicated Servers: Alternatively, if an on-premises hosting solution is preferred, virtual private servers (VPS) or dedicated servers with adequate computing resources, storage capacity, and network bandwidth should be provisioned to support the deployment and operation of the HealthSpace System.

Networking Equipment:

Network Connectivity: Reliable internet connectivity is essential for accessing cloud-based hosting environments, facilitating communication between system components, and enabling user interaction with the HealthSpace System. High-speed internet access with sufficient bandwidth is recommended to ensure optimal system performance and responsiveness.

Network Security: Implementation of robust network security measures, such as firewalls, intrusion detection/prevention systems (IDS/IPS), and secure socket layer (SSL) encryption protocols, is crucial to protect the HealthSpace System from unauthorized access, data breaches, and cyber threats.

Backup and Recovery Solutions:

Data Backup: Regular backups of system data, configurations, and user-generated content should be performed to mitigate the risk of data loss due to hardware failures, software errors, or security incidents. Cloud-based backup solutions or on-premises backup servers can be employed to store backup copies securely.

Disaster Recovery: A comprehensive disaster recovery plan should be developed to minimize downtime and ensure business continuity in the event of hardware failures, natural disasters, or other catastrophic events. This plan should include procedures for data restoration, system recovery, and failover to redundant infrastructure components.

Development and Testing Equipment:

Development Workstations: Developers working on the HealthSpace System should have access to adequately equipped development workstations or laptops with development environments, integrated development environments (IDEs), version control systems, and debugging tools installed.

Testing Devices: Testing of the HealthSpace System should be conducted across a variety of devices and screen sizes to ensure compatibility and responsiveness. This may include smartphones, tablets, and other mobile devices, as well as different web browsers and operating systems.

5.2.2 Software requirements

Integrated Development Environment (IDE):

Visual Studio Code (VS Code): The primary IDE for development, editing, and debugging of codebase files, including ReactJS frontend components, Node.js backend scripts, and MongoDB database queries.

Version Control System (VCS):

Git: A distributed version control system used for tracking changes to project files, facilitating collaboration among developers, and managing codebase revisions. Git integration with VS Code allows for seamless version control operations directly within the IDE.

Package Managers:

npm (Node Package Manager): npm is utilized for managing project dependencies, installing third-party libraries, and executing scripts within the Node.js environment. It is essential for installing and managing frontend and backend dependencies, such as ReactJS libraries and Node.js modules.

Web Browsers:

Google Chrome, Mozilla Firefox, Safari, Microsoft Edge: A selection of modern web browsers is necessary for testing and debugging the HealthSpace System across different browser environments. Developers should ensure cross-browser compatibility and adherence to web standards during development.

Database Management System (DBMS):

MongoDB: As the chosen database management system for HealthSpace, MongoDB is used for storing, querying, and managing donation-related data. Integration with VS Code may include MongoDB extensions or plugins for enhanced database interaction and administration.

Operating System:

Windows, macOS, Linux: VS Code is compatible with various operating systems, allowing developers to work on the HealthSpace System across different platforms. The software requirements should accommodate the preferred operating system environment of development team members.

Web Development Tools and Extensions:

React Developer Tools: Browser extensions or developer tools for inspecting React components, debugging React applications, and analyzing component hierarchies during frontend development.

Node.js Debugger: Debugging tools and extensions for Node.js applications, enabling developers to troubleshoot server-side code and diagnose runtime errors effectively.

ESLint, Prettier: Code linting and formatting tools for maintaining code quality, enforcing coding standards, and ensuring consistent code formatting practices across the project.

6 Chapter 6 implementation

The implementation of the HealthSpace System involves the development of a web-based platform using ReactJS for frontend user interfaces, Node.js for backend logic, and MongoDB for database management. Through agile development methodologies, iterative feature implementation, and continuous testing, the project aims to address challenges in healthcare donation management by providing a user-friendly interface for donors and recipients to interact, streamline donation processes, and optimize resource allocation. The implementation process prioritizes scalability, security, and usability, ensuring that the HealthSpace System meets the evolving needs of stakeholders while maintaining high standards of performance and reliability.

6.1 User interface implementation

UI (User Interface) implementation for the HealthSpace System involves designing and developing the visual elements and interactive components that users interact with to navigate the platform, input data, and access information. The UI implementation process encompasses several key steps:

1. Design Phase:

- User Interface Design: Based on user requirements, wireframes, and mockups are created to visualize the layout, structure, and flow of the interface. This includes defining the placement of elements such as navigation menus, forms, buttons, and content areas.
- Visual Design: The visual design of the UI, including color schemes, typography, iconography, and branding elements, is established to create a cohesive and visually appealing user experience. Design tools such as Adobe XD, Sketch, or Figma may be used to create high-fidelity designs.

2. Frontend Development:

- HTML/CSS Markup: The UI design is translated into HTML markup and styled using CSS to define the layout, appearance, and styling of interface elements. Responsive design techniques are employed to ensure compatibility with various screen sizes and devices.
- Component Development: UI components such as navigation bars, buttons, input fields, modals, and cards are developed using frontend frameworks like ReactJS. These components

are reusable, modular, and customizable, facilitating consistency and efficiency in UI development.

- Interaction Design: Interactive elements such as dropdown menus, sliders, tabs, and accordions are implemented to enhance user engagement and facilitate intuitive navigation. JavaScript libraries and frameworks may be utilized to add dynamic behavior and interactivity to the UI.

