ENHANCING SECURITY FEATURES OF A PWA-BASED STUDENT INFORMATION MANAGEMENT SYSTEM (CASE STUDY OF COMPUTER ENGINEERING)

BY

JOEL ONOME EBIALA
ENG1503560

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF BENIN

SEPTEMBER, 2023.

ENHANCING SECURITY FEATURES OF A PWA-BASED STUDENT INFORMATION MANAGEMENT SYSTEM (CASE STUDY OF COMPUTER ENGINEERING)

BY

JOEL ONOME EBIALA
ENG1503560

A PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF ENGINEERING (B.ENG) IN COMPUTER ENGINEERING, UNIVERSITY OF BENIN, EDO STATE, NIGERIA.

SEPTEMBER, 2023.

CERTIFICATION

I hereby certify that this project **ENHANCING THE SECURITY FEATURES OF A PWA-BASED STUDENT INFORMATION MANAGEMENT SYSTEM (A CASE STUDY OF COMPUTER ENGINEERING)** for the award of B.Eng. was conducted and duly presented by Joel Onome Ebiala (ENG1503560) of The Department of Computer Engineering, Faculty of Engineering, University of Benin, Benin City has been accepted for defense.

Engr. Dr (Mrs) O. Okosun **Head of Department**	**Date**
Engr. Dr (Mrs) O. Okosun **Project Supervisor**	**Date**

DECLARATION

I hereby declare that this project **ENHANCING THE SECURITY FEATURES OF A PWA-BASED STUDENT INFORMATION MANAGEMENT SYSTEM (A CASE STUDY OF COMPUTER ENGINEERING)** was carried out by Joel Onome Ebiala (ENG1503560) and it is a record of my project work in the Department of Computer Engineering, Faculty of Engineering, University of Benin, Benin City, in partial fulfilment of a Bachelor of Engineering in Computer Engineering degree. It has not been presented before in any previous application for a bachelor's degree. References made to published literature have been duly acknowledged.

_____ _____

Joel Onome Ebiala Date

DEDICATION

I would like to dedicate this work to God Almighty, my Parents, Mr and Mrs J. Ebiala and my Siblings

ACKNOWLEDGEMENTS

I wish to express my gratitude to God almighty for his grace and guidance thus far.

I would also like to express my deepest appreciation to my project supervisor Engr. Dr. Oduware Okosun for her guidance and invaluable insights during the course of this project. Also I wish to acknowledge all Computer Engineering lectures for their time spent in teaching me, so as to gain knowledge about this field of study. I also want to thank Engr Dr. Dele Ogbede for his kind-heartedness and to my project partner Thikan Bolaiyefa Mackson whose effort and commitment to seeing this project work become perfect, I say God bless you richly.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This work is aimed at designing and implementing a Progressive Web App (PWA) based student information management system to improve the security features of the system

The methodology used in carrying out the security feature includes, the Two-Factor authentication and hashing method. The major technologies used in the implementation of the system are React.js in designing the frontend for simplicity and modularity, Node.js for the backend to support multiple synchronous activities and MongoDB for the database. The system was designed to provide a more secured method in ensuring the user's data is safe from third-party intrusion.

The results of the study showed that the PWA based student information management system was better secured in terms of accessing and retrieving data from the system, without fear of an intruder. The system provided significant improvements in the security features which enabled effective and efficient use of the system.

CHAPTER ONE

INTRODUCTION

1.1 Background of Study

Educational institutions employ computerized systems called Student Information Management Systems (SIMS) to manage student data, such as attendance, grades, and registration. The importance of these systems has grown along with the volume of data produced by educational institutions.

Controlling the processes and methods that businesses use to collect, create, arrange, distribute, and utilize information is known as information management.

In recent years, there has been a shift towards web-based SIMS, which allows administrators, teachers, and students to access information remotely using web browsers. As a result, Progressive Web Apps (PWAs) were developed as a way to create web applications that are dependable, quick to load, and visually appealing.

Online apps known as PWAs give users on any device using a web browser a native app-like experience. As with regular websites, they may be visited by a URL and are constructed with contemporary web technologies like HTML, CSS, and JavaScript. Push notifications, offline functionality, and the ability to be linked to mobile devices' home screens are just a few of the benefits PWAs have over standard web applications. PWAs are the perfect medium for distributing SIMS because of these characteristics.

The Progressive Web App Student Information Management System, or PWA-SIM, is an online tool that combines the advantages of PWAs and SIMSs. PWA-SIM can be accessed from any device with a web browser and provides a seamless user experience. It allows

students to access their grades, attendance records, and other academic information, while also enabling teachers and administrators to manage student records and generate reports. PWA-SIM has the potential to revolutionize the way educational institutions manage student information.

While PWA-SIM offers many benefits, it is important to ensure that it is secure and protected against threats such as data breaches and cyber-attacks. This research project focuses on enhancing the security features of the PWA-SIM. Enhancing PWA-SIM's security and resilience can be achieved through the use of web application firewalls, technology for detecting and preventing intrusions, secure coding techniques, two-factor authentication, frequent security audits, and penetration testing. This research project has significant implications for educational institutions that use SIMS and can serve as a basis for future research on securing web-based applications.

Numerous security and privacy flaws in the expanded capabilities of PWA are examined in a paper by Lee *et al.* (2018). One of those weaknesses relates to the way push notifications are shown in browsers. The authors clarify that even if the vulnerability were resolved, the sender might still be spoof since little to no authenticity checks are made. They further describe how the sender's low visibility in a notice makes them vulnerable to being utilized for phishing attacks. Using the name and logo of the company in the notification is all that is needed to pull off this phishing, which is the process of fooling the user into believing that an attacker is a legitimate business.

When this happens, a gullible user may believe it to be real and divulge information that could be harmful. (Lee *et al.* 2018).

Another work by Subramani *et al.* (2020) delves deeper into the topic of push notifications in PWA. The authors developed a method named PushAdMiner to measure the quantity of push notifications being misused by advertisements and the proportion of fraudulent advertising. In the end, Subramani *et al.* (2020) examined 21541 push notifications, of which 5143 were advertisements that were sent out from 572 ad campaigns. The authors found that 51% of those 5143 advertisements were malevolent.

1.2 Problem Statement

Creating a Student Information Management System with a PWA Foundation for UNIBEN is a laudable initiative; however, the lack of robust security features exposes the system to various cyber threats, giving the availability, security, and integrity of the sensitive data stored in the system a serious risk. This concern is buttressed by the increasing incidence of cyber security attacks on educational systems worldwide, as reported by Kouicem*et al.* (2019). It is, therefore, essential to address this issue by implementing enhanced security measures to safeguard the system against possible cyber threats. However, the security measures must not compromise the system's performance and usability, as emphasized by Zhou *et al*. (2020).

1.3 Aim of the Study

The project's goal is to enhance the safety of the PWA-based Student Information Management System developed for the University of Benin- using hash function and two-factor authentication

1.4 Objectives of the Study

The goal of this research is to improve the security of the student information management system (SIM) built for the University of Benin (UNIBEN) using PWA technology by:

1. Putting strong security measures in place to safeguard students' private information and sensitive data, such as authorization and authentication processes.
2. Additionally, the implementation of these features should not compromise the performance and usability of the system.

1.5 Scope of Work

The scope of this study is to identify the security loopholes in the existing PWA-based student information management system and to implement enhanced security measures to mitigate potential risks. The study will cover various aspects of security, including but not limited to, authentication, authorization, encryption, data integrity, and access control. To verify the system's resilience, the suggested security measures will be put to the test and assessed against well-known cyberthreats including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). The study will focus solely on the PWA-based student information management system developed for UNIBEN and will not cover other systems or applications.

1.6. Significance of Work

The result of this work will improve the security features of the student information management system, making it secured and free from intrusion, and safeguarding the data of its users.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview of Information Management

Information is an organizational resource while managing information is an organizational strategy for proper information and data handling in an organization or institution. The deliberate procedure for assembling and using data to bolster decision-making at all organizational levels can be understood as information management (IM). Opoku, in 2015 carried out a Evaluation of the Literature on "Information Management and Organizational Performance" to understand existing knowledge and practices, find any holes, and argue for more research in the field of information management. According to Opoku, the purpose of information management is to create value for people, organizations, communities, and society at large by intelligently and creatively planning, creating, managing, and utilizing information.

2.2 What is Information Management System?

An electronic document tracking and storage system, or an electronic document storage system comprising image files of paper documents, is called an information management system (IMS). History tracking, or tracking many versions made by various users, is typically another feature it can do.

An information management system (IMS) is a whole suite of software, procedures, and hardware used for data collection, archiving, and management. Improving decision-makers' access to information is the aim of an information management system (IMS).

Financial services and healthcare enterprises alike can use IMS software. Updated frequently to stay abreast of information management trends, it can be used for anything from customer data management to patient record management.

2.3 Benefits of Information Management Systems

1- Increase productivity

An information management system is a necessary component of every kind of enterprise. All crucial information can be arranged and kept in one place by using an information management system. Employee productivity, efficiency, and success are all increased as a result.

2- Improve communication

An IMS that can monitor and manage the data flow inside an organization is crucial. It will support better customer, staff, partner, and supplier communication. Teams can prevent miscommunications and incomplete information that could lower customer satisfaction by putting in place an efficient framework. It will also facilitate the fastest possible search for the information you require.

3- Reduce errors

The deployment of a successful information management system has several advantages. By assisting staff in finding the appropriate file or document for their needs, it helps lower error rates. It also helps with the numerous hours of paperwork required at an organization. An agency can utilize the system to help find trends or patterns in its data when it puts in place a procedure for monitoring and keeping track of information. They can see what is and is not working well thanks to this. Additionally, it enables them to fix processes as needed or learn from missteps.

4- Increase competitive advantage

The success of businesses in the digital age is greatly influenced by the way they handle and store data. According to a Forrester Research report, any firm must have a "data-driven" information management strategy in this day of expanding customer expectations and intensifying competitive challenges.

5- Cost savings

Setting up a productive information management system has several advantages. Saved costs are one advantage. Data maintenance and archiving might be costly, but they can be significantly decreased with the appropriate solution.

2.4 Key Features of Information Management Systems

1- Data security

Any information management system must have data security. To guard against the theft or illegal access, use, or destruction of sensitive data, it entails a variety of tactics and procedures, including both preventive and investigative methods.

To protect sensitive data, a corporation with more than 500 employees should have distinct security controls at the system and user levels, as well as risk management overseen by the chief information officer (CIO).

2- Data governance

There are several ways in which an information management system might support information governance. Maintaining accurate and current data can be aided by tracking and managing it. Additionally, it can be useful to audit the data to make sure compliance needs

are met. Controlling who has access to the data and what they can do with it can also be beneficial.

3- Data analytics

Data analytics ought to be the primary component of every information management system. The process of gathering unprocessed data, compiling it, and forecasting its contents is known as data analytics. Customers are able to make more accurate decisions as a result of having a better knowledge of their data.

4- Data integration

By offering a central store for data that can be accessed and shared by numerous applications, an IMS can aid in data integration. To make sure the data is compatible with the apps that will use it; the system can additionally offer tools for data transformation and purification.

