A

Project Report

on

ASSET MANAGEMENT

BTech-IT, Sem VI

Prepared By:
Ayush Patel (IT-089)
Darshan Patel (IT-092)

Guided By:
Prof. Deepak C. Vegda
Dept. of Information Technology

Department of Information Technology
Faculty of technology,
Dharmsinh Desai University
College road, Nadiad- 387001

April, 2024

CANDIDATE'S DECLARATION

We declare that 6th semester report entitled **"ASSET MANAGEMENT"** is our own work conducted under the supervision of the guide **PROF. DEEPAK C. VEGDA**.

We further declare that to the best of our knowledge the report for B.Tech. VI semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate's Signature
Candidate's Name: Ayush Patel
Candidate's ID: 21ITUON135

Candidate's Signature
Candidate's Name: Darshan Patel
Candidate's ID: 21ITUON147

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project carried out in the subject of Project-I, entitled "ASSET MANAGEMENT" and recorded in this report is a bonafide report of a work of

1) Ayush Patel	**Roll No.: IT089**	**ID No: 21ITUON135**
2) Darshan Patel	**Roll No.: IT092**	**ID No: 21ITUON147**

of Department of Information Technology, Semester VI. They were involved in Project work during academic year 2023 -2024.

Prof. Deepak C. Vegda
(Project Guide),
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. (Dr.) V. K. Dabhi
Head, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Information Technology at Dharmsinh Desai University for their invaluable support and guidance throughout the duration of this project. We are deeply thankful to Deepak C. Vegda for their unwavering encouragement and expert advice, which have been instrumental in shaping the direction of this project. We are also indebted to the faculty members of the department for their expertise, insightful feedback, and continuous support, which have greatly contributed to the success of this endeavor. We would also like to acknowledge the contributions of our peers and colleagues who have provided valuable insights and encouragement during the course of this project.

This project would not have been possible without the collective efforts and support from the Information Technology. Thank you for believing in us and fostering an environment conducive to learning and innovation.

With Sincere Regards,

1. **Patel Ayush**

2. **Patel Darshan**

TABLE OF CONTENTS

Title **Page No**

ABSTRACT

Every organization has its own Asset Management System in order to perform resource activities. Managing assets or utilizing assets or keeping track of assets is a significant task of the IT team. The main work of the IT team is to maintain all assets of the company like laptops and any other devices and also maintain the condition of that Asset. In order to support IT's, there are some electronic-based systems called Asset management systems (AMS). Many Organizations maintain asset details with excel sheets which are a very tedious job. But this application is a cost-effective one that allows them to manage their asset's data in a simple manner. This project belongs to a category of the web application that can be accessed through PC with an internet connection. This Asset Management System allows the IT team to provide various types of permission to other IT members and also manage Asset's data, status, category, location, etc. Asset Management systems are used to maintain assets of the company and require a significant amount of time if we do it without software

Asset Management Systems (AMS) serve as indispensable tools for organizations aiming to streamline asset tracking and maintenance processes. In contrast to manual methods or reliance on Excel sheets, which often prove cumbersome and prone to errors, AMS offers a more efficient and organized approach. By centralizing asset data and providing functionalities for categorization, status tracking, and location management, these systems empower IT teams to effectively monitor and manage company assets. Moreover, AMS enhances collaboration within IT departments by allowing for the delegation of permissions and responsibilities among team members, thereby optimizing workflow and increasing productivity. As organizations recognize the critical role of asset management in driving operational efficiency and cost-effectiveness, the adoption of AMS emerges as a strategic imperative in today's competitive business landscape.

LIST OF FIGURES AND TABLES

Figures:

Tables:

1. INTRODUCTION

1.1 Project Details: Broad specifications of the work entrusted to you

The Asset Management project is an ambitious endeavor aimed at revolutionizing asset management practices across industries. With a focus on developing a sophisticated web-based system, our team is tasked with creating a comprehensive platform that caters to the diverse needs of organizations. Central to this project is the design and implementation of a user-friendly interface, ensuring stakeholders can effortlessly access and interact with asset-related information. Through intuitive navigation and visually appealing design, we aim to streamline asset tracking, maintenance, and utilization processes, fostering efficiency and productivity.

Integral to the success of the project is the seamless integration with existing databases and third-party applications. By synchronizing asset data with key systems such as ERP and financial software, we ensure data consistency and interoperability, facilitating informed decision-making. Moreover, the implementation of automation features, robust security measures, and scalability provisions underscores our commitment to delivering a solution that not only meets current requirements but also anticipates future needs. Through continuous improvement and innovation, we endeavor to stay at the forefront of asset management technology, driving efficiency, and effectiveness in organizational asset management practices.

1.2 Purpose

The purpose of the Asset Management project is to revolutionize organizational asset management practices by developing a comprehensive, user-friendly web-based system. This system aims to streamline asset tracking, maintenance, and utilization processes, leading to enhanced operational efficiency, reduced downtime, and optimized resource allocation. By providing stakeholders with real-time visibility into asset data and facilitating informed decision-making, the project seeks to improve organizational productivity and profitability. Additionally, the project endeavors to ensure compliance with industry regulations, mitigate security risks, and support organizational growth through scalability and flexibility. Ultimately, the purpose of the Asset Management project is to empower organizations to effectively manage their assets throughout their lifecycle, driving sustainable success and innovation.

1.3 Scope

The scope of the Asset Management project is to develop a robust web-based platform that addresses the diverse asset management needs of organizations across various industries. This encompasses the design and implementation of modules for asset tracking, maintenance management, data integration, reporting, and security measures. The project aims to streamline processes such as asset registration, tracking, and maintenance scheduling while ensuring data integrity and compliance with industry regulations. Additionally, the scope includes providing scalability and flexibility to accommodate organizational growth and technological advancements. By delivering a comprehensive solution with user-friendly interfaces and automation features, the project endeavors to optimize asset utilization, reduce downtime, and drive operational efficiency for organizations.