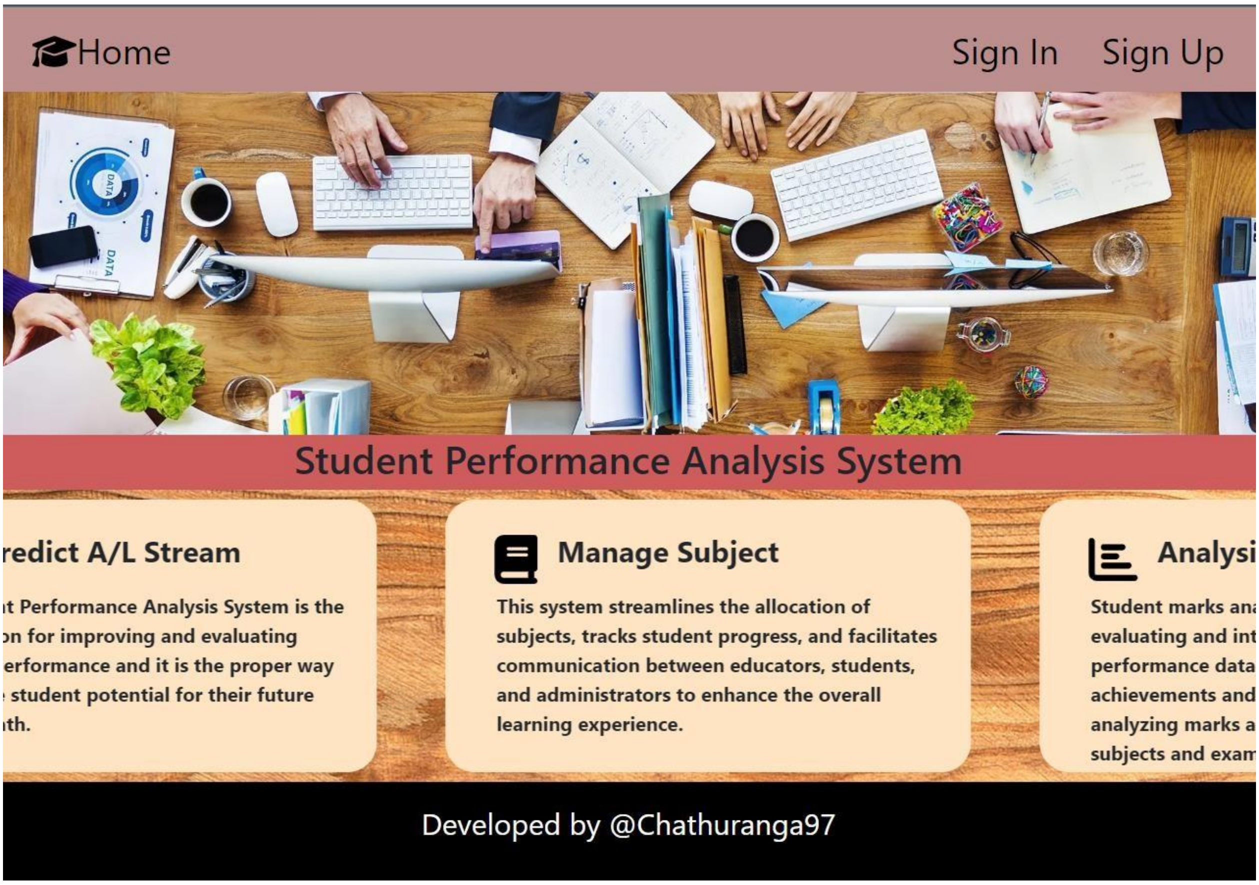
3. Accessibility and Usability:

- Accessibility Compliance: The UI is designed and developed to adhere to accessibility standards such as WCAG (Web Content Accessibility Guidelines), ensuring that users with disabilities can navigate and interact with the platform effectively. This includes providing alternative text for images, semantic HTML markup, and keyboard navigation support.
- Usability Testing: The UI undergoes usability testing to evaluate the effectiveness, efficiency, and satisfaction of user interactions. Feedback from usability tests is used to identify areas for improvement and refine the UI design and implementation accordingly.

4. Integration and Testing:

- Integration with Backend: The frontend UI components are integrated with backend APIs and services developed using technologies like Node.js and MongoDB. This integration enables data exchange, authentication, and business logic execution between the frontend and backend layers.
- Cross-Browser and Cross-Device Testing: The UI is tested across different web browsers (e.g., Chrome, Firefox, Safari, Edge) and devices (e.g., desktops, laptops, tablets, smartphones) to ensure compatibility, consistency, and responsiveness.
- Performance Optimization: Performance optimizations such as code minification, image compression, lazy loading, and caching are applied to enhance the speed and responsiveness of the UI, providing a seamless user experience.

Overall, UI implementation for the HealthSpace System focuses on creating an intuitive, accessible, and visually appealing interface that enables users to accomplish their tasks efficiently while aligning with the project's objectives and user needs.



Home

Sign In Sign Up

Student Performance Analysis System

redict A/L Stream

t Performance Analysis System is the on for improving and evaluating erformance and it is the proper way student potential for their future th.

Manage Subject

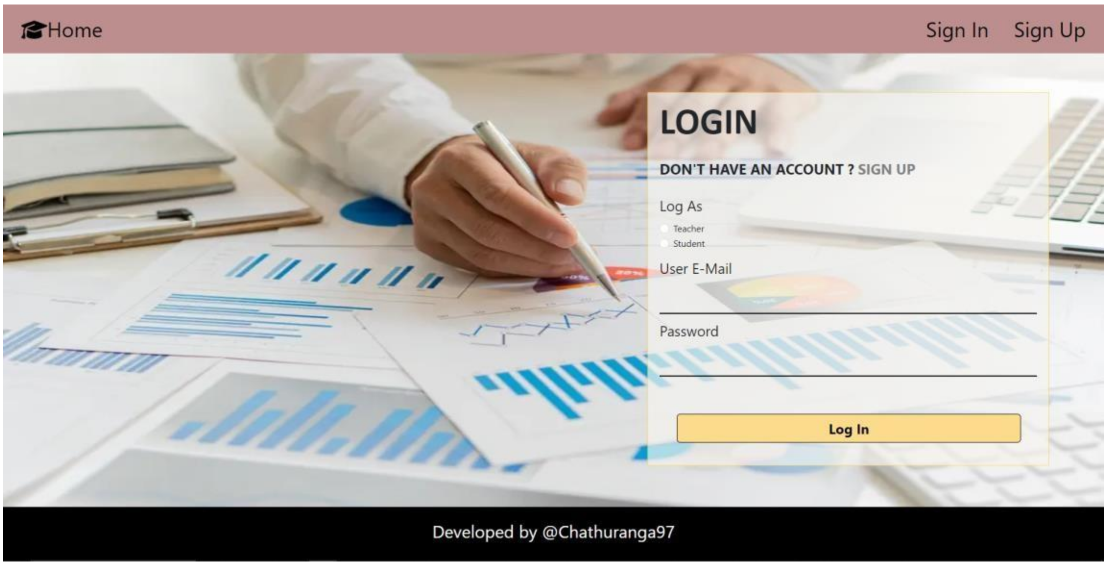
This system streamlines the allocation of subjects, tracks student progress, and facilitates communication between educators, students, and administrators to enhance the overall learning experience.