5- Big data capabilities

Big data capabilities should be available in an IMS for several reasons. Firstly, big data may give firms more precise and comprehensive information, which can aid in better decision-making. Furthermore, by giving businesses a deeper understanding of their internal business procedures, big data can aid in operational optimization. Lastly, by offering more extensive security and privacy controls, big data may assist businesses in safeguarding their information assets.

2.5 Characteristics of Good Information Management Systems

Samer&Rawan, (2018) in their research highlighted the basic characteristics of a good or effective information management system:

1. Quality

2. Flexibility

3. Timely

4. Accessibility

2.6 Student Information Management Systems

A student information management system is a piece of data management software that assists educational institutions in digitizing and managing student information. The technology may capture school-specific data and store it online for easy access by students, instructors, parents, and school officials. Student management software includes a variety of modules that assist in tracking, storing, and managing student and school data, such as student registration information, financial management, grades, attendance, and activit y records. Such complete data may be easily shared with many users to guarantee that school administration and student activities run smoothly, methodically, and without error.

2.6.1 Benefits of Student Information Management System

Student information systems allow lecturers and school administrators to store and manage data in a single area. It maintains track of every student activities and increases an organization's efficiency. Every educational institution needs a powerful and secure student management system to overcome the complexities of traditional student information management and manage all duties with maximum efficiency. Some of the benefits include:

1. **Better Student Evaluation** - Student management systems collect data in real time and use analytics to evaluate each student's performance. With such evaluation techniques, instructors and administrators may make changes to better serve their pupils. The system also allows parents to view their children's school activities, such as assignments, examinations,

results, and grade books. It also allows students to monitor and track their records in order to gain insight into how to improve their performance.

2. **Effortless Communication** - The program facilitates effective communication among teachers, students, non-teaching personnel, parents, and school administration. Most school management systems include discussion boards and communication channels such as video conferencing, audio conferencing, SMS, automatic emails, instant messaging, quick alerts, and so on, allowing for seamless contact between users. An intuitive solution like this facilitates involvement and boosts transparency, which leads to more productivity and quicker communication.

The management of the school will automate time-consuming tasks by using a student information system. For administrators, it is an economical and efficient way to manage their daily tasks. The following enumerates a few benefits of utilizing a student management system:

- Data privacy and sensitive information security Office task automation

- Easy-to-use platform with role-based access and authenticated profiles

- Simplified admissions, administrative processes, and non-administrative duties

- Communication is open across all departments

- Complete application control

- Records are available 24 hours a day, seven days a week. Centralized data storage

2.7 Security of Students Information Management System

With the advancement of social science, technology, and education in the modern era, learning and instruction are growing increasingly sophisticated and enlightening. The use of contemporary information technology in scientific research and teaching management is known as educational informatization, and it emphasizes the creation and deployment of educational information resources. The effectiveness of the teaching process as well as the capacity to organize and distribute educational resources have both significantly increased thanks to educational intelligence and informatization, which consistently supports the advancement of education. However, a lot of data regarding education, instruction, and student privacy will be included in the process of digitizing teaching and managing educational resources. Inadequate management and storage of the data will result in the disclosure of sensitive student privacy information as well as critical information used in the teaching and learning process. This will be extremely detrimental to the teaching and learning process as well as the privacy protection of the students. It will also potentially jeopardize social security and education as a whole. Thus, there is a pressing need for a system technology that integrates security, dependability, and privacy protection in a modern culture where privacy protection issues are becoming more and more important. It maintains and safeguards sensitive student privacy data as well as critical education and teaching data in a way that is reasonable throughout the process of education and teaching informatization.

2.8 Web Application Vulnerabilities

The exponential rise in web application deployment has led to increasingly scattered, complicated, and difficult-to-secure IT infrastructures. In order to safeguard their IT infrastructures, businesses have relied on network perimeter security mechanisms like firewalls for over ten years. But as attacks increasingly focus on web application security

issues, like injection flaws, typical network security protection might not be enough to protect apps from these kinds of threats. A web vulnerability is a flaw or configuration in the coding of a website or online application that allows an attacker (hacker) to take over the website and maybe the hosting server.

According to Open Web Application Security Project OWASP (Smithine, Stock, Gigler, &Glas, 2017) top 10 Application security risks as of 2017 are listed below.

i. **Injection**: Sending untrusted data to an interpreter as part of a command or query can lead to injection vulnerabilities including SQL, OS, and LDAP injection. The interpreter can be tricked into executing unwanted commands or gaining unauthorized access to data by the 11 attackers' malicious data.

ii. **Broken Authentication**: Authentication and session management operations within applications are frequently implemented improperly, which leaves room for attackers to get credentials, keys, or session tokens through compromise or to take advantage of other technical defects that allow them to temporarily or permanently assume the identities of other users.

iii. **Sensitive Data Exposure**: Sensitive data, including PII, financial, and medical information, is often not adequately protected by online apps and APIs. Such inadequately protected data may be stolen or altered by attackers in order to commit identity theft, credit card fraud, and other crimes. Extra security measures, like encryption while in transit or at rest, and extra care should be taken when exchanging sensitive data with the browser.

iv. **XML External Entity (XXE)**: External entity references in XML documents are evaluated by a large number of outdated or improperly configured XML processors. Internal port scanning, internal SMB file shares on unpatched Windows servers, remote code execution, file URI handlers, and denial-of-service assaults like the Billion Laughs attack can all be leveraged by external organizations to reveal internal files.

v. **Broken Access Control**: Limitations on the actions that authorized users can take are not adequately implemented. Attackers can utilize these vulnerabilities to get access to unauthorized features and/or data, including the ability to view private files, alter other users' data, alter access rights, and access other users' accounts.

vi. **Security Misconfiguration**: Insecure default configurations, open S3 buckets, misconfigured HTTP headers, error messages containing sensitive information, and not patching or upgrading systems, frameworks, dependencies, and components in a timely manner (or at all) are some of the causes of security misconfiguration, which is the most frequent issue in the data.

vii. **Cross-Site Scripting (XSS)**: Whenever an application uses a browser API that generates JavaScript to update an existing web page with user-supplied data or incorporates untrusted data into a new page without sufficient validation or escaping, an XSS vulnerability results. XSS gives hackers the ability to run scripts in the victim's browser that can take control of user sessions, alter websites, or send users to dangerous websites.

viii. **Insecure Deserialization**: When an application gets hostile serialized objects, vulnerabilities in secure deserialization arise. Remote code execution results from

insecure deserialization. Serialized objects can be replayed, altered, or removed to carry out injection attacks, escalate privileges, and spoof users, even in cases where deserialization issues do not lead to remote code execution.

ix. **Using Components with Known Vulnerabilities**: Libraries, frameworks, and other software modules are examples of components that operate with the same permissions as the application. An attack of this nature has the potential to cause significant data loss or server takeover if a vulnerable component is used. Application defenses may be compromised and a variety of attacks and effects may be enabled by applications and APIs that use components with known vulnerabilities.

x. **Insufficient Logging & Monitoring**: The lack of adequate logging and monitoring, in conjunction with inadequate or nonexistent incident response integration, permits 13 attackers to continue attacking systems, persist, switch to different systems, and alter, remove, or delete data. According to the majority of breach studies, it takes more than 200 days to find a breach, which is usually discovered by outside sources rather than internal procedures or monitoring.

2.8.1 Web Application Security

Since dynamic online applications have evolved, web application security has become a significant challenge in the field of information technology. According to (Yadav, 2014), some vulnerabilities come from network-layer insecurity, session-less protocols, the overall complexity of web technologies, and untrusted client access points. Client software for web applications is typically not always under the application owner's control. As a result, input received directly from a

client using the software cannot be entirely trusted. An attacker can replicate a user's identity, produce fraudulent messages and cookies, or spoof an identity to appear like a valid client. Furthermore, because HTTP is a session-less protocol, replay and injection attacks can be easily launched against it. Messages sent over the Hypertext Transport Protocol are easily intercepted, spoofable, and altered.

2.9 Technologies Used

2.9.1 Node.js

I/O-intensive online applications, such chat apps and multimedia streaming websites, are mostly created with Node.js, a JavaScript-based framework. Powered by the V8 JavaScript engine, it runs on Google Chrome. A client browser that accesses all of the application's resources via the internet renders software that runs on a server called a web application.

A typical web application consists of the following components:

- **Client:** The user who sends requests to the server to communicate with it is referred to as a client.
- **Server:** It is the server's responsibility to receive requests from clients, process them as needed, and provide the customers with the results. It enables customers to manipulate the data by acting as a link between the front-end and the stored data.
- **Database:** The data used by a web application is kept in databases. The data can be created, edited, and removed at the client's discretion.

VPS servers provide the foundational features and setting needed to combine developer tools and APIs with Node.js applications. With Hostinger's VPS, you get far more value for your money and increased control and flexibility over your hosting environment. It includes a Node.js template built for Ubuntu 22.04 that includes Node.js. This makes

getting started really simple and quick. Additionally, OpenLiteSpeed server is included. Additionally, they provide CloudPanel templates, which facilitate the building of Node.js apps and facilitate their startup and management. Even if you have no prior VPS expertise, you can rapidly become familiar with everything thanks to the sleek, user-friendly UI.

2.9.2 Node.js Server Architecture

A "Single Threaded Event Loop" concept is used by Node.js to handle several concurrent clients. The Node.js Processing approach uses the JavaScript callback mechanism in addition to the JavaScript event-based approach. Two basic concepts are utilized by it:

1. The model is asynchronous
2. I/O activities are not blocked

These capabilities improve Node.js web applications' throughput, performance, and scalability.

2.9.3 Components of the Node.js Architecture

- **Requests:** The type of requests a user makes to the server might be either non-blocking (simple) or blocking (complicated), depending on what has to be done.
- **Node.js Server:** Users submit queries to the Node.js server, which then handles them and responds with results.
- **Event Queue:** Event Queue is primarily used to hold incoming client requests and forward them to the Event Loop one after the other.
- **Thread Pool:** A Node.js server's Thread pool holds the threads that are available to carry out the tasks needed to handle requests.

- **Event Loop:** The Event Queue transmits requests to Event Loop, which then responds to the clients.
- **External Resources:** External resources are employed to address blocking client requests. Any kind of them is possible (computation, storage, etc.).