1.4 Objective

The objectives of the Asset Management project delineate the scope of functionalities that the system aims to achieve, along with its limitations:

What it Can Do:

- Streamline Asset Management Processes: Develop intuitive interfaces and automation features to facilitate asset tracking, maintenance scheduling, and utilization optimization.
- Ensure Data Integrity and Security: Implement robust security measures and data integration techniques to safeguard sensitive asset information and maintain data consistency.
- Enhance Compliance and Reporting: Incorporate features for accurate depreciation calculation and reporting to support regulatory compliance and financial decision-making.
- Promote Scalability and Flexibility: Design the system to accommodate organizational growth and technological advancements without requiring significant redevelopment.

What it Can't Do:

- Perform Physical Asset Tracking: The system relies on manual data entry and updates and does not include capabilities for tracking assets physically.
- Replace Human Decision-Making: While the system provides data-driven insights, it does not replace human judgment in strategic decision-making processes.
- Ensure 100% Security: Despite robust security measures, the system cannot guarantee absolute security against all potential threats, and continuous monitoring and updates are necessary.
- Provide Unlimited Scalability: While designed for scalability, the system may have limitations in handling extremely large volumes of data or rapid organizational expansion without additional infrastructure support.

1.5 Technology and Literature Review

Established Technologies:

- HTML, CSS, JavaScript, and PHP: These foundational web technologies are leveraged for developing the user interface and backend functionality of the asset management system, ensuring cross-platform compatibility and user accessibility.

- Relational Databases: The project utilizes relational database management systems (RDBMS) such as MySQL or PostgreSQL for storing and managing asset data, ensuring data integrity, scalability, and efficient querying.
- Web Services and APIs: Integration with existing databases and third-party applications is facilitated through web services and APIs, enabling seamless data exchange and interoperability

2. PROJECT MANAGEMENT

2.1 Feasibility Study

This section delves deeper into the feasibility of Asset Management project, providing a comprehensive analysis.

2.1.1 Technical Feasibility

Data Availability:

Asset Inventory Data: Gather existing asset inventory data from the college, including details such as asset types, quantities, locations, and ownership information. This data will form the basis for the asset management system.

Asset Information Sources: Explore internal databases, spreadsheets, and records to ensure comprehensive coverage of all college assets. Additionally, consider integrating external data sources such as supplier databases or equipment catalogs for enhanced asset information.

Data Integration: Develop mechanisms to integrate and synchronize asset data from multiple sources, ensuring data consistency and accuracy within the asset management system.

Computational Resources:

Hardware Requirements: Assess the computing resources required to host and run the asset management system. This may include a web server with adequate processing power, storage capacity, and memory to handle database operations and user interactions.

Software Requirements: Identify the necessary software components for system development, including web development frameworks (HTML, CSS, JavaScript), server-side scripting (PHP), and database management systems (MySQL, PostgreSQL).

Technical Expertise:

Web Development: Proficiency in HTML, CSS, and JavaScript is essential for designing and developing the user interface of the asset management system. Additionally, knowledge of PHP is required for server-side scripting to interact with the database.

Database Management: Understanding of database design principles and SQL (Structured Query Language) is necessary for creating and managing the database schema, querying data, and ensuring data integrity within the asset management system.

Security Measures: Familiarity with security best practices, such as encryption, authentication, and authorization mechanisms, is crucial for protecting sensitive asset data from unauthorized access or manipulation.
User Training and Support: Consideration should be given to providing user training and support to ensure that stakeholders are proficient in using the asset management system effectively. This may involve developing user manuals, conducting training sessions, and offering ongoing technical assistance.

2.1.2 Time Schedule Feasibility

Frontend Development:

Requirement Analysis (1 week): Conduct thorough requirements gathering sessions to define the desired functionalities and user interface design elements.

UI/UX Design (2 weeks): Collaborate with UI/UX designers to create wireframes, mockups, and prototypes based on user requirements and feedback.
Frontend Development (4 weeks): Implement the UI/UX designs using HTML, CSS, and JavaScript frameworks like React.js or Angular.js to create responsive and user-friendly interfaces.

Testing and Debugging (1 week): Conduct extensive testing to ensure compatibility, responsiveness, and usability across different devices and browsers. Address any issues or bugs identified during testing.

Backend Development:

Database Design (1 week): Design the database schema based on the asset management system's requirements, including tables, relationships, and constraints.

Server-side Scripting (3 weeks): Develop server-side scripts using PHP to handle data processing, user authentication, and database interactions.

Integration and Testing (1 week): Integrate the frontend and backend components, conduct end-to-end testing to ensure seamless functionality, and address any integration issues or inconsistencies.

2.1.3 Operational Feasibility

User Acceptance: Assessing the willingness and readiness of college staff members to adopt and utilize the asset management system.

Training Needs: Identifying and addressing the training requirements for users to effectively navigate and utilize the asset management system.

Organizational Impact: Evaluating the potential impact of implementing the asset management system on existing workflows, processes, and organizational structures within the college.

Resource Availability: Determining the availability of resources, including personnel, equipment, and budget, to support the implementation and ongoing operation of the asset management system.

2.2 Project Planning

2.2.1 Project Development Approach and Justification

For the Asset Management project, we will adopt an Agile development approach to ensure flexibility, adaptability, and continuous improvement throughout the software development lifecycle. Agile methodologies are particularly suitable for projects like Asset Management, which require frequent feedback loops, iterative development, and the ability to respond quickly to changing requirements and priorities.

Approach:

Our Agile development approach will be structured around the following phases:

Requirements Gathering:
Engage stakeholders to understand their needs and expectations regarding asset management functionalities.
Prioritize requirements based on their importance and feasibility for each iteration.

Design the Requirements:
Translate gathered requirements into user stories, defining the desired features and functionalities of the asset management system.
Collaborate with stakeholders to refine and validate the design decisions.

Construction/Iteration:
Develop the asset management system iteratively, focusing on delivering small, incremental improvements in each iteration.
Conduct regular sprint planning sessions to define the scope of work for each iteration and allocate resources accordingly.
Implement feedback mechanisms to gather input from stakeholders and end-users, ensuring that the developed features meet their expectations.

Testing/Quality Assurance:
Integrate quality assurance activities throughout the development process, including automated testing, code reviews, and user acceptance testing.
Ensure that each increment of the asset management system meets predefined quality standards and functional requirements.