Analysi

Student marks ana evaluating and int performance data achievements and analyzing marks a subjects and exam

Developed by @Chathuranga97

Figure 8: Landing Page



Home

Sign In Sign Up

LOGIN

DON'T HAVE AN ACCOUNT ? SIGN UP

Log As

○ Teacher

○ Student

User E-Mail

Password

Log In

Developed by @Chathuranga97

Figure 9: Login Page

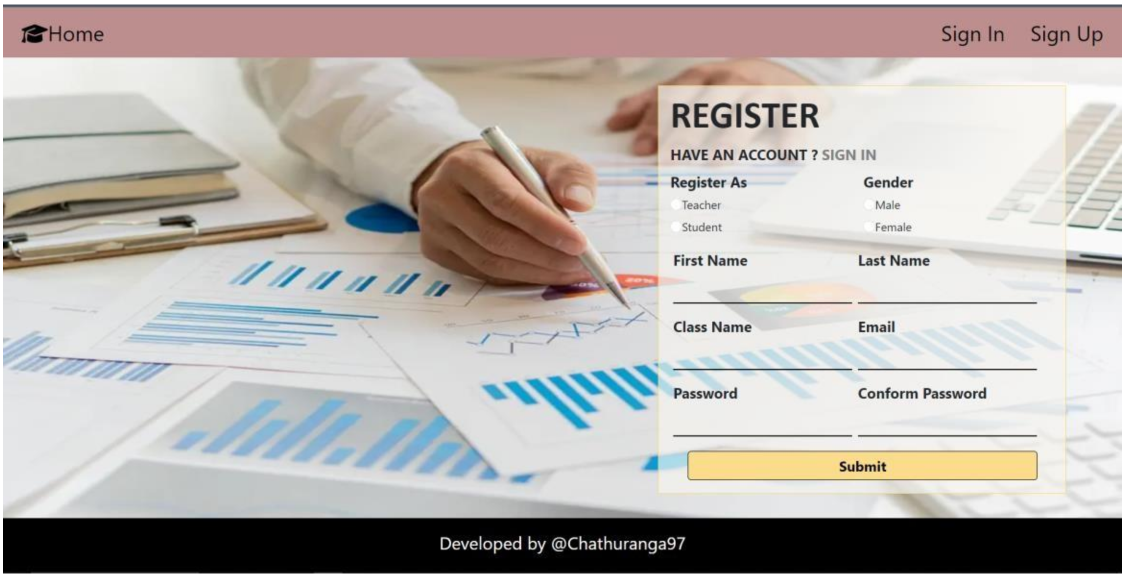
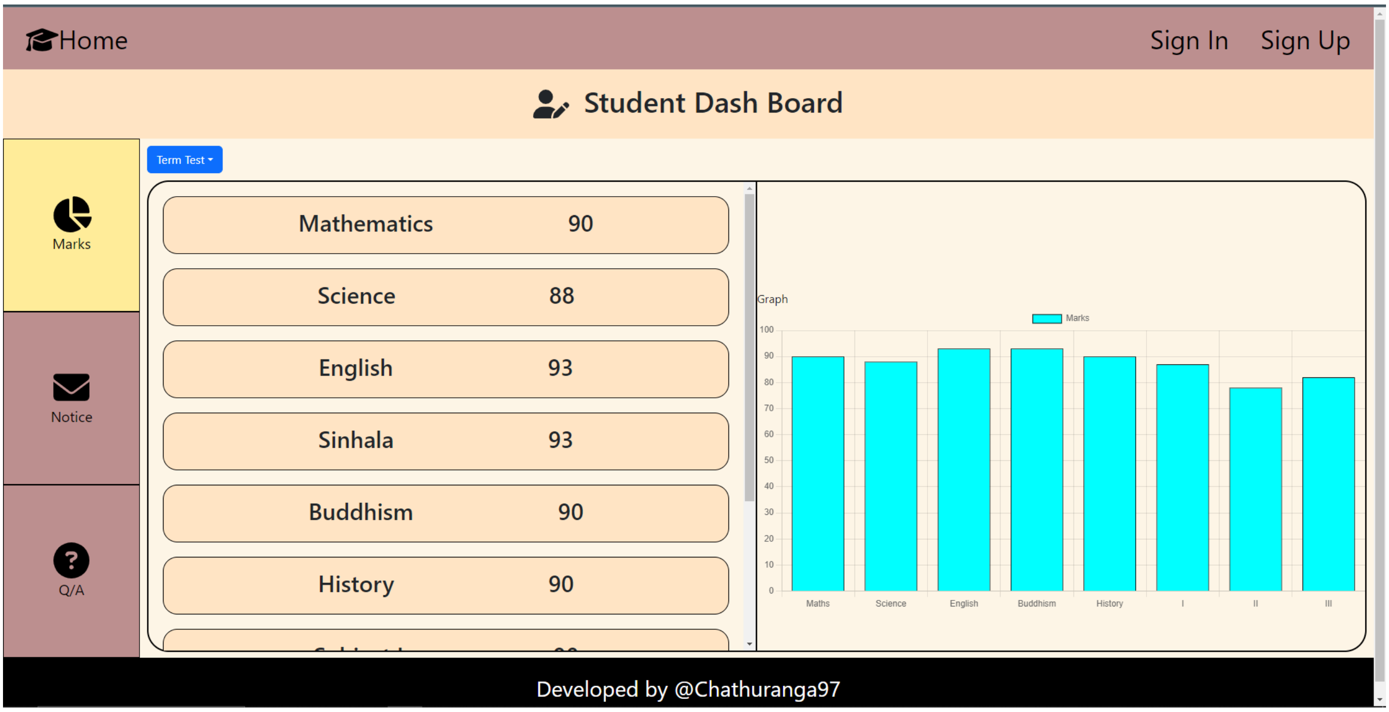