2.9.4 Workflow of Node.js Server:

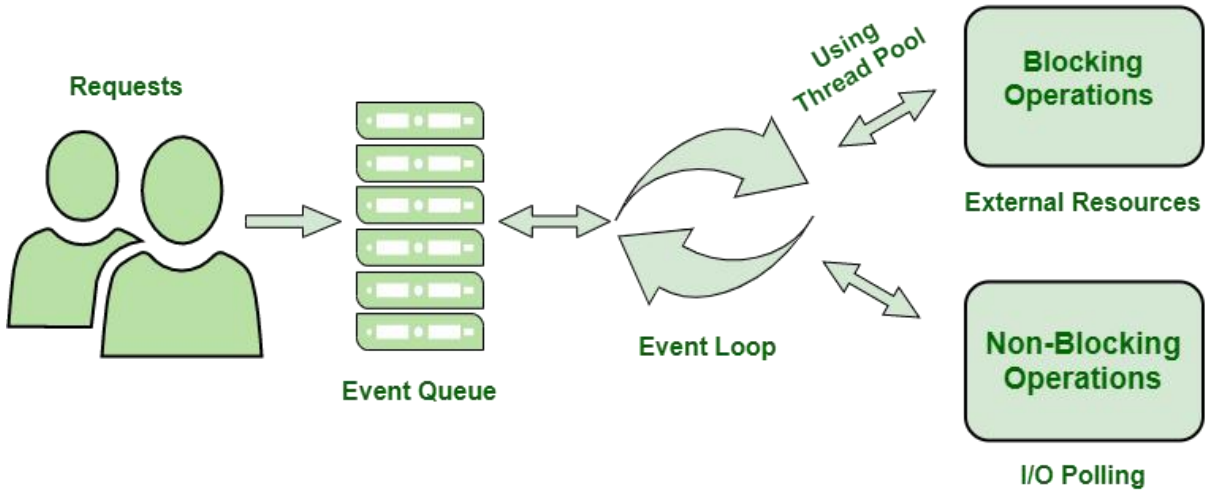

Figure 2.1: Workflow diagram of Node.js Server

• To perform operations, users submit requests—blocking or non-blocking—to the server.

• At the server-side, the requests go straight into the Event Queue.

*Requests are sent in sequence to the event loop via the event queue. The event loop determines if the request is blocking or not.

• Event Loop replies to the relevant clients after processing non-blocking requests that don't call for outside resources.

• When a request is blocked, one thread is tasked with leveraging external resources to finish the task.

• Following the operation's conclusion, the request is routed to the Event Loop, which provides the client with the response.

2.9.5 Advantages

• Event Queue and Thread Pool enable the Node.js server to effectively manage large volumes of requests.

• Since Event Loop handles each request one at a time, it is not necessary to create many threads; one thread will suffice.

• Because each request is handled individually, sending them to a Node.js server uses less memory and server resources overall.

2.10 React.js

A JavaScript framework and library created by Facebook, React.js is available as open source software. By using a lot less code than you would with vanilla JavaScript, it's utilized to swiftly and effectively create interactive user interfaces and web apps.

By building reusable components—which you might imagine as separate Lego blocks—you build applications with React. Collectively, these constituents comprise the application's user interface, which is comprised of separate parts.

In an application, React's main function is to manage the view layer by offering the optimal and most effective rendering execution, much like the V in a model-view-controller (MVC) paradigm. React.js encourages developers to break down these intricate user interfaces into discrete, reusable components that serve as the framework for the entire UI, as opposed to

handling the entire UI as a single entity. In order to generate web pages more quickly and provide highly dynamic and responsive online applications, the React framework does this by fusing the speed and efficiency of JavaScript with a more effective way to manipulate the DOM.

2.11 MongoDB

The well-liked NoSQL document-oriented database system MongoDB is made to be simple to use, adaptable, and scalable. MongoDB is the best option for managing unstructured and semi-structured data since it stores data in documents that resemble JSON, unlike conventional relational databases.

2.11.1 Features of MongoDB

1. **Document-oriented data model**: Documents, which can have a variety of forms and be nested to depict intricate relationships, are how MongoDB keeps data. As a result, complicated joins are not necessary, making data storing and retrieval easier.

2. **Scalability**: Large data loads and heavy traffic loads can be handled by MongoDB because of its horizontal scalability across several servers.

3. **High availability**: With its integrated replication and automated failover capabilities, MongoDB makes sure that your data is always accessible, even in the case of hardware malfunctions.

4. **Flexible indexing**: MongoDB facilitates easy query optimization for various use cases by offering a wide range of indexing options, such as hashed indexes, geospatial indexing, and text search.

5. **Aggregation framework**: With MongoDB's robust aggregation structure, you can operate on enormous data sets and execute intricate queries and data analysis tasks.

6. **Sharding**: In order to increase performance and scalability, MongoDB enables sharding, which divides data among several servers.

7. **Native support for JSON**: MongoDB natively supports JSON, which makes it easy to work with modern web and mobile applications that use JSON as their data interchange format.

8. **ACID transactions**: MongoDB supports ACID transactions, ensuring that your data remains consistent even in the event of concurrent writes.

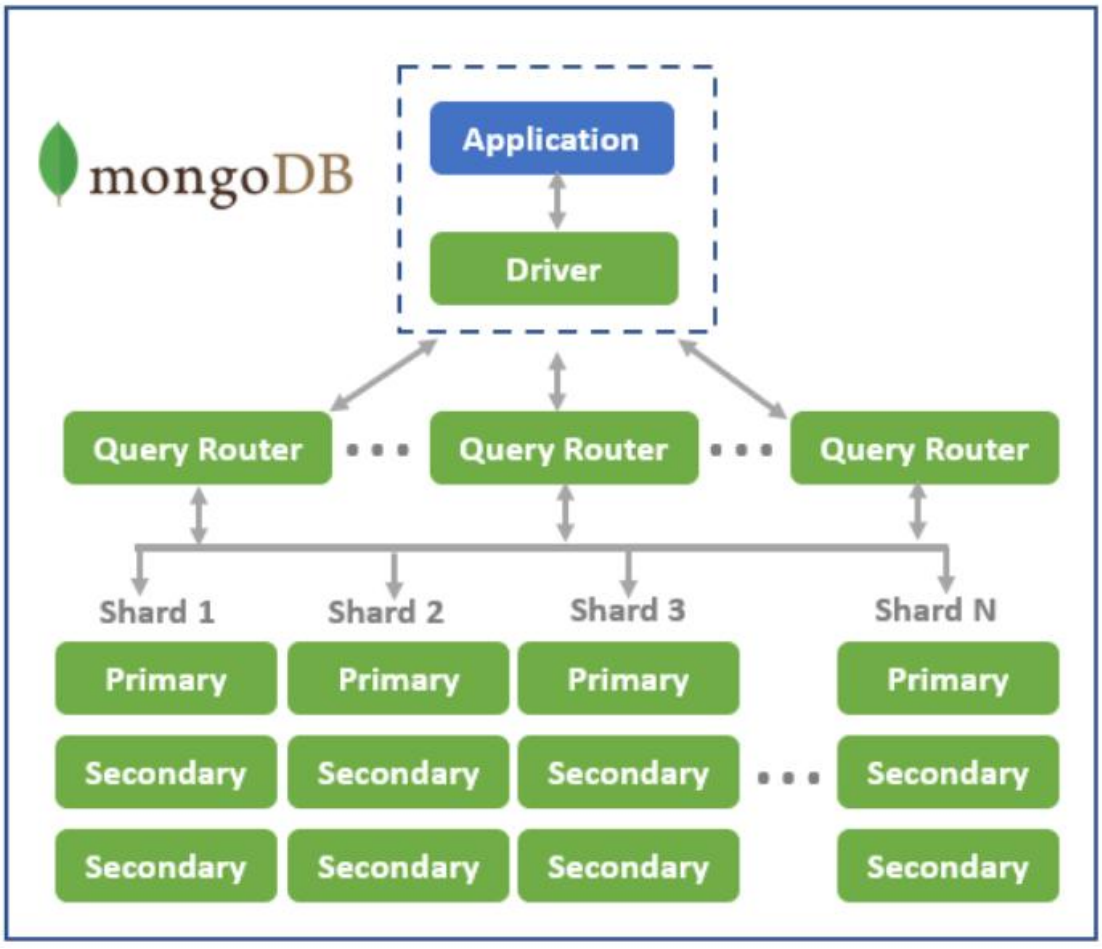

Figure 2.2:ACID transactions

2.12 Express.js

With its foundation in Node.js, Express.js is a JavaScript framework that facilitates server-side and user-side development. The Node.js web framework Express is incredibly quick, necessary, aggressive, and mild. Express is essentially a layer that has been added to Node.js to assist with server and route management. For creating both web and mobile applications, it offers a powerful feature set. This is a minimal node.js framework. To construct our applications, Express offers a simple interface. The resources needed to construct our app are provided by it. The node package manager (npm) offers a variety of modules that may be integrated with Express.js, making it highly adaptable.

2.12.1 Features of Express

1. Web applications with one page, multiple pages, and hybrid designs can be created using Express.
2. Express makes it possible to put up middleware to reply to HTTP requests.
3. Express provides a routing table that is utilized to carry out various operations according to the URL and HTTP method.
4. By providing arguments to templates, Express enables the dynamic rendering of HTML pages.

2.13 Related Works

Numerous security and privacy flaws in the expanded capabilities of PWA are examined in a paper by Lee *et al.* (2018). One of those weaknesses relates to the way push notifications are shown in browsers. The authors clarify that even if the vulnerability were resolved, the sender might still be spoof since little to no authenticity checks are made. They further describe how the sender's low visibility in a notice makes them vulnerable to being utilized for phishing attacks. Using the name and logo of the company in the notification is all that is needed to

pull off this phishing, which is the process of fooling the user into believing that an attacker is a legitimate business. When this happens, a gullible user may believe it to be real and divulge information that could be harmful. Lee *et al.* (2018)

Another work by Subramani *et al.* (2020) delves deeper into the topic of push notifications in PWA. The authors developed a method named PushAdMiner to measure the quantity of push notifications being misused by advertisements and the proportion of fraudulent advertising. In the end, Subramaniet al. (2020) examined 21541 push notifications, of which 5143 were advertisements that were sent out from 572 ad campaigns. The authors found that 51% of those 5143 advertisements were malicious.

Ad campaigns may target the notification system specifically because of this. Subramani *et al.* (2020) also discovered that ad-blocking programs and blacklists were ineffective at preventing malicious use of push notifications, in part because the ad-blocking services were unable to access service workers.

Lee *et al.* (2018) discovered an additional problem: employees providing services could be exploited to steal information from users' devices, such as cryptocurrency mining. According to the research, this is made possible by the fact that service workers can function silently in the background and are automatically installed when users browse websites. After a whole day of experimentation, Lee *et al.* (2018) were able to make a tiny profit. If an assault of this kind were to infect hundreds of devices, a tiny amount of money may be created continuously (Lee *et al.* 2018). Additionally, it would significantly shorten the devices' battery life and performance (Lee *et al.* 2018).

In addition, Lee *et al.* (2018) detail a technique for getting a user's history by loading numerous pages inside iframes in the hopes that some may load when the user is not connected to the internet. In order for this to function, as described in the study, it basically

needs to brute force every website that exists. This is a privacy violation even though it would be a laborious procedure that needs to be targeted at particular websites in order to be successful.

CHAPTER THREE

METHODOLOGY

3.1. The Design of a PWA Student Information Management System.

The project approach, which includes the Software Development Life Cycle (SDLC), is explained in this chapter. To provide more detailed explanations, Unified Modeling Language (UML) tools like the class and activity diagrams were used.