Deployment:
Deploy new releases of the asset management system in a timely manner, following established deployment procedures and best practices.
Monitor system performance and stability post-deployment, addressing any issues or bugs that may arise.

Feedback:
Solicit feedback from stakeholders and end-users at the end of each iteration to identify areas for improvement and prioritize future enhancements.
Use feedback to adapt and refine the development approach, ensuring that the asset management system continues to evolve in line with changing needs and priorities.

Justification:

Advantages of Agile Model for Asset Management Project:

Customer Satisfaction: Agile promotes continuous delivery of useful software, ensuring that stakeholders have early and frequent access to new features and improvements, thereby enhancing customer satisfaction.

Flexibility and Adaptability: The iterative nature of Agile allows for flexibility in responding to changing requirements, priorities, and market conditions, ensuring that the asset management system remains relevant and effective over time.

Collaboration and Communication: Agile emphasizes collaboration and communication between project stakeholders, fostering a shared understanding of project goals and requirements, and facilitating timely decision-making.

Quality Assurance: By integrating testing and quality assurance activities throughout the development process, Agile helps ensure that the asset management system meets predefined quality standards and functional requirements, reducing the likelihood of defects and issues.

Continuous Improvement: Agile encourages regular reflection and adaptation, enabling the project team to identify areas for improvement and implement changes to enhance productivity, efficiency, and effectiveness.

2.2.2 Project Plan

1. Project Initiation
2. Design and Planning
3. Development
4. Testing and Quality Assurance
5. Deployment
6. Launch and Post-Deployment
7. Documentation and Reporting

2.2.3 Milestones and Deliverables

System Design Development:

Create a comprehensive system design addressing the asset management requirements and challenges identified during the analysis phase.
Deliverable: Detailed system design documentation outlining the architecture, components, and interactions of the asset management system.

Design Implementation and Testing:

Implement the system design and conduct thorough testing to ensure functionality and address any identified enhancements.
Deliverable: Tested implementation of the system design with documented enhancements and improvements.

2.2.4 Roles and Responsibilities

Roles are, we have group of 2 person in developing project. So both are involved in making of both backend and frontend of half of the total modules.

Name	Analysis	Design	Coding	Testing	Documentation	Maintenance
Ayush	✓	✓	✓	✓	✓	✓
Darshan	✓	✓	✓	✓	✓	✓

Table: 2.2.4 roles and responsibility

2.2.5 Group Dependencies

- Users have sufficient privileges to access the internet.
- Browser on Device is running smoothly.
- Database updates are giving expected and accurate results.

2.3 Project Planning

Sr. No	Documents	Timeline
1	Thinking for Project Definition	23/12/2023
2	Approval for the project title and framework	04/01/2024
3	Implementation	09/01/2024 to 29/02/2024
4	Testing	05/03/2024

Table: 2.3.1 Project Scheduling Chart

3. SYSTEM REQUIREMENTS STUDY

3.1 Study of Current System

In the current asset management system, colleges predominantly rely on manual methods and legacy systems to track and manage assets. This includes spreadsheets, paper-based records, and decentralized data management practices. Administrative personnel and facility managers oversee asset tracking, maintenance, and allocation using these traditional methods.

.

3.2 Problems and Weaknesses of Current System

Inefficient Tracking Methods: Reliance on manual spreadsheets or outdated software for asset tracking leads to inefficiencies in monitoring and managing college assets, resulting in difficulties in locating and utilizing resources.

Lack of Integration: Fragmented data storage and disparate systems hinder seamless integration between different departments, causing difficulties in coordinating asset management efforts and sharing information effectively.

Limited Visibility and Accountability: The current system may lack real-time visibility into asset status and usage, leading to challenges in tracking asset movements and holding individuals accountable for asset responsibilities.

Manual Maintenance Processes: Manual workflows for asset maintenance and servicing result in delays and inconsistencies, impacting the reliability and performance of college assets.

3.3 User Characteristics

There is only one user:
Asset Manage User:

- Login
- Registration
- Purchase
- Recouring
- Update Asset Information
- Assign to Lab
- Download Data
- Edit Profile

3.4 Hardware and Software Requirements (Minimum requirements to run your system)

There are no such specific hardware requirements other than basic requirements such as a computer with good internet connectivity and a decent browser that supports React & JavaScript.
Software: -

- Operating System: Windows Operating System 2000 and above and Linux
- Visual Studio Code
- HTML, CSS, JAVASCRIPT
- PHP Database

PHP Database: A PHP database refers to a database system that is used in conjunction with the PHP programming language. This could include relational databases like MySQL or PostgreSQL. In PHP applications, these databases are utilized for storing, retrieving, and managing data efficiently.
Visual Studio Code: Visual Studio Code is the Integrated Development Environment (IDE) for Web app development.

3.5 Constraints

3.5.1 Regulatory Policies

Regulatory policies in the Asset Management System outline guidelines for legal compliance and data integrity. They ensure adherence to privacy regulations like GDPR, defining access control and audit trail requirements. Compliance maintains accountability and security within the system.

3.5.2 Hardware Limitations

Hardware requirements dictate system performance and scalability. Processor speed, memory, and network bandwidth affect operation. Upgrades may be necessary to handle increased workload or maintain optimal performance.

3.5.3 Interfaces to Other Applications

Integration with external systems streamlines workflows and data exchange. APIs enable seamless communication with ERP, CRM, and inventory management platforms. Integration enhances operational efficiency and data accuracy across interconnected systems.

3.5.4 Parallel Operations

Parallel processing optimizes task execution and resource utilization. It improves data retrieval, analysis, and maintenance tasks. Effective coordination is essential to manage shared resources and maintain data consistency.

3.5.5 Higher Order Language Requirements

Support for languages like HTML, CSS, JavaScript, and PHP enables dynamic web-based applications. These languages facilitate user interface development and business logic implementation. Proficiency ensures customization and alignment with user requirements.

3.5.6 Reliability Requirements

System uptime and fault tolerance are critical for uninterrupted operation. Redundant components and backup mechanisms ensure data integrity. Proactive monitoring detects issues to maintain reliability and performance.