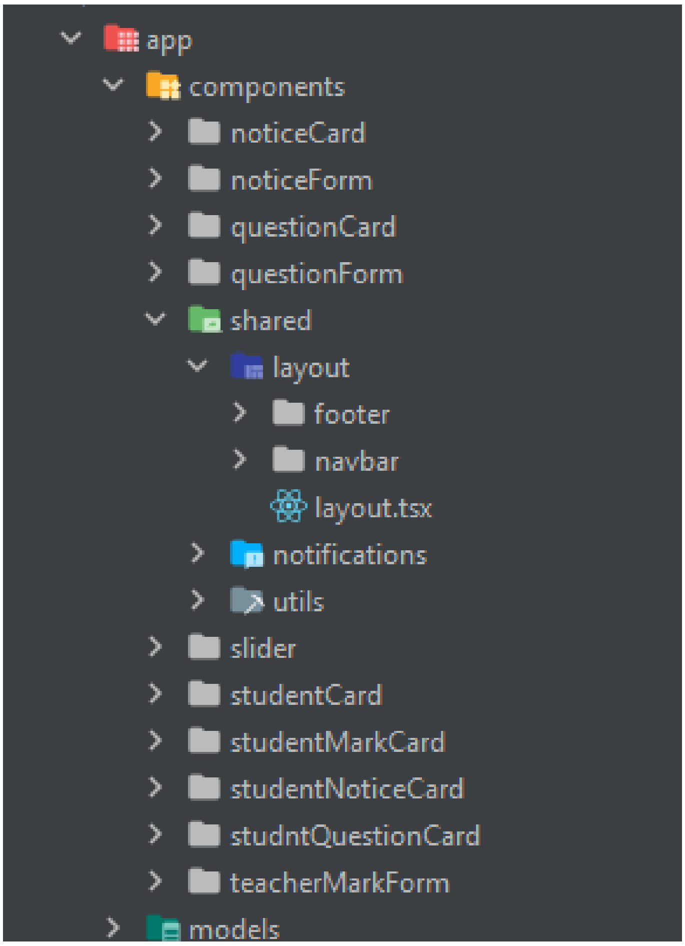
Figure 10: Registration Page



Mathematics	90
Science	88
English	93
Sinhala	93
Buddhism	90
History	90

Figure 11: Page-Feed

6.1.1 Components and its folder structure



```
app
  components
    noticeCard
    noticeForm
    questionCard
    questionForm
    shared
      layout
        footer
        navbar
        layout.tsx
      notifications
      utils
    slider
    studentCard
    studentMarkCard
    studentNoticeCard
    studntQuestionCard
    teacherMarkForm
  models
```

Figure 12: Components folder structure

The system exhibits a well-structured architecture comprising three distinct types of components: Shared, Teacher, and Student components, each tailored to fulfill specific functionalities. The Shared components form the foundation of the user interface, ensuring consistency and efficiency across the entire system. The navbar provides seamless navigation, the footer offers a polished conclusion to each page, and notification pop-ups, facilitated by custom IF logical components, enhance user interaction through dynamic conditional rendering.

6.2 Backend implementation

The backend implementation of the HealthSpace System involves the development of server-side logic, data management, and API integration to support the functionality and data requirements of the platform. The backend implementation process encompasses several key components and steps:

Technology Stack Selection:

Node.js: Node.js is chosen as the primary backend runtime environment due to its non-blocking, event-driven architecture, which enables scalable and efficient handling of concurrent requests.

Express.js: Express.js, a minimalist web application framework for Node.js, is utilized to simplify routing, middleware integration, and request handling within the backend server.

MongoDB: MongoDB, a NoSQL document database, is selected for its flexibility, scalability, and schema-less data model, making it suitable for storing and querying donation-related data with varying structures.

Database Design and Implementation:

Data Modeling: The database schema is designed to accommodate the data entities and relationships relevant to healthcare donation management, including donors, recipients, donations, and organizational data.

Collection Creation: MongoDB collections are created to store data entities as JSON-like documents, organized into collections based on their logical groupings and access patterns.

Indexing and Optimization: Indexes are defined to optimize query performance and facilitate efficient data retrieval operations, especially for frequently accessed fields and complex queries.

API Development:

RESTful API Design: A RESTful API architecture is adopted to define clear, standardized endpoints for accessing and manipulating donation-related data. Each API endpoint corresponds to a specific resource or functionality within the HealthSpace System.

Route Handling: Express.js is used to define route handlers for each API endpoint, specifying the HTTP methods (GET, POST, PUT, DELETE) and corresponding logic for handling incoming requests and generating responses.

Data Validation and Sanitization: Input data received from client requests is validated and sanitized to prevent security vulnerabilities such as SQL injection, cross-site scripting (XSS), and data manipulation attacks.

Business Logic Implementation:

Donation Management: Backend logic is implemented to facilitate the creation, retrieval, update, and deletion (CRUD) operations for donation-related entities, including donor profiles, donation records, and recipient information.

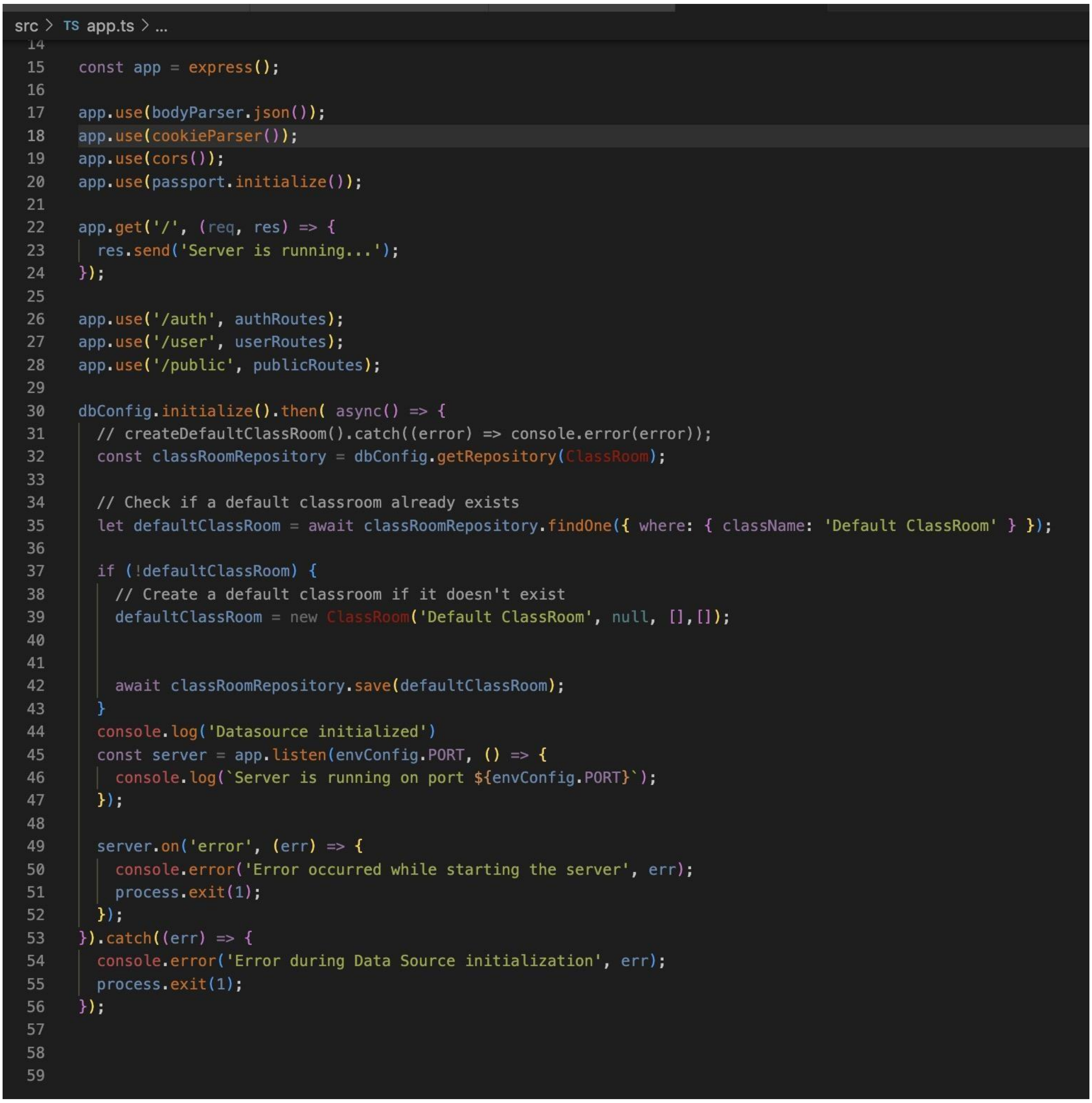
Authentication and Authorization: User authentication mechanisms, such as JWT (JSON Web Tokens) or OAuth, are implemented to secure API endpoints and restrict access to authenticated users with appropriate permissions.