3.2 System Design

Methodology encompasses a series of phases that assist systems developers in selecting appropriate techniques for each project stage. It is essential for organizing, supervising, regulating, and assessing projects or systems. In the context of information systems, It is made up of a number of steps, instruments, methods, and documentation tools that make putting in place a new information system easier. For this project, the chosen methodology is Object-Oriented Analysis and Design (OOAD) due to its alignment with the project's nature.

3.3 Architectural Design

The Object-Oriented Analysis and Design (OOAD) software development technique is used in this project. The two primary phases of OOAD, which sees a system as a collection of interacting objects, are Object-Oriented Analysis and Object-Oriented Design.

Object Modeling, a component of this approach, shares similarities with traditional system design but employs a different approach. The key steps involved in system design using Object Modeling typically include:

1. System Analysis
2. System Design

3. Object Design
4. Implementation

3.3.1 System Analysis Phase

Examining the current system is the aim of this step. To put it another way, it entails gathering and analyzing data regarding the existing system in order to have a thorough grasp of it. The objective is to diagnose any issues and utilize this knowledge to enhance the system by implementing improved procedures and methods. This stage is very important in system development since it helps make the new system a reality to a large extent. These studies are essential because they provide an overview of the kinds of data that will be input, produced, and processed by the system in the future, which is needed during the system design phase.

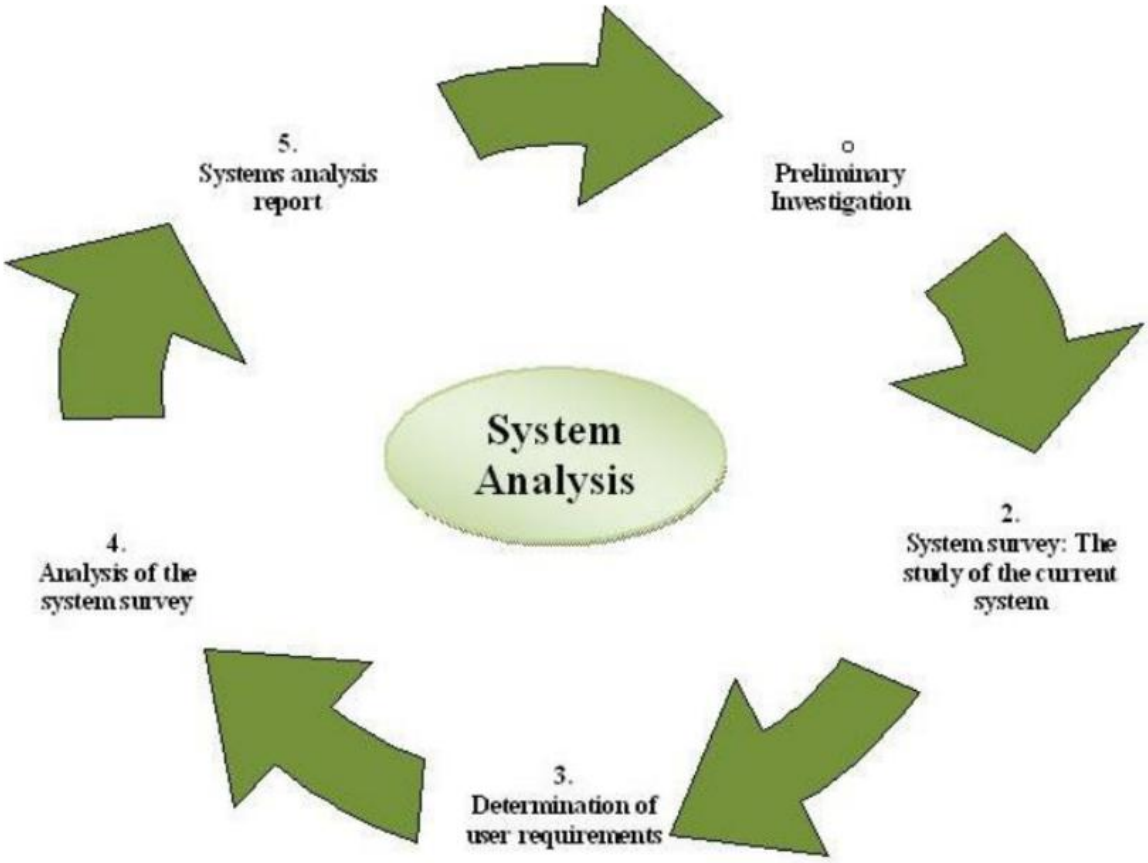

Figure 3.1: System analysis phase diagram

3.3.2 Requirements Modeling

In this stage, the process includes gathering information to characterize the existing system and identifying the needs and specifications for the new system. This encompasses aspects such as outputs, inputs, processes, performance, and security.

3.3.3 Current System Overview

The existing system uses a paper-based approach to gather information and starts with the manual registration of new employees and pupils. Student course allocations for each semester also rely on a paper-based approach. Course assignments to lecturers are managed by the Department Head. Lecturers manually input student attendance and grades into Excel spreadsheets, with validations performed by the user. This process entails several risks and substantial manual effort, necessitating user vigilance when entering data into MS Excel.

3.3.4 Challenges in the Current System

An information system based on paper requires a lot of work to manage and track. Accessing, editing, and storing paper records require physical labor that provides little value. Moreover, the physical space required to maintain paper records leads to data redundancy (duplicate records in separate locations) and data inconsistency (the same student's records appearing in multiple Academy departments/units simultaneously). Notifications intended for intended recipients may be delayed when posted on notice boards to distribute information to students.

3.3.5 New System Requirements Assessment

To initiate the development of the new system, it is imperative to conduct a comprehensive requirement determination process. This entails a thorough examination of the existing system to gain insights into its functionality and identify areas ripe for improvement. A requirement, in this context, refers to any essential attribute or functionality that must be

incorporated into the new system to enhance the situation as it is right now, which may involve data capture and processing methods, information generation, and more. The efficacy of the new system hinges on the accuracy and completeness of the requirements gathered. To attain this, a combination of interviews was employed as a means to gather valuable insights and details about the framework.

3.3.6 Object Modeling in System Analysis

'Objects' are the means by which data and related processes are integrated in the field of object-oriented analysis (O-O). These objects are simulations of real-world entities, which include people, physical objects, transactions, and events that have an impact on the system's functionality (Shelly & Rosenblatt, 2009). Throughout the course of system development, analysts frequently employ a blend of modeling techniques to extract comprehensive insights. In the context of this project, the principal objects encompass:

i. Staff
ii. Student
iii. Admin
iv. Courses

3.4 Use Case Diagram

Users, sometimes called actors in your system, and their interactions with it can be summarized using a use case diagram in the Unified Modeling Language (UML). A collection of certain symbols and connectors are needed to construct one. Your team can debate and express the following with the aid of an efficient use case diagram:

• Situations where people, organizations, or external systems engage with your system or application;

• Objectives that your system or application assists those entities (referred to as actors) in achieving