3.5.7 Criticality of the Application

The system's importance lies in its impact on business operations and decision-making. Downtime or data loss can disrupt asset management workflows. Reliability and security measures mitigate risks and ensure operational continuity.

3.5.8 Safety and Security Considerations

Data security measures protect sensitive asset information from unauthorized access. Role-based access control and encryption safeguard data integrity. Regular audits and incident response protocols address security breaches promptly.

3.6 Assumptions and Dependencies

Assumption: The Asset Management System assumes that users have access to stable internet connectivity throughout the system's operation. Additionally, it assumes that users possess basic computer skills necessary to interact with the system efficiently.

Dependency: The successful implementation of the Asset Management System relies on the integration with external databases for seamless data synchronization. Moreover, the system's functionality may depend on the availability and compatibility of third-party APIs for additional features and service.

4. SYSTEM ANALYSIS

4.1 Requirements of New System (SRS)

4.1.1 Functional Requirements

1. Login:

- Users should be able to securely log in to the asset management system using their credentials.
- Authentication mechanisms should ensure only authorized users gain access to the system's functionalities.

2. Registration:

- New users should have the ability to register for an account by providing necessary details.
- The registration process should include validation checks to ensure data accuracy and integrity.

3. Purchase:

- Authorized users should be able to initiate asset purchase requests within the system.
- Purchase functionalities should include options for specifying asset details, quantities, and other relevant information.

4. Recurring:

- The system should support recurring asset management tasks such as regular maintenance schedules or subscription renewals.
- Users should have the ability to set up recurring tasks and define their frequency and duration.

5. Update Asset Information:

- Authorized users should be able to update existing asset information as needed.
- Updates may include changes to asset details, quantities, statuses, or other relevant attributes.

6. Assign to Lab:

- Users with appropriate permissions should be able to assign assets to specific laboratories or departments.
- Assignment functionalities should allow users to specify asset allocations accurately and efficiently.

7. Download Data:

- Users should have the capability to download asset-related data and reports from the system.
- Download options should include formats such as PDF, or Excel for ease of use and compatibility.

8. Edit Profile:

- Users should be able to edit and update their profile information within the system.
- Profile editing functionalities should include options for modifying personal details, contact information, and preferences.

4.1.2 Non-Functional Requirements

1. Performance

- The website should respond to users inputs within 5 sec.
- The prediction model should handle concurrent requests efficiently.

2. Security

- Implement secure encryption for user data.
- Ensure secure user authentication practices.

3. Usability

- The website should have an intuitive and user-friendly interface.
- Provide accessibility features for users with disabilities.

4. Reliability

- The system should be available 99.9% of the time.
- Implement regular backups to prevent data loss.

5. Scalability

- Design the system to handle a growing user base without significant performance degradation.

6. Compliance

- Comply with relevant data protection and privacy regulations.
- Adhere to ethical considerations in handling health-related data.

4.2 Features of New System

- Asset Inventory Management: Users can maintain a comprehensive inventory of all assets owned or managed by the organization, including details such as asset type, quantity, location, condition, and other relevant attributes.

- Asset Tracking and Identification: The system provides tools for tracking assets throughout their lifecycle, including acquisition, assignment, maintenance, and disposal. Each asset is assigned a unique identifier for easy tracking and identification.

- Maintenance Scheduling: Users can schedule and manage maintenance activities for assets to ensure optimal performance and longevity. The system sends reminders for scheduled maintenance tasks and tracks maintenance history for each asset.

- Procurement and Purchase Order Management: The system facilitates the procurement process by allowing users to create and manage purchase orders for acquiring new assets. Users can track the status of purchase orders and manage vendor relationships efficiently.

- Reporting and Analytics: The system generates reports and analytics to provide insights into asset utilization, performance, maintenance costs, and other key metrics. Users can analyze data trends, identify opportunities for optimization, and make data-driven decisions.

- User Permissions and Access Control: The system enforces user permissions and access control to ensure that only authorized users can view, edit, or delete asset information. Role-based access control allows administrators to define user roles and access levels.

4.3 Navigation Chart

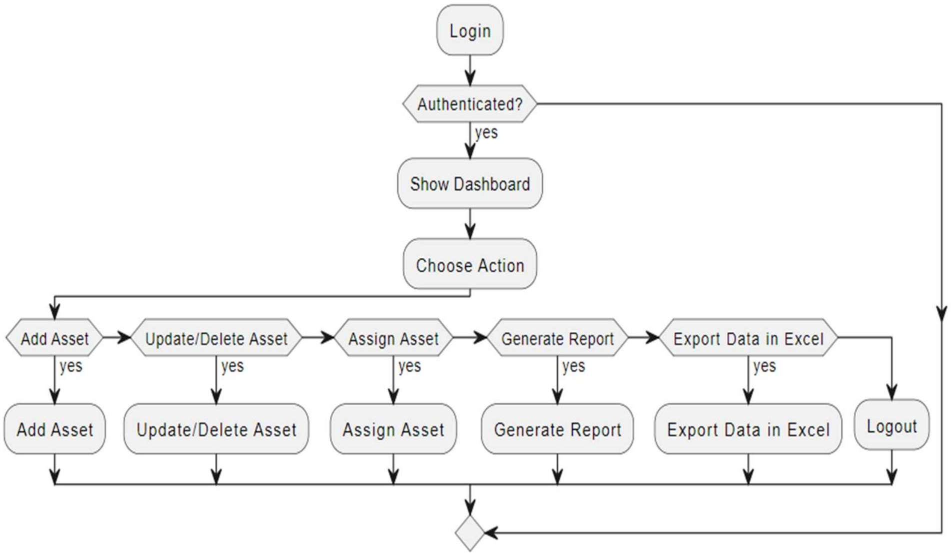


Figure 5.4 Flow Chart

4.4 Data Modeling

4.4.1 Data Dictionary

For User:

UserID: Unique identifier for each user record. (Auto-generated by database)
Username: Username chosen by the user. (String, Required, Unique)
Email: Email address of the user. (String, Required, Unique)
Password: Encrypted password for user authentication. (String, Required)
Role: Role or access level of the user in the system. (String, Required)

For Asset:

AssetID: Unique identifier for each asset record. (Auto-generated by database)
Name: Name or description of the asset. (String, Required)
Type: Category or type of the asset. (String, Required)
Location: Current location or storage area of the asset. (String, Optional)