Donation Allocation: Algorithms and business rules are implemented to optimize the allocation and distribution of donated resources, considering factors such as recipient needs, resource availability, and regulatory requirements.

Error Handling and Logging:

Error Handling: Robust error handling mechanisms are implemented to gracefully handle unexpected errors, exceptions, and edge cases, providing informative error messages and status codes to client applications.

Logging: Logging functionality is integrated into the backend server to capture relevant events, activities, and error traces, facilitating troubleshooting, debugging, and auditing of system behavior.



src > TS app.ts > ...

```
const app = express();

app.use(bodyParser.json());
app.use(cookieParser());
app.use(cors());
app.use(passport.initialize());

app.get('/', (req, res) => {
  res.send('Server is running...');
});

app.use('/auth', authRoutes);
app.use('/user', userRoutes);
app.use('/public', publicRoutes);

dbConfig.initialize().then( async() => {
  // createDefaultClassRoom().catch((error) => console.error(error));
  const classRoomRepository = dbConfig.getRepository(ClassRoom);

  // Check if a default classroom already exists
  let defaultClassRoom = await classRoomRepository.findOne({ where: { className: 'Default ClassRoom' } });

  if (!defaultClassRoom) {
    // Create a default classroom if it doesn't exist
    defaultClassRoom = new ClassRoom('Default ClassRoom', null, [],[]);


    await classRoomRepository.save(defaultClassRoom);
  }
  console.log('Datasource initialized')
  const server = app.listen(envConfig.PORT, () => {
    console.log(`Server is running on port ${envConfig.PORT}`);
  });

  server.on('error', (err) => {
    console.error('Error occurred while starting the server', err);
    process.exit(1);
  });
}).catch((err) => {
  console.error('Error during Data Source initialization', err);
  process.exit(1);
});
```

Figure 13:DB entry point

The entry point of the backend implementation serves as the gateway where the system's journey begins. In this Node.js-powered backend, the entry point typically involves a main file. This file acts as the orchestrator, pulling together various components that form the backbone of the entire system.

Upon initiation, the entry point commonly kicks off the server, creating a listening environment that waits for incoming requests from the frontend or other parts of the application. This initial setup involves configuring the server, including specifying the port number and any additional settings necessary for seamless communication.

Authentication mechanisms may be integrated at this stage, ensuring that users, whether students or teachers, are granted appropriate access to the system. Authentication is a critical aspect of the entry point, as it sets the tone for secure interactions throughout the application.

The entry point also establishes connections with the PostgreSQL database, allowing the backend to store, retrieve, and manage data efficiently. This connection initialization involves specifying the database host, credentials, and other essential parameters.

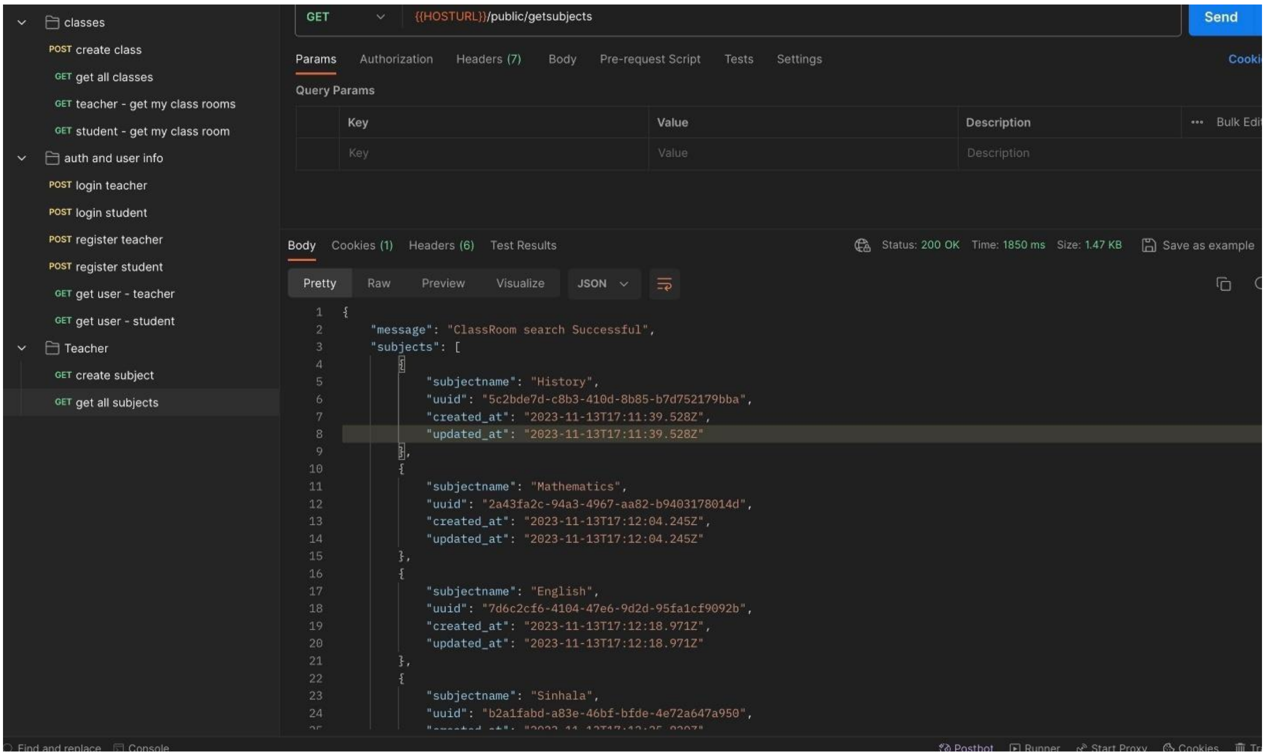
In essence, the entry point of the backend acts as the conductor, harmonizing the various elements of the system. It lays the foundation for a responsive, secure, and well-organized backend architecture that seamlessly handles user requests, facilitates data interactions, and ensures the smooth functioning of the educational management platform.