• How big your system is

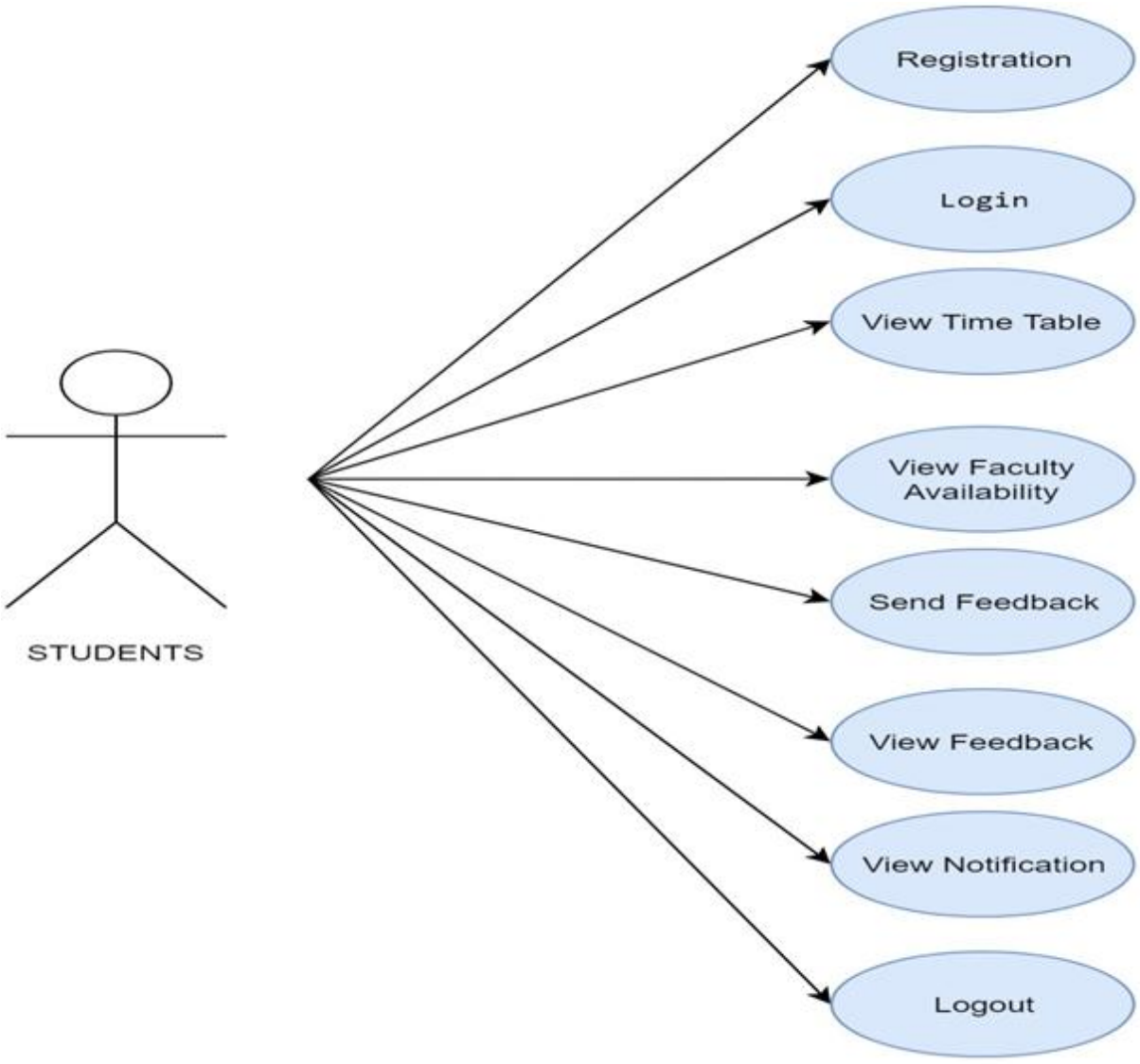

Figure 3.2: Student user interface

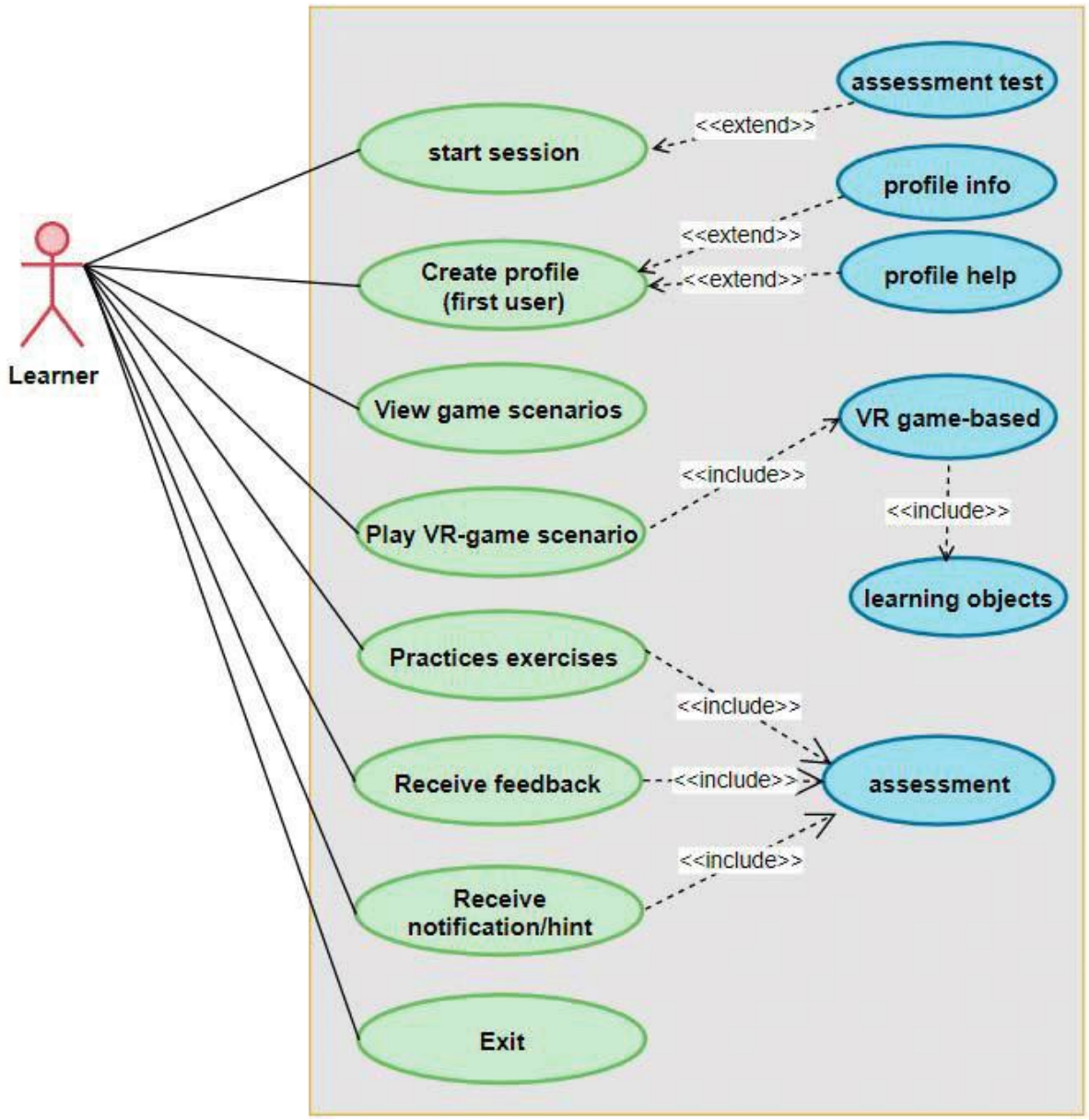

Figure 3.3: Learner user interface

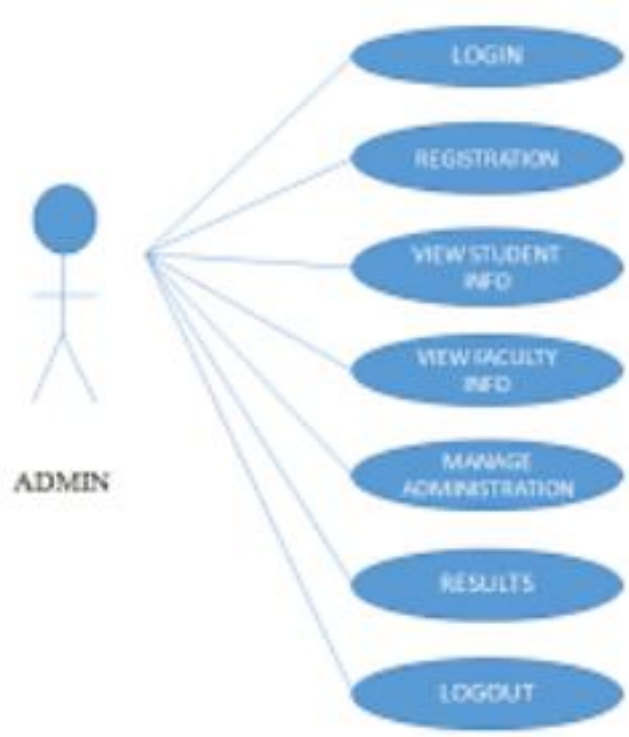

Figure 3.4: Administrators user interface

3.5 Utilizing Activity Diagrams

Activity diagrams serve as a valuable tool for representing the behavior within a business process, devoid of specific object dependencies. These diagrams prove versatile, capable of modeling a spectrum of scenarios, ranging from high-level business workflows encompassing various use cases, down to the intricate specifics of individual use cases or even individual methods. In essence, activity diagrams offer a comprehensive means to model processes of virtually any nature.

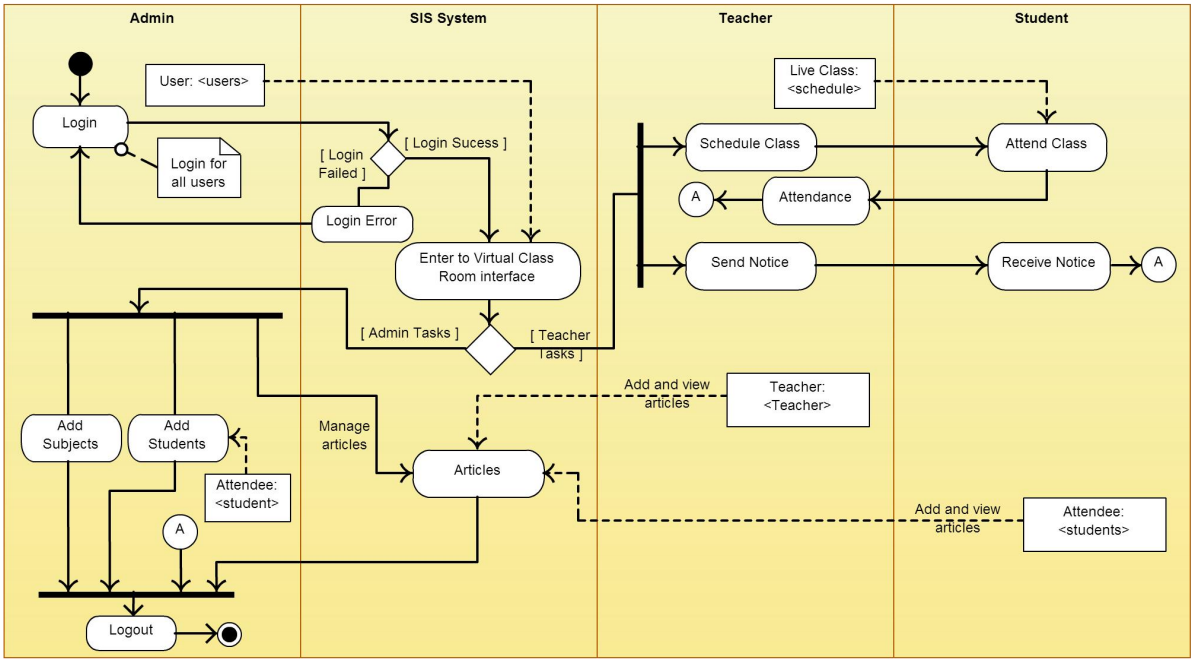

Figure 3.5: Utilization of diagrams

3.6 Class Diagram Overview

An object class in a system is represented visually in a class diagram, which also shows the relationships between the various classes. The different kinds of items that are present in a system are clearly defined by this diagram, along with the static relationships that bind them together. A rectangular shape with one or more horizontal segments is used in the Unified Modeling Language (UML) to represent a class. The class name is the only required component in a class diagram, and it appears in the top section. By convention, class names always start with a capital letter. An optional list of class members or attributes can be placed in the middle segment, and an optional inventory of the class's associated operations or methods can be found in the lower segment.

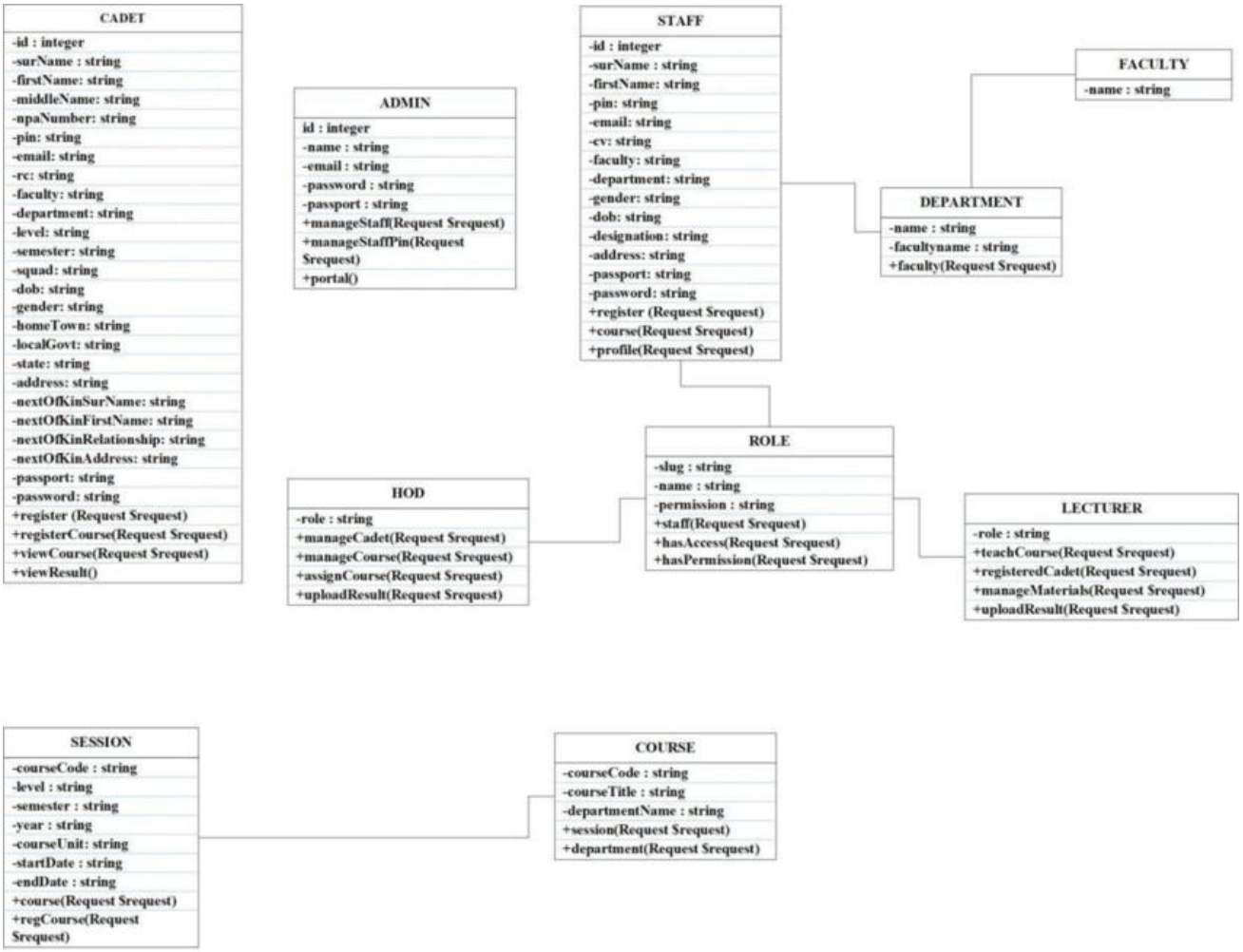

Figure 3.6: Class diagram overview

3.7 System Design Phase

During the system design phase, the main goal is to enhance the analysis model and incorporate essential technologies and implementation constraints. The essence of design lies in making critical decisions regarding how to construct the system effectively. This stage includes a number of tasks that are essential to deciding how the user interacts with the system, including designing the user interface, specifying system inputs, and figuring out system outputs.