Quantity: Number of units available for the asset. (Integer, Required)
Status: Current status of the asset. (String, Required)
PurchaseDate: Date when the asset was purchased. (Date, Required)
PurchaseCost: Cost of purchasing the asset. (Decimal, Required)

For Transaction:

TransactionID: Unique identifier for each transaction record. (Auto-generated by database)
UserID: Foreign key linking to the User table, representing the user involved in the transaction. (Integer, Required)
AssetID: Foreign key linking to the Asset table, representing the asset involved in the transaction. (Integer, Required)
TransactionType: Type of transaction (e.g., purchase, assignment). (String, Required)
TransactionDate: Date when the transaction occurred. (Date, Required)
Notes: Additional notes or details about the transaction. (String, Option)

5. SYSTEM DESIGN

5.1 Use Case Diagram

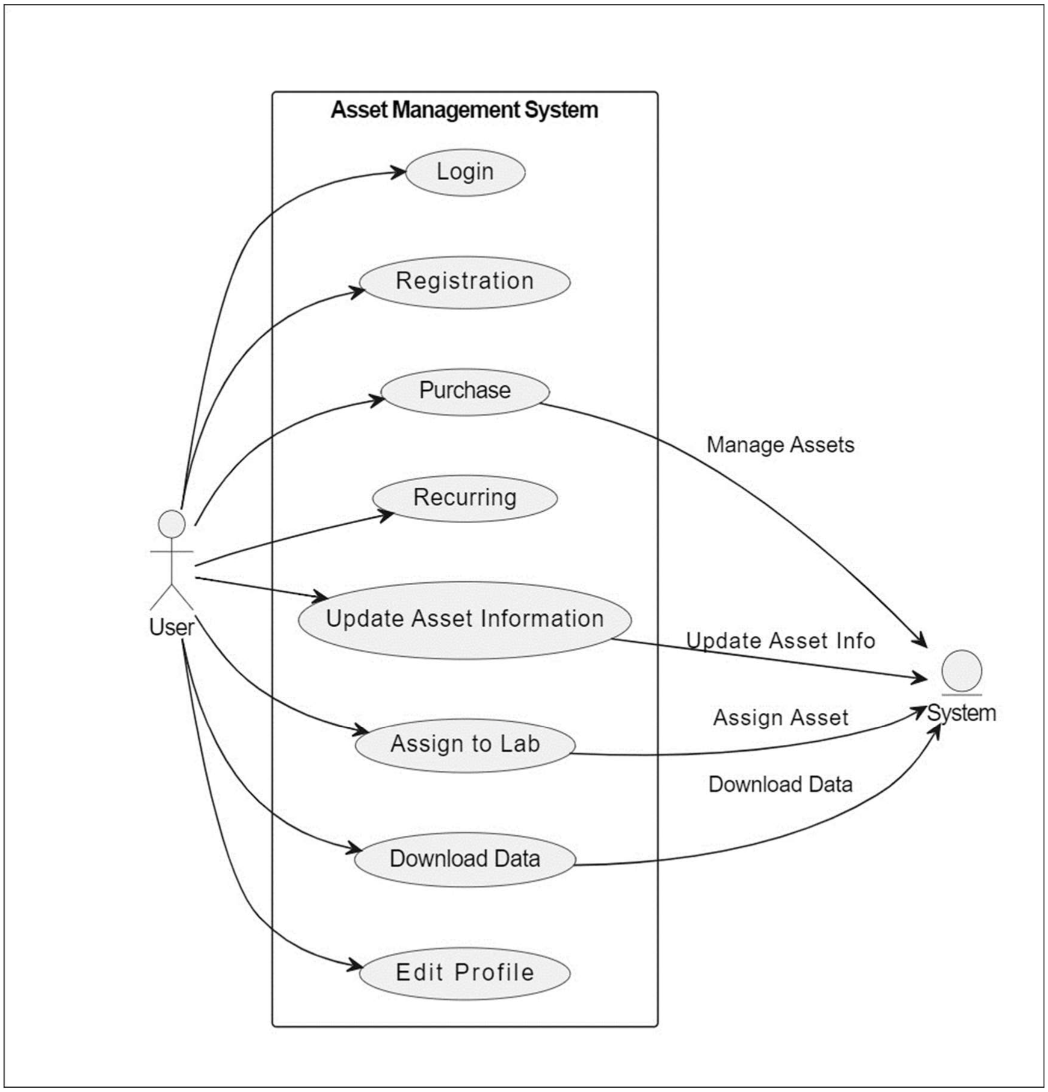


Figure 5.1 Use Case Diagram

5.2 Class Diagram

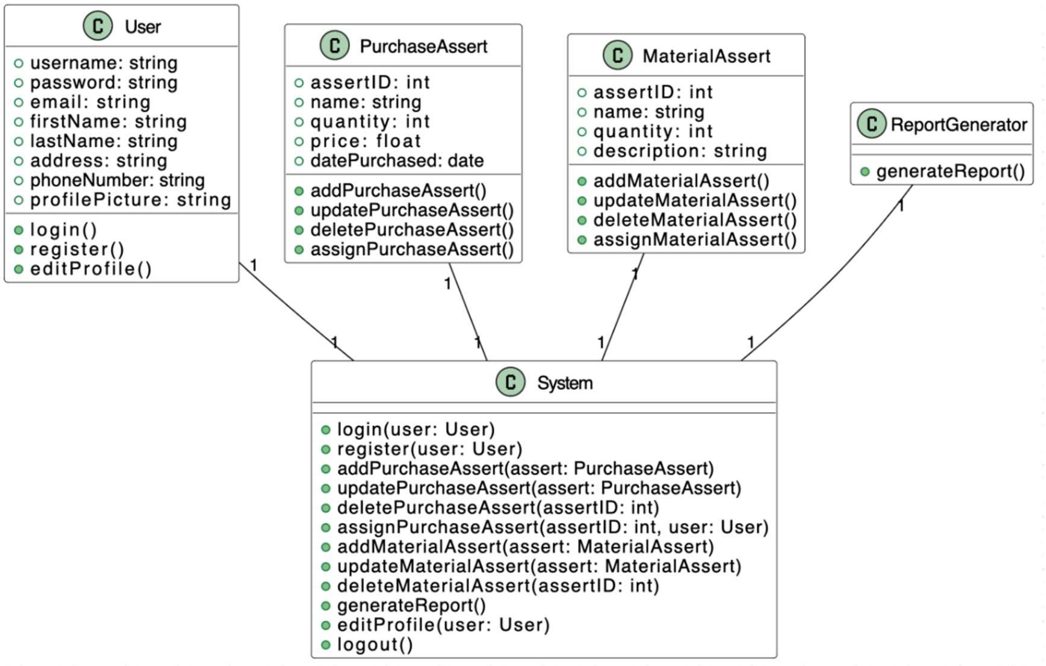


Figure 5.2 Class Diagram

5.3 Activity Diagram

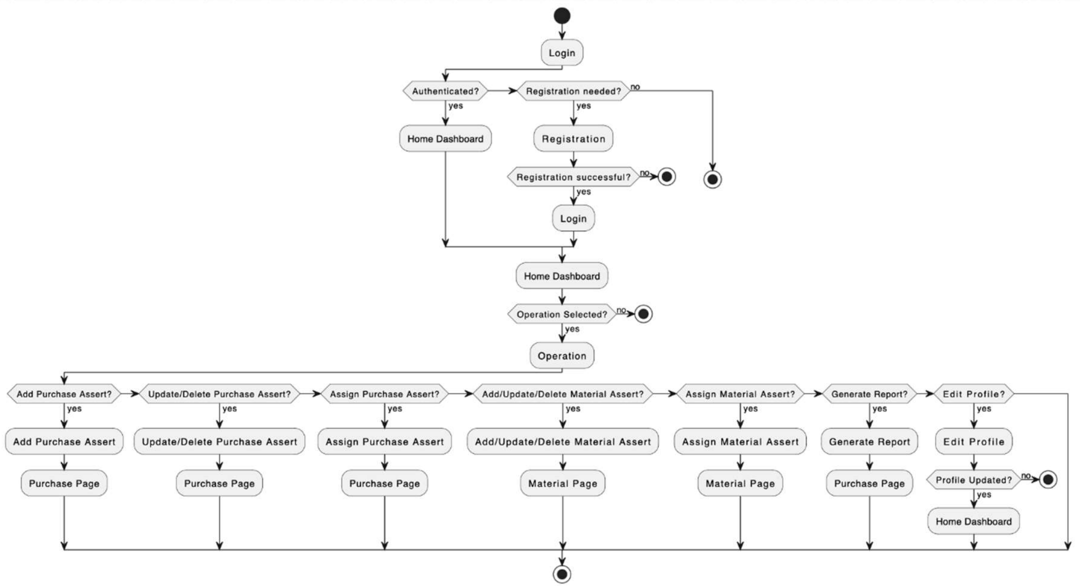


Figure 5.2 Activity Diagram

5.4 Sequence Diagram

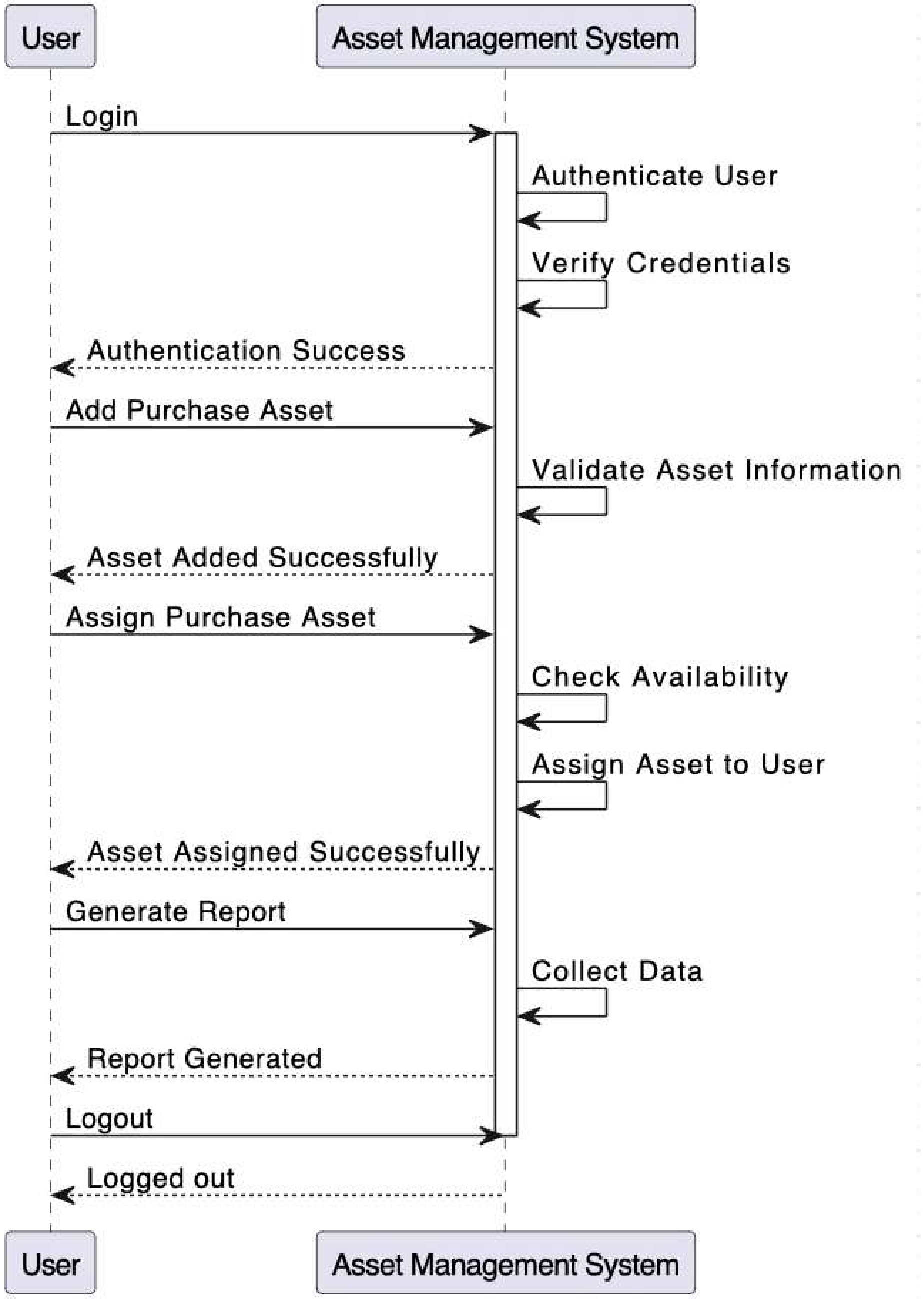


Figure 5.2 Sequence Diagram

6. IMPLEMENTATION PLANNING

6.1 Implementation Environment

For the implementation, we have used: Visual Studio Code Our project is built using Visual Studio Code seeing that it is a web application we saw fit that Visual studio code provided us with all the required basis for the successful implementation of our web app. Also, for storing our data we have used phpMyAdmin database which enables our web app to run at all times.