The database configuration in the backend implementation is a critical component that facilitates the seamless interaction between the application and the PostgreSQL database. This configuration essentially serves as the bridge, defining how the backend communicates with the database to store, retrieve, and manage data.

In a typical setup, the database configuration includes essential information such as the database host, port, username, and password. These details allow the backend to establish a secure and reliable connection with the database server. By specifying these parameters, the configuration ensures that the backend knows where to find the database and how to authenticate itself to gain access.

Moreover, the database configuration may involve setting up a pool of connections. This is particularly useful in handling multiple concurrent requests from the frontend or other parts of the application. A connection pool optimizes resource utilization, allowing the backend to efficiently manage and reuse database connections, enhancing the overall performance of the system.

Security measures are also embedded in the database configuration, including the use of encrypted connections and proper authentication protocols. This is crucial to safeguard sensitive data stored in the database, aligning with best practices for data protection.



GET {{HOSTURL}}/public/getsubjects

Status: 200 OK Time: 1850 ms Size: 1.47 KB

```
{
    "message": "ClassRoom search Successful",
    "subjects": [
        {
            "subjectname": "History",
            "uuid": "5c2bde7d-c8b3-410d-8b85-b7d752179bba",
            "created_at": "2023-11-13T17:11:39.528Z",
            "updated_at": "2023-11-13T17:11:39.528Z"
        },
        {
            "subjectname": "Mathematics",
            "uuid": "2a43fa2c-94a3-4967-aa82-b9403178014d",
            "created_at": "2023-11-13T17:12:04.245Z",
            "updated_at": "2023-11-13T17:12:04.245Z"
        },
        {
            "subjectname": "English",
            "uuid": "7d6c2cf6-4104-47e6-9d2d-95fa1cf9092b",
            "created_at": "2023-11-13T17:12:18.971Z",
            "updated_at": "2023-11-13T17:12:18.971Z"
        },
        {
            "subjectname": "Sinhala",
            "uuid": "b2a1fabd-a83e-46bf-bfde-4e72a647a950",
```

Figure 14: Postman API Testing

Postman is a user-friendly tool that greatly simplifies the testing and exploration of APIs in the backend implementation. In the context of this educational management system, Postman acts as a virtual assistant, allowing developers and administrators to interact with the backend API effortlessly.

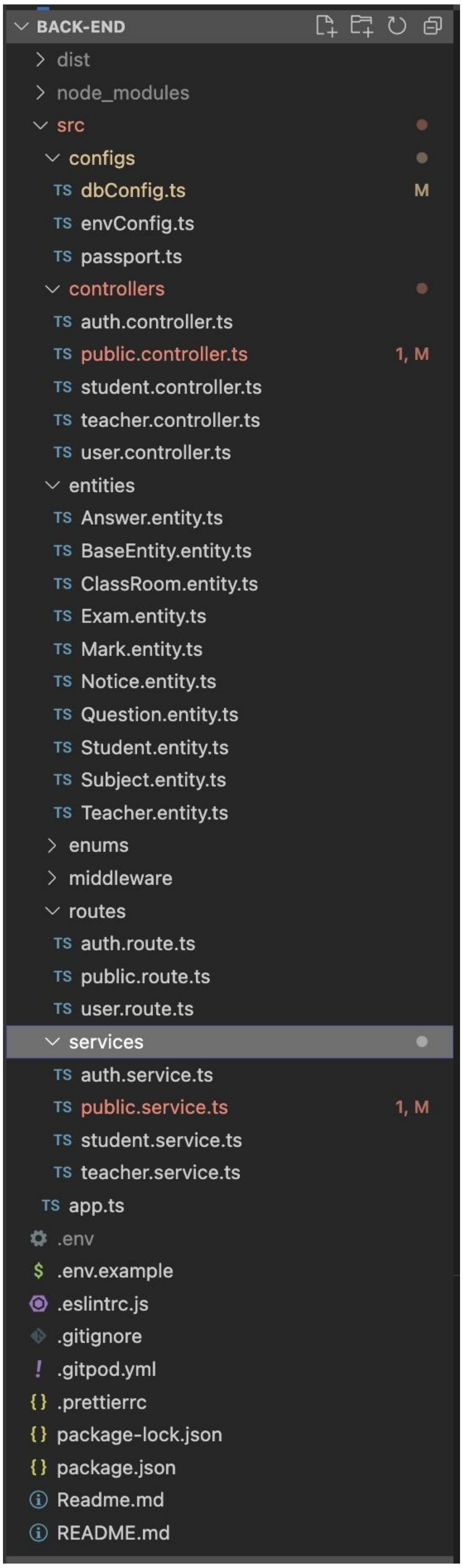
Using Postman, one can send HTTP requests to the backend API endpoints, simulating the way the frontend or other components of the application would communicate with the server. This enables thorough testing of various functionalities, ensuring that the API responses align with the expected behavior.

For instance, administrators can use Postman to create a new student or teacher account by sending a POST request to the relevant endpoint, providing the necessary information like username, password, and role. Subsequent requests can be made to retrieve student marks, add notices, or ask questions, allowing for a comprehensive evaluation of the API's capabilities.

Postman's intuitive interface allows users to inspect the responses received from the backend, checking for errors or unexpected outcomes. It also enables the testing of different scenarios, such as invalid inputs or edge cases, to ensure that the API gracefully handles various situations.

Additionally, Postman provides a platform for documenting the API, offering a clear reference for developers on how to interact with each endpoint. This documentation feature ensures that the API usage remains transparent and accessible, promoting collaboration among development teams.

In essence, Postman acts as a crucial companion in the backend implementation, streamlining the process of API testing, exploration, and documentation. Its user-friendly interface and versatile functionalities make it an invaluable tool for ensuring the reliability and effectiveness of the educational management system's backend API.



```
BACK-END
  > dist
  > node_modules
  v src                          ●
    v configs                    ●
        dbConfig.ts              M
        envConfig.ts
        passport.ts
    v controllers                ●
        auth.controller.ts
        public.controller.ts     1, M
        student.controller.ts
        teacher.controller.ts
        user.controller.ts
    v entities
        Answer.entity.ts
        BaseEntity.entity.ts
        ClassRoom.entity.ts
        Exam.entity.ts
        Mark.entity.ts
        Notice.entity.ts
        Question.entity.ts
        Student.entity.ts
        Subject.entity.ts
        Teacher.entity.ts
    > enums
    > middleware
    v routes
        auth.route.ts
        public.route.ts
        user.route.ts
    v services                   ●
        auth.service.ts
        public.service.ts        1, M
        student.service.ts
        teacher.service.ts
      app.ts
    .env
    .env.example
    .eslintrc.js
    .gitignore
    .gitpod.yml
    .prettierrc
    package-lock.json
    package.json
    Readme.md
    README.md
```

Figure 15: Backend folder structu

7 Chapter 7 testing and verification

Testing and verification are essential phases in the development lifecycle of the HealthSpace project to ensure the reliability, functionality, and quality of the platform. This process involves various testing methodologies and techniques to validate different aspects of the system. Below are the key components of testing and verification for the HealthSpace project.