3.8 Choice of Development Technologies

The choice of what technologies would be used in the system's design was made in this phase. There are numerous web technologies and programming languages that can make legitimate claims to be suitable for actualizing this system, still considering our design and end goal the following technologies were employed:

3.8.1 Programming Language

JavaScript was employed extensively throughout the design and implementation of the system, it was considered because of its versatility as it could be used for developing both the frontend end and backend of the system, making our code base unified and easy to extend.

3.8.2 Frontend

The frontend which is the part the users of the system would interact with was designed with React.js, the decision to use React.js is because of its robust architecture, React.js is very flexible in terms of design and it is also very easy to test because of its component-based configuration. Development using React.js is very fast and page rendering in React.js is optimized Furthermore, React.js also integrates perfectly with the PWA technologies which is a core feature of our design.

Figure3.7: Front end for PWA-SIMS.

3.8.3 Backend

For the backend of our system, there were several options as to what architecture would be suitable for our application logic, Node.js was considered, and this was largely due to two key factors: 49 1. Faster synchronization because the same language is utilized on the client and server, the frontend was written with JavaScript, and similar tools used in the frontend can also be used on the backend end, saving us valuable time implementing the same system using a different language and learning its different tools, 2. JavaScript supports asynchronous programming, meaning it can handle multiple threads of execution and as such, it can respond to multiple requests coming into the server better than other programming languages, because of its asynchronous nature it doesn't need to wait for one process to complete before attending to the next process, which is what want for our system since it would be used by a large number of users. 3. The need for a real-time application and Node.js specification offers the low latency required for real-time applications to work effectively.

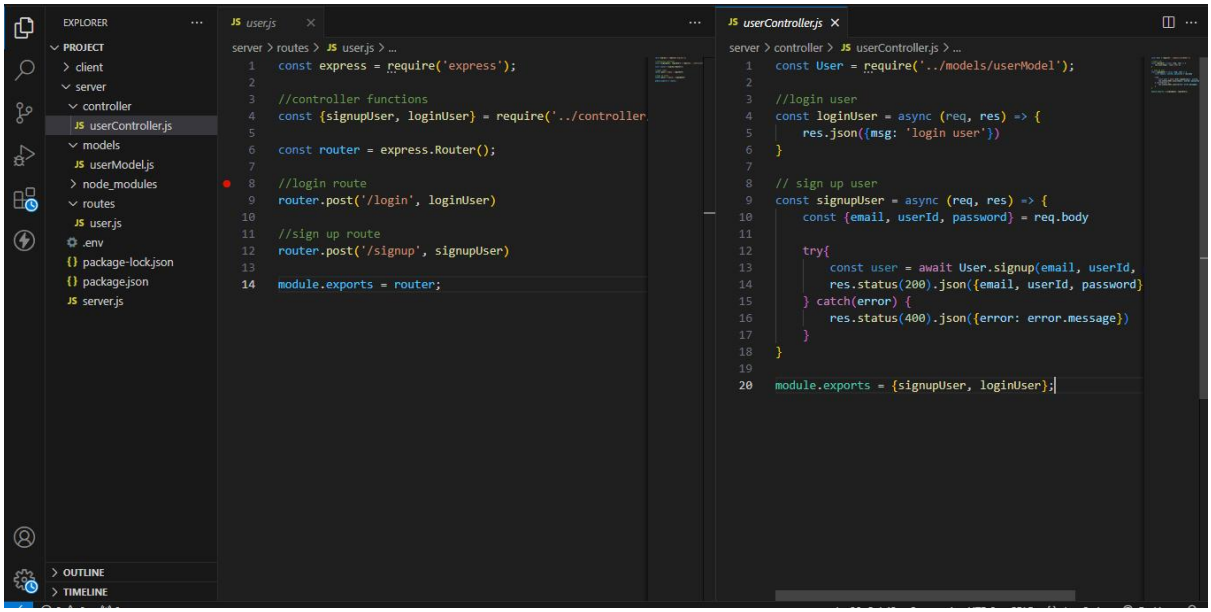

Figure3.8: Backend for PWA-SIMS.

3.8.4 Database

For our database which is where our information would be stored and retrieved from. From our system design retrieving information from the database was more important to us than writing to the database, because a large number of users would be requesting information from the database, and sometimes the same information make be requested by a sizable number of users, also the need to scale our database as the number of users increase to ensure the performance of the database does not drop, we opted for MongoDB a No-SQL database to store our information, MongoDB is more versatile in performing reads from the database and it is also able to scale without much structural change to the database as opposed to regular relational databases.

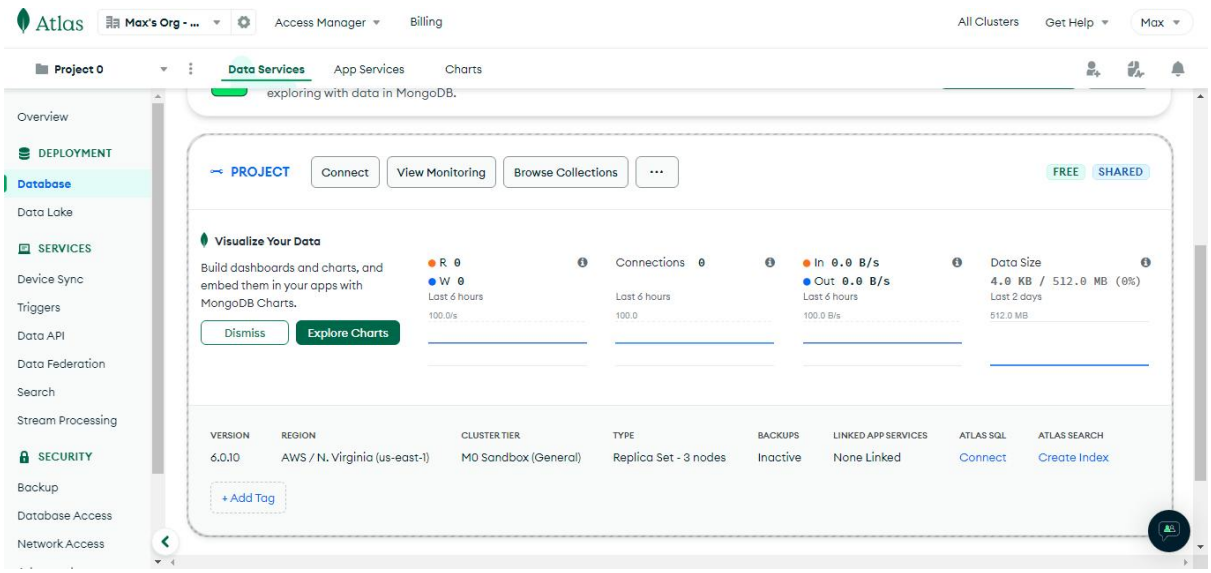

Figure 3.9:MongoDb database for PWA-SIMS.

3.8.5 Security

Security features where added to ensure the protection of vital user data, preventing unauthorized access, ensuring data integrity, and complying with regulations. The enhanced security features added to the include Salting, Hashing and 2FA. Postman software was used to simulate the outcome of the security features.

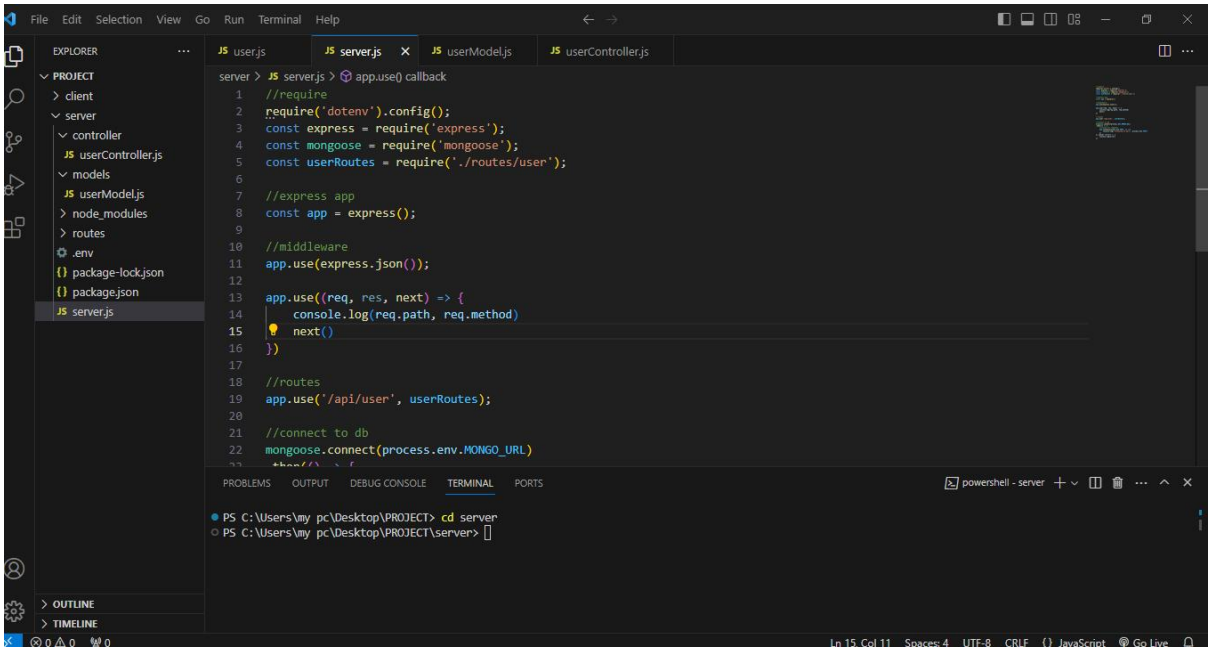

Figure 3.10: Adding Salting and Hashing to the Backend.