6.2 Program/Modules Specification

The following Modules are implemented:
USER:

- Login
- Registration
- Add Asset
- Update and Delete Asset
- Assign Asset
- Generate Report
- Export Data in Excel

6.3 Coding Standards

To make the system coding easy, easy to remember, and reduce the chances of errors, some techniques are used at the time of coding of the application which is called coding standard. The coding standard which we adopted during the coding is explained as follows:

- Each nested block should be properly indented and spaced. The code should be properly commented on for understanding easily.
- Comments regarding the statements increase the understandability of the code. Better to avoid the use of digits in variable names.
- The names of the function should be written in camel case starting with small letters. The name of the function must describe the reason for using the function clearly and briefly.

7. TESTING

7.1 Testing Plan

The testing technique that is going to be used in the project is White box testing. In White box testing the Tester has knowledge about the internal structure of the code or the program of the software.

7.2 Testing Strategy

The development process repeats this testing subprocess a number of times for the following phases.

a) Unit Testing.

b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed.

Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

7.3 Testing Methods

Black Box and White Box Testing:

In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited.

Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

Out of the 2 methods for testing, black box testing and white box testing, we would be using the white box testing as we are well aware of the internal functionalities of our application unlike in the black box testing, where we require a 3rd party to test our cases and the internal details are hidden from him.

7.4 Test Cases

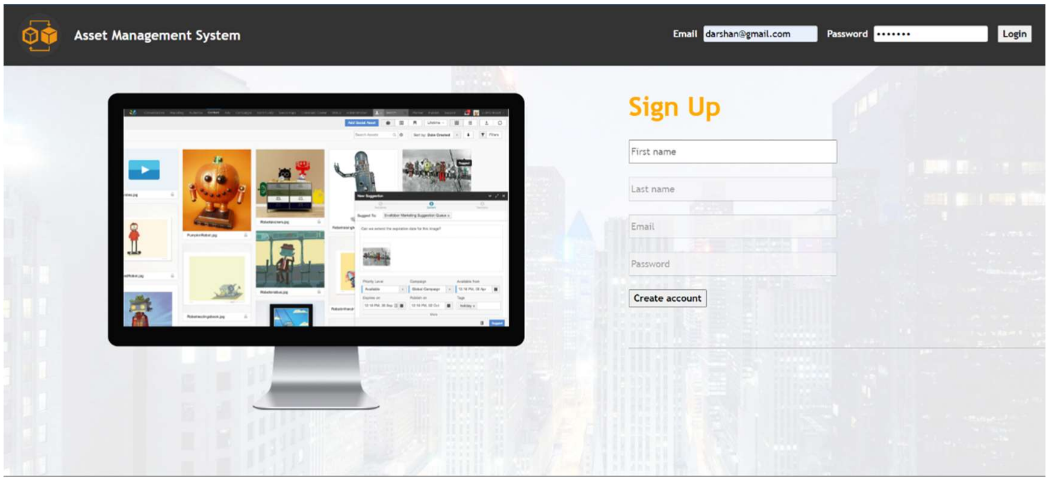
ENITITY	TEST CASE	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
User Log in	Validation	Successfully logged in and redirect to Home page	Successfully logged in and redirect to Home page	Pass
Add Purchase asset	Enter asset details	Purchase asset should be added	Purchase asset added and displayed on screen	Pass
Add Material asset	Enter asset details	Material asset should be added	Material asset added and displayed on screen	Pass
Update asset details	Update asset details	Asset details should be updated	Updated asset details saved and displayed on screen	Pass
Assign asset	Validation	Asset should be assigned to lab	Asset successfully assigned to lab	Pass

Generate Report	Generate Report	Report generated successfully	Report generated successfully	Pass
Export data in Excel	Validation	Data should be exported in excel	Data exported successfully	Pass