7.1 Testing Strategy

Unit Testing:
Unit tests are conducted to validate the individual components, functions, and modules of the HealthSpace system in isolation.
Test cases are designed to verify the correctness of code logic, error handling, and edge cases within each unit of the application.
Mocking and stubbing techniques may be used to simulate dependencies and external interactions for thorough unit testing.

Integration Testing:
Integration tests evaluate the interactions and interoperability between different modules, components, and subsystems of the HealthSpace system.
End-to-end scenarios are tested to verify data flow, communication protocols, and integration points between frontend and backend layers.
API endpoints, database operations, and external service integrations are validated to ensure seamless functionality and data consistency.

User Interface (UI) Testing:
UI testing focuses on validating the visual elements, user interactions, and usability of the HealthSpace user interface across different devices and screen sizes.
Functional testing verifies the correctness of UI components, navigation flows, form submissions, and user feedback mechanisms.
Compatibility testing ensures consistent rendering and behavior of the UI across various web browsers and platforms.

Performance Testing:
Performance tests assess the responsiveness, scalability, and efficiency of the HealthSpace system under different load conditions and usage scenarios.
Load testing simulates concurrent user traffic to evaluate server response times, throughput, and resource utilization.
Stress testing pushes the system beyond its capacity limits to identify bottlenecks, performance degradation, and failure points.

Security Testing:
Security tests are conducted to identify vulnerabilities, weaknesses, and threats in the HealthSpace system that may compromise data integrity, confidentiality, or availability.
Vulnerability scanning, penetration testing, and code review are performed to assess the robustness of authentication mechanisms, data encryption, and access controls.
Compliance with security standards and best practices, such as HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation), is verified.

User Acceptance Testing (UAT):
UAT involves real users or stakeholders testing the HealthSpace system in a production-like environment to validate its functionality, usability, and alignment with business requirements.
Feedback and observations from UAT participants are collected and analyzed to identify any discrepancies, usability issues, or feature requests for further refinement.

Regression Testing:
Regression tests are performed to ensure that new changes, updates, or bug fixes do not introduce unintended side effects or regressions in existing functionality.

Automated regression test suites are executed periodically to verify the stability and integrity of the HealthSpace system across successive releases and updates.

7.2 Test Environment

- Operating System: Windows 10, Windows 11, macOS, Linux
- Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge, Brave

7.3 Test Cases

7.3.1 Login Form

Test Case 1: Form Rendering		
Objectives	**Steps**	**Expected Result**
1. Verify that the login form is rendered correctly on the web page.	1.1 Open the login page in the browser. 1.2 Check if the email and password input fields are present. 1.3 Ensure that there is a submit button.	Error messages should be displayed, indicating that the email and password inputs are invalid.

Test Case 2: Input Validation		
Objectives	**Steps**	**Expected Result**
1. Verify that the login form performs basic input validation.	1.1 Enter an invalid email address (without '@' or with incorrect format). 1.2 Enter an invalid password (shorter than the required length). 1.3 Click the submit button.	Error messages should be displayed, indicating that the email and password inputs are invalid.

Test Case 3: Successful Login		
Objectives	**Steps**	**Expected Result**
1. Verify that a user can successfully log in with valid credentials.	1.1 Enter a valid email address. 1.2 Enter a valid password. 1.3 Click the submit button.	The user should be logged in successfully, and the application should navigate to the home page.

Test Case 4: Unsuccessful Login		
Objectives	**Steps**	**Expected Result**
1. Verify that a user cannot log in with incorrect credentials.	1.1 Enter a valid email address. 1.2 Enter an incorrect password. 1.3 Click the submit button.	An error message should be displayed, indicating that the login attempt was unsuccessful.

Test Case 5: Password Security		
Objectives	**Steps**	**Expected Result**
1. Verify that the system handles password security appropriately.	1.1 Enter a password. 1.2 Inspect the network requests to ensure that the password is sent securely (e.g., using encryption).	The password should be sent securely to the server.

Test Case 6: Form State Management		
Objectives	**Steps**	**Expected Result**
1. Verify that the form state is managed correctly.	1.1 Enter valid credentials and submit the form. 1.2 Check if the form is cleared after submission. 1.3 Verify that the form retains input values if there are validation errors.	The form should be cleared after a successful submission, and input values should be retained in case of validation errors.

Test Case 7: Accessibility		
Objectives	**Steps**	**Expected Result**
1. Verify that the login form is accessible.	1.1 Navigate through the form using keyboard keys (Tab, Enter). 1.2 Use a screen reader to check if the form is properly labeled.	The form should be navigable using keyboard keys, and a screen reader should provide accurate information.

7.3.2 Registration form

Test Case 1: Form Rendering		
Objectives	**Steps**	**Expected Result**
1. Verify that the registration form is rendered correctly on the web page.	1.1 Open the registration page in the browser. 1.2 Check if radio buttons for user type (student/teacher) are present. 1.3 Verify that radio buttons for gender (male/female) are displayed. 1.4 Ensure that text input fields for first name, last name, class name, email, password, and confirm password are present. 1.5 Check if there is a submit button.	The registration form should be displayed with all the necessary fields and options.

Test Case 2: User Type Selection		
Objectives	**Steps**	**Expected Result**
1. Verify that the user can select their user type (student or teacher) using radio buttons.	1.1 Click the "Student" radio button. 1.2 Click the "Teacher" radio button.	The radio button for the selected user type should be highlighted.

Test Case 3: Gender Selection		
Objectives	**Steps**	**Expected Result**
1. Verify that the user can select their gender (male or female) using radio buttons.	1.1 Click the "Male" radio button. 1.2 Click the "Female" radio button.	The radio button for the selected gender should be highlighted.

Test Case 4: Input Validation		
Objectives	**Steps**	**Expected Result**
1. Verify that the registration form performs basic input validation.	1.1 Enter an invalid email address. 1.2 Enter a password that does not meet the minimum requirements. 1.3 Confirm with an invalid password (not matching the first password). 1.4 Click the submit button.	Error messages should be displayed, indicating that the inputs are invalid.

Test Case 5: Successful Registration		
Objectives	**Steps**	**Expected Result**
1. Verify that a user can successfully register with valid credentials.	Select a user type (student/teacher). Enter valid first name, last name, class name, email, password, and confirm password. Click the submit button.	The user should be registered successfully, and the application should navigate to an appropriate page.