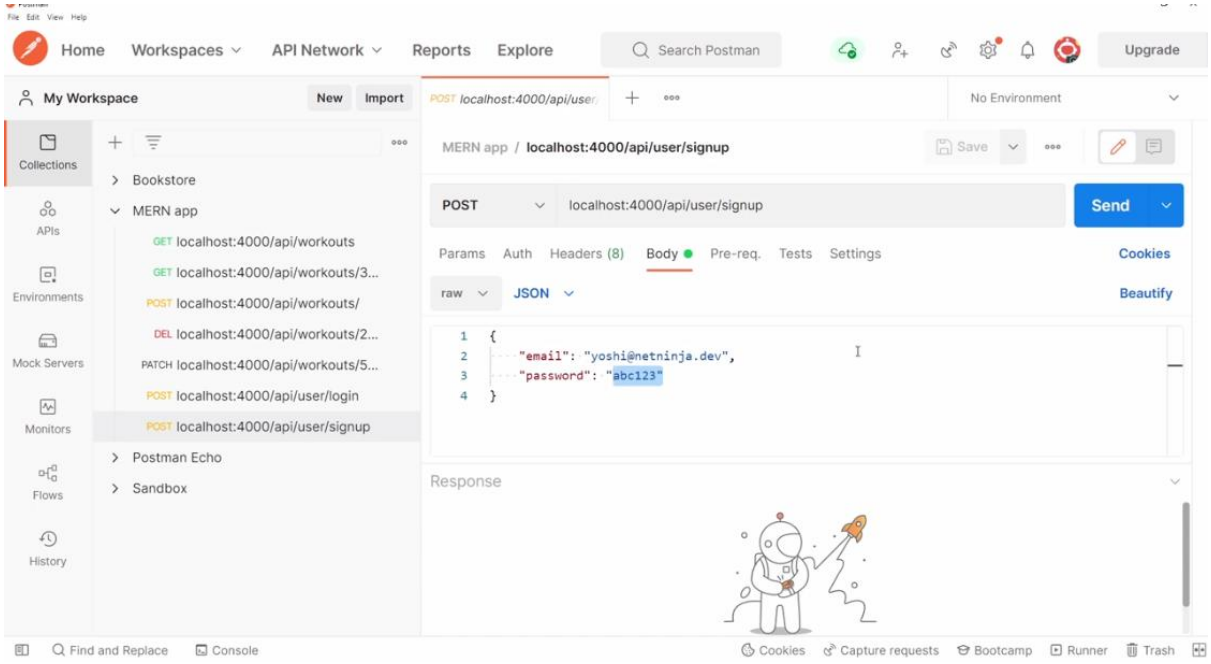

Figure 3.11: Before Hashing user data in POSTMAN.

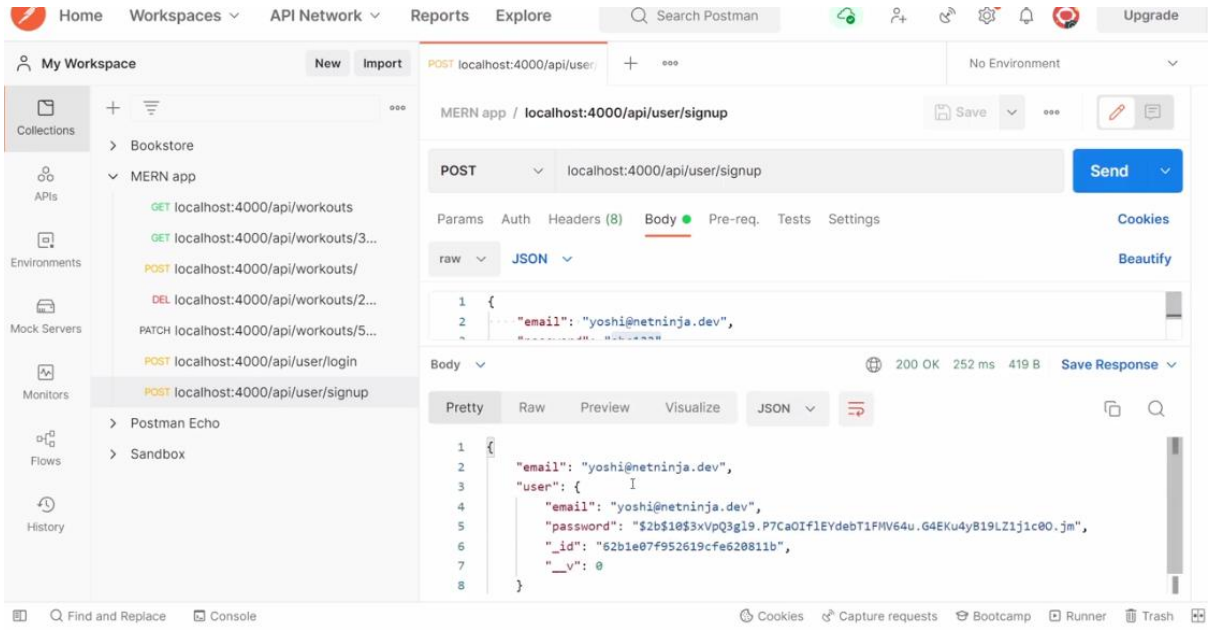

Figure 3.12: After Hashing in POSTMAN.

3.9Software Tools

1. 64-bit Windows 11 Operating System
2. Node.js
3. Google Chrome Browser
4. Postman

3.10Hardware Tools

Personal Computer with the following Specification:

1. Local Disk(C:)- 297GB
2. Installed RAM- 4.00GB
3. System type: 64-bit Operating System, x64-based processor
4. Processor: Intel(R) Core(TM) i3 M 380@2.53GHz.

CHAPTER FOUR

RESULT AND TESTING

4.1 Testing

This provides a comprehensive overview of the test for the design of the PWA-based SIMS using React.js as well as security features that were added to protect students' data against cyber-attacks.

The PWA-based SIMS for the department of Computer Engineering was designed to manage her students' records, track their progress and provide insights to lecturers and administrators.

It was build using front end technologies like JavaScript, CSS and React.js(which is a front end JavaScript library). With the help of service worker technology, the PWA can be used online and can also be installed offline for use on mobile devices.

Security features like Salting, Hash function and 2FA were added to protect the PWA-based SIMS to protect against cyber-attacks into the database system where sensitive student data are kept.

The test covers the following core areas of the project:

1. Functionality
2. Security test

Functionality Tests

Here the functionalities of the various aspects of the PWA are tested to ensure they meet the required functionality requirement of the project. This ensures the project is free from bugs and is efficient and practical from the user's perspective. Some of the core functionalities tested are discussed below:

Case 1: Register a user

Here a user-friendly interface is designed for registering a user. It takes in a new user's data like email, user ID and password, stores the data in the database and helps create an account for the user. The data can be used subsequently to login upon input of correct data and security checks.

Steps in registering a user include:

1. Lunch the web app

2. On the home page, locate the navigation bar

3. To the far right, click on the sign up button

4. Sign up button takes you to the sign up page where a new user can fill up the provided input fields

5. Click on the sign up button to create an account.

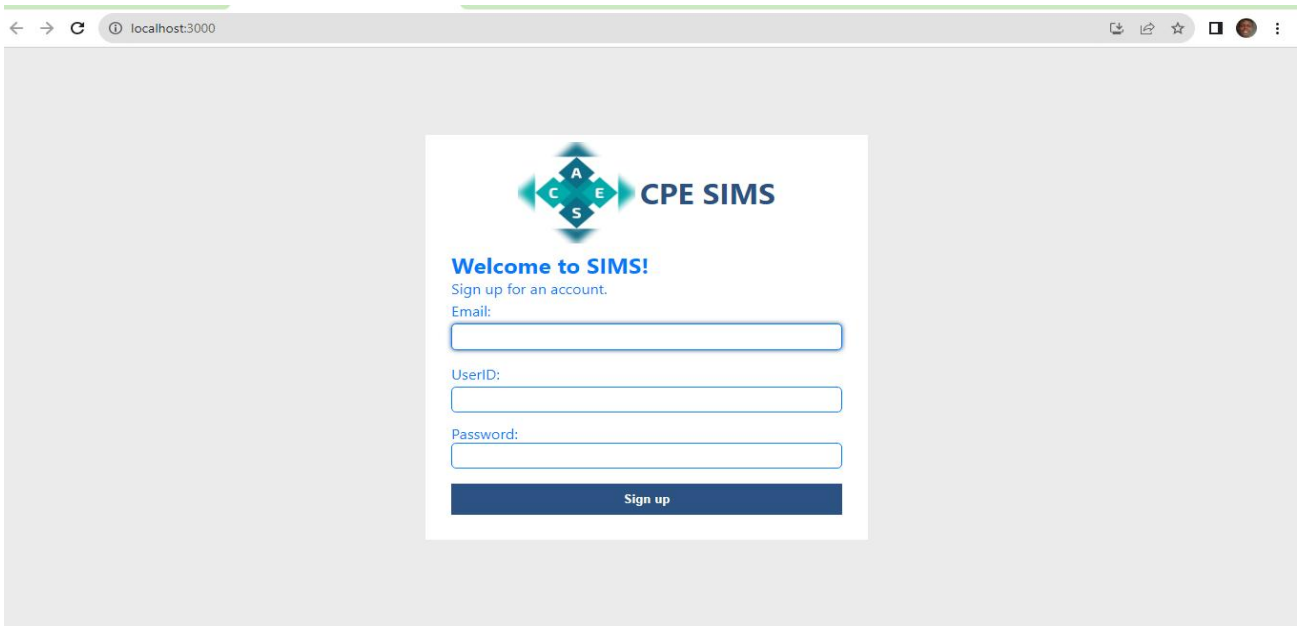

Figure 4.1: Sign up page.

Case 2: Login a user

Here a user-friendly Login interface is designed that takes in user data, checks them against corresponding data in the database, validates the user and grant the user access to his/her dashboard.