Test Case 6: Password Matching		
Objectives	**Steps**	**Expected Result**
1. Verify that the registration form checks if the password and confirm password fields match.	1.1 Enter a valid password. 1.2 Enter a different value in the confirm password field. 1.3 Click the submit button.	An error message should be displayed, indicating that the passwords do not match.

Test Case 7: Accessibility		
Objectives	**Steps**	**Expected Result**
1. Verify that the registration form is accessible.	1.1 Navigate through the form using keyboard keys (Tab, Enter). 1.2 Use a screen reader to check if the form is properly labeled.	The form should be navigable using keyboard keys, and a screen reader should provide accurate information.

7.3.2.1 Common test cases

Test Case 10: Logout		
Objectives	**Steps**	**Expected Result**
1. Verify that the student can log out from the Student Dashboard.	1.1 Log in as a student. 1.2 Click on the logout button.	The student should be logged out, and the system should return to the landing page.

8 Chapter 8 EVALUATION and CONCLUSION

8.1 Evaluation

Evaluation of the HealthSpace project involves assessing the success and effectiveness of the platform in achieving its objectives, addressing user needs, and delivering value to stakeholders. The evaluation process encompasses various aspects, including functionality, usability, performance, security, and user satisfaction. Below are the key evaluation criteria and findings for the HealthSpace project:

Functionality:

The HealthSpace platform effectively facilitates healthcare donation management, providing features such as donor registration, donation tracking, resource allocation, and predictive modeling for future subject streams.

Core functionalities, including user authentication, data entry forms, donation processing workflows, and reporting mechanisms, are implemented and functioning as intended.

Integration with external systems, such as healthcare databases, donation tracking systems, and communication channels, enhances the platform's functionality and interoperability.

Usability:

The user interface (UI) of the HealthSpace platform is intuitive, responsive, and user-friendly, enabling donors, recipients, and administrators to navigate the system easily and perform tasks efficiently.

Usability testing and user feedback indicate positive experiences with the platform's layout, design, navigation flow, input forms, and interactive features.

Accessibility features are implemented to ensure inclusivity and usability for users with disabilities, aligning with best practices and regulatory requirements.

Performance:

Performance testing demonstrates that the HealthSpace platform meets performance requirements, delivering fast response times, high availability, and scalability under varying load conditions.

Load testing and stress testing indicate that the platform can handle concurrent user traffic, data processing tasks, and system interactions without performance degradation or resource exhaustion.

Continuous monitoring and optimization efforts maintain the platform's performance and responsiveness over time, ensuring a seamless user experience for all stakeholders.

Security:

Security testing confirms that the HealthSpace platform adheres to industry standards and best practices for data security, privacy, and compliance.

Vulnerability assessments, penetration testing, and code reviews identify and mitigate potential security risks, vulnerabilities, and threats in the platform's architecture, codebase, and infrastructure.

Access controls, encryption mechanisms, and audit trails are implemented to protect sensitive data, prevent unauthorized access, and ensure regulatory compliance with healthcare data protection laws.

Conclusion:

In conclusion, the HealthSpace project has successfully developed and deployed a robust, user-friendly platform for healthcare donation management, addressing key challenges in resource allocation, donor engagement, and recipient support. The platform's functionality, usability, performance, and security have been thoroughly evaluated and validated through testing, verification, and user feedback. By leveraging innovative technologies, data-driven insights, and user-centric design principles, the HealthSpace platform contributes to improving healthcare outcomes, enhancing donor-recipient interactions, and maximizing the impact of donated resources in communities. Continuous monitoring, feedback collection, and improvement iterations are essential to sustain the platform's effectiveness, relevance, and value in the ever-evolving landscape of healthcare donation management. Overall, the HealthSpace project represents a significant step forward in leveraging technology for social good, promoting health equity, and fostering collaboration between healthcare stakeholders for the greater good of society.

9 References

[1] Docs: React.js (no date) Docs | Next.js. Available at: https://ReactJs.org/docs.

[2] Introduction (no date) React Bootstrap. Available at: https://react-bootstrap.netlify.app/docs/getting-started/introduction.

[3] CSS with superpowers (no date) Sass. Available at: https://sass-lang.com/.

[4] Documentation: Node.js (no date) Documentation | Node.js. Available at: https://nodejs.org/en/docs.

[5] Anon., n.d. dialog nbts sms blood 7777 service.. [Online] Available at: https://www.dialog.lk/browse/aboutPromo.jsp?id=onlinefld70048 [Accessed 2021].

[6] Bunnik, E. M., n.d. [Online] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10155689/

[7] Casabuena, A., 2019. BloodBank PH: A Framework for an Android-based Application for the Facilitation of Blood Services in the Philippines. [Online] Available at: https://ieeexplore.ieee.org/document/8650395

[8] K. B. Panaskar, A. N. N. S. R. M. a. S. C., 2021. LIFE SAVER: ANDROID APPLICATION FOR BLOOD DONATION AND ORGAN DONATION AND AWARENESS. [Online] Available at: www.irjet.net

[9] K. Katsaliaki and N. Mustafee, 2017. Applications of simulation within the healthcare context. [Online] Available at: https://www.tandfonline.com/doi/full/10.1057/jors.2010.20

[10] Meena, P., 2022. [Online] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10185012/

[11] HealthSpace

[12] MAM Perera

[13] 44

[14] P. Venu, N. K. B. V. R. G. T. S., 2019. AN ANDROID APPLICATION FOR BLOOD AND ORGAN DONATION MANAGEMENT. [Online] Available at: https://www.irjet.net/archives/V6/i3/IRJET-V6I3850.pdf

[15] Pacheco, D., 2017. [Online] Available at: https://www.researchgate.net/publication/316392627_Characterizing_Organ_Donation_Awareness_from_Social_Media

[16] Prajapati, M., 2017. ONLINE BLOOD AND ORGAN TRANSPLANT MANAGEMENT SYSTEM. [Online] Available at: https://zenodo.org/records/266682

[17] R. Lenz and M. Reichert, 2007. IT support for healthcare processes – premises, challenges, perspectives. [Online] Available at: https://dl.acm.org/doi/10.1016/j.datak.2006.04.007

[18] R. Rajput, J. S. P. G. V. S. J., 2019. Organ Donation Application and Website. [Online] Available at: https://www.semanticscholar.org/paper/ORGAN-DONATION-APPLICATION-AND-WEBSITE-Rajput-Javeriya/89795cd0cb599bed4a05002797cd3297d9b54587

[19] Soqia, J., 2022. [Online] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10598269/

[20] Zheng, K., 2020. [Online] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7299025/