Steps to Login a user include:

1. Lunch the web app.

2. On the home page, locate the navigation bar.

3. To the far right, click on the sign in button

4. Sign in button takes the user to the login page where the user can input his/her data in the input fields provided

5. Click the sign in button to login.

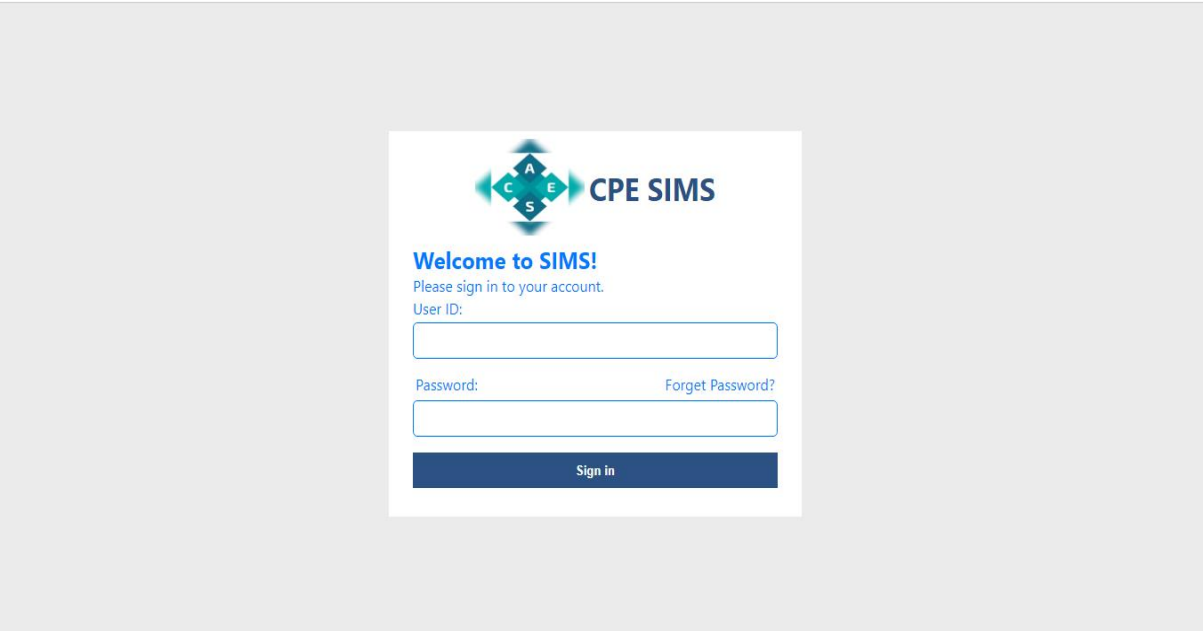

Figure 4.2: Login page.

Case 3: Logout a user

The Logout button enables the user of the PWA to terminate their session in the PWA. When the user logs out, he or she is no longer authenticated. The Logout functionality prevents unauthorized user gain access to user data when the data leaves his or her device unattended to or otherwise.

Steps to Logout from user dashboard:

1. Login into your account
2. Locate the top right corner of the user dashboard
3. Click on the sign out button
4. On successful sign out, the user is taken to the homepage.

Case 4: PWA can be installed

Here upon setting up the service worker file and source code, you run a build command in your terminal in Visual Studio Code or command project. Ensure to run the command in the appropriate directory i.e. the client.

Steps to installing the PWA

1. In Visual Studio Code or command prompt terminal, run a build command
2. In your terminal and in same client directory, run the npm start command to Lunch the PWA to the browser
3. Locate the three dot icon at the top right corner of your browser
4. Click on the "Add to home screen"
5. In the prompt, rename your PWA to any name of your choice
6. Click okay.

7. On the home screen of your device, you'll see the PWA icon indicating successful installation.

Case 5: PWA works offline

The PWA can be installed and used offline without the need of switching on your mobile data. It will work properly with near native app experience allowing for adding or deleting of data etc.

Steps to using the PWA offline:

1. Install the PWA
2. Locate the PWA icon on your mobile or desktop device.
3. Click on the icon to Lunch the PWA
4. Interact with the PWA with your data on
5. Turn off your data and interact with the web app like you would in step 4.
6. Exist the app
7. Launch the web app to inspect changes.

Security Test

The security test for the web app involves assessing its security posture to identify vulnerabilities and weaknesses that could be exploited by attackers. The goal of a security test is to ensure that your PWA is resilient against potential threats and to protect users' data.

Some key aspects of a security test include:

Case 1: SQL injection

One kind of cyberattack known as SQL injection happens when an attacker inserts malicious SQL code into an online application's input fields, including search bars or login forms. SQL injection aims to modify the SQL query of the program in a way that permits illicit data retrieval, data manipulation, or even total control over the database, as well as unauthorized database access.

An example of typical SQL code is:

*SELECT * FROM users WHERE username = 'provided_username' AND password = 'provided_password';*

However, SQL injection will not work in a MongoDB database because MongoDB is a NoSQL database, and SQL injection is a specific type of attack that targets SQL databases. In essence, using the MongoDb database protects against SQL injection by default.

Case 2: Cross-site scripting

Cross-Site Scripting (XSS) is a web security flaw where attackers inject malicious JavaScript code into a site, exploiting users' browsers to steal data or take unauthorized actions.

To prevent against XSS:

1. Install "helmet" and "helmet-csp" using terminal.
2. Require the middleware
3. Configure CSP policies
4. Set the CSP headers:
5. Continue adding your other middleware and routes.
6. Start the Express.js server:

Test XSS prevention using Postman software:

1. Open Postman software.
2. Create a new request.
3. Set the request URL to a route in your web application.
4. Send the request.
5. Inspect the response headers in Postman for CSP-related headers.
6. Test different scenarios with multiple requests.
7. Review the results to ensure CSP policies are correctly applied.
8. Adjust CSP policies in your Express.js application based on test results.

Case 3: Two Factor Authentication (2FA)

2FA ensures that access to the system requires the user to provide two different authentication factors, enhancing the overall security of the application.

Steps to install 2FA to the backend:

1. Install the Twilio Node.js Package.
2. Initialize Twilio Client.
3. Generate and Send 2FA Codes.
4. Receive and Validate 2FA Codes.
5. Implement Code Expiry and One-Time Use.
6. Handle Code Resending and Recovery.
7. Logging and Monitoring.
8. Test the Twilio Integration.
9. Deployment.

10. User Support.
11. Compliance and Regulations.
12. Security Audits.

Steps to test 2FA in postman software:

1. Create a new Postman request.
2. Set the request URL to your 2FA validation endpoint.
3. Choose the appropriate HTTP method (e.g., POST).
4. Add necessary headers (e.g., "Authorization" with API key).
5. Provide parameters for 2FA validation in the request body.
6. Send the request to your Express.js backend.
7. Inspect the response for validation success.
8. Test various scenarios (e.g., success, invalid code).
9. Check error handling (e.g., incorrect parameters).
10. Monitor backend logs for accurate recording.

Case 4: Hashing a password

Hashing uses a hashing function to convert a plain password into a more abstracted and secure form before storing it into the database.

To add hashing function to the Node.js and Express.js backend of the PWA:

1. Install "bcrypt" library using terminal.

2. Require the "bcrypt" library in your backend file.

To check hashing functionality:

1. Define the appropriate schema for the database.
2. Lunch postman software.
3. Enter the local host parameters.
4. Enter a raw JSON data of the appropriate input parameters defined in the schema.
5. Click on send.
6. The response provided by the postman software will show the password field with the Hash password.

CHAPTER FIVE

CONCLUSION

This study offers insights into improving security features in the PWA-based student information management system created with React.js for the front end and Node.js/Express.js for the back end. The study results showcase the mitigation of security vulnerabilities within the PWA, highlighting the implementation of enhanced security measures like salting, hash functions, and Two-Factor Authentication (2FA) to prevent potential attacks.

The implementation of security measures such as salting, hashing, and Two-Factor Authentication (2FA) significantly enhances the security of the PWA-based Student Information System. These security measures ensure that sensitive student data is protected effectively. Users, including course advisors and students, can rely on the system's robust security features to safeguard their data, providing a user-friendly, efficient, and secure solution for managing student information. This added layer of security addresses potential vulnerabilities and limitations of the existing systems, reinforcing the system's overall effectiveness and reliability.

As a result, this study not only contributes to improving the security of student information management but also serves as a practical model for educational institutions seeking to modernize their systems, enhance data protection, and provide a high-quality experience for their students and staff.

5.1 Recommendation

The recommendations listed below are necessary to improve the new system in an effective manner:

i. Regular application upgrades are necessary to guarantee security against the most recent web application vulnerabilities, as new threats appear on a daily basis.

ii. The use of stronger security measures should be employed

REFERENCES

Ali, m. a. (2018, October 15). Laravel Framework. Retrieved from https://www.linkedin.com/pulse/benefits-using-laravel-php-frameworkmuhammadarslan-ali

Bharaagoudar, S. R., Geeta, R. B., &Totad, S. G. (2013). Web-based Student Information Management System. International Journal of Advanced Research in Computer and Communication Engineering, 2(6).

Boi, F. O., Ehigimetor, A. C. (2022). The Design and Implementation of a PWA-Based Stdents Information Management System (Case Study of Computer Engineering)

Cambridge, D. (2019). Meaning of paper-based. Retrieved February 26, 2019, from https://dictionary.cambridge.org/dicitionary/english/paper-based

Dirk-Jan, R. The Current State of Progressive Web Apps (A Study on the performance, compatibility, consistency, security and privacy and user and business impact of progressive web apps)

Falabita, O. S. (2018). Secured Web-based Student Information Management System

ITU.(2019). ICT Security standard roadmap.INTERNATIONAL TELECOMMUNICATION UNION. Retrieved January 24, 2019, from https://www.itu.int/en/ITUT/studygroups/com17/ict/Pages/default.aspx

Kvist, J., &Mathiasson, P. (2019). Progressive Web Apps and other mobile developing techniques: a comparison. Malmo University.

Marcel, G., &Faustin, U. (2019). DEVELOPMENT OF AN ONLINE INTEGRATED STUDENT MANAGEMENT INFORMATION SYSTEM: CASE STUDY "UNIVERSITY OF GITWE". International Journal of Advanced Research in Computer Science, 10(5), 59-67. http://dx.doi.org/10.26483/ijarcs.v10i5.6479

O'Brien, J. A., &Marakas, G. M. (2011).Management Information System. United States: McGrawHill Education-Europe.

Olipas, C. N. P. (2020). THE DESIGN AND DEVELOPMENT OF STUDENT INFORMATION AND VIOLATION MANAGEMENT SYSTEM FOR A HIGHER EDUCATIONAL INSTITUTION. International Journal for Innovating Research in Multidisciplinary field, 6(8), 72-80.

Opoku, M. O. (2015). Information Management and Organisational Performance: A Review of Literature. Mediterranean Journal of Social Sciences, 6(6), 62-70. 10.5901/mjss. 2015.v6n6s1p62 Raja, R., &Nagasubramani, P. C. (2018). Impact of modern technology in education. Journal of Applied and Advanced Research, v3(1), 33-35.

Samer, A., &Rawan, M. (2018).Evaluating the Role of Management Information System Characteristics in Managerial Decision-Making: A Study of Mutah University. International Journal of Academic Research in Business and Social Sciences, 8(5), 185 - 196. WHO/OMS: Extranet Systems. (n.d.). 17. Information management. WHO/OMS: Extranet Systems. Retrieved December 3, 2022, from https://extranet.who.int/lqsi/sites/default/files/attachedfiles/LQMS%2017-1%20to%2017.3%20Information%20management.pdf

Smithine, N., Stock, A. v., Gigler, T., &Glas, B. (2017). OWASP Top 10 2017.OWASP.

Valve, S. (2018, August 7). Laragon. Retrieved from https://www.laragon.org

Yadav, B. P. (2014). Web application vulnerabilities. Helsinki Metropolia University of Applied Sciences , Software Engineering, Helsinki.