Table: 7.4.1 Test Cases

8. USER MANUAL

- **Login Page**



Email darshan@gmail.com

Password •••••••

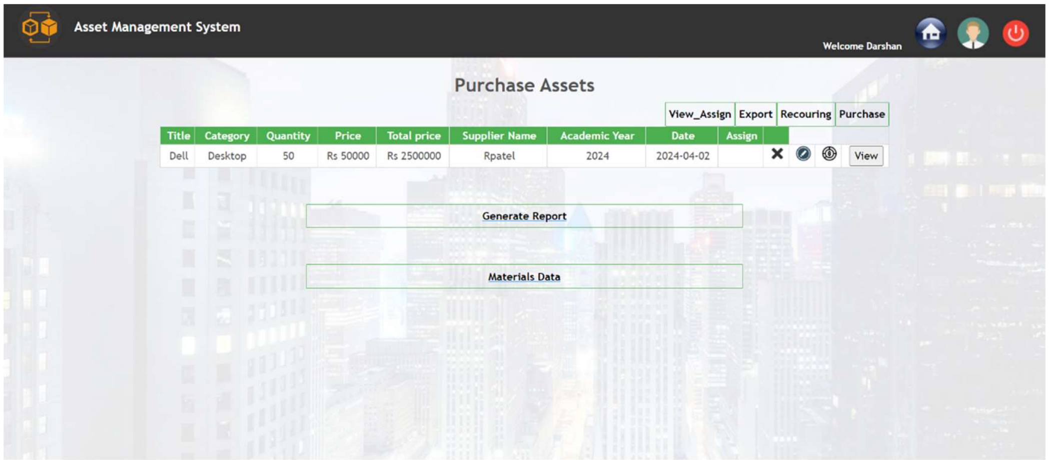
First name

Last name

Email

Password

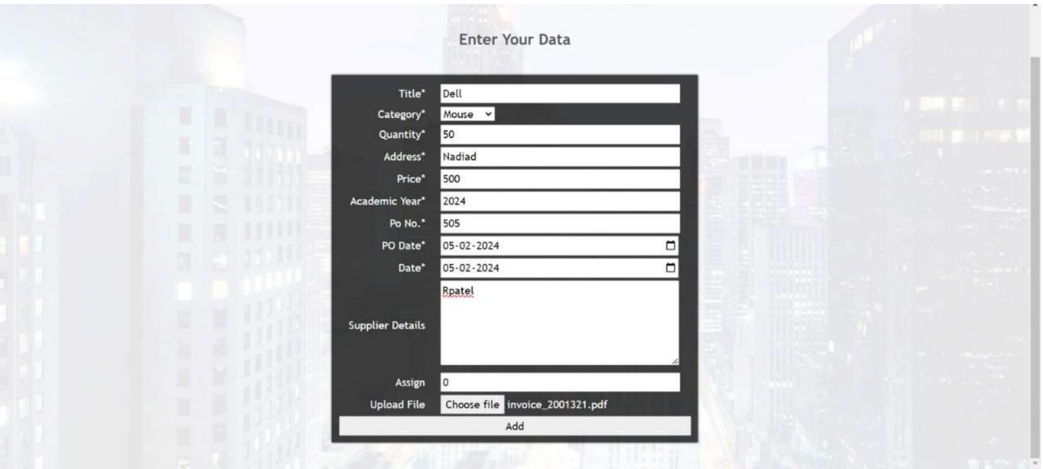
- **Home Page**



Purchase Assets

Title	Category	Quantity	Price	Total price	Supplier Name	Academic Year	Date	Assign	
Dell	Desktop	50	Rs 50000	Rs 2500000	Rpatel	2024	2024-04-02		✕ View

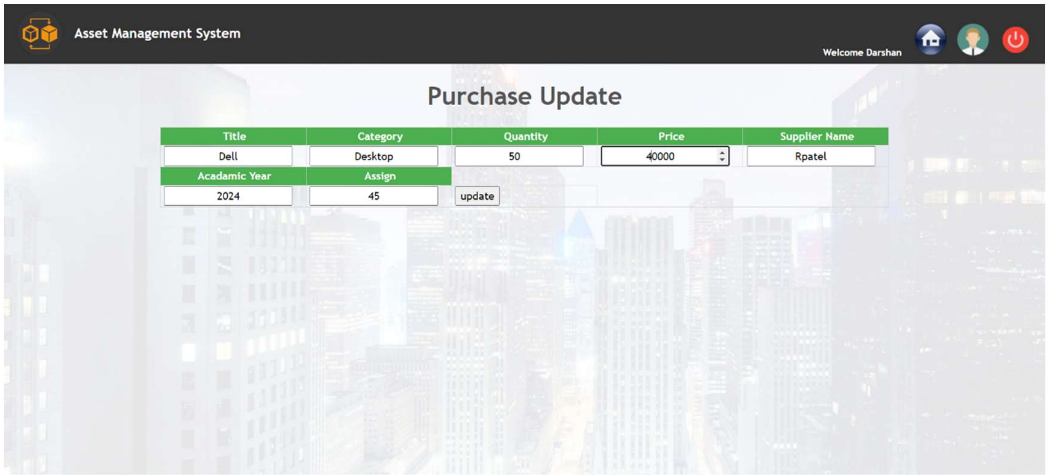
- **Add Purchase Asset**



Enter Your Data

Title*	Dell
Category*	Mouse
Quantity*	50
Address*	Nadiad
Price*	500
Academic Year*	2024
Po No.*	505
PO Date*	05-02-2024
Date*	05-02-2024
Supplier Details	Rpatel
Assign	0
Upload File	Choose file invoice_2001321.pdf

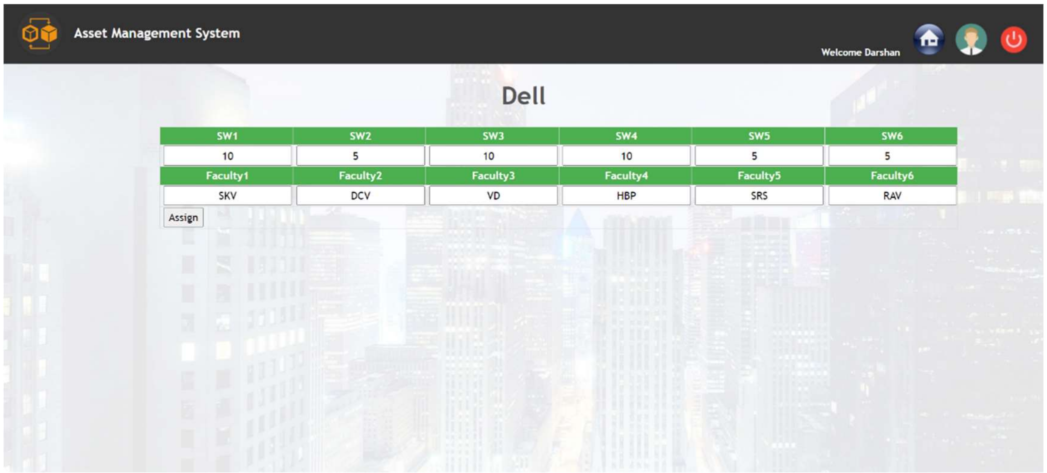
- **Update Purchase Asset**



Purchase Update

Title	Category	Quantity	Price	Supplier Name
Dell	Desktop	50	40000	Rpatel
Acadamic Year	**Assign**			
2024	45	update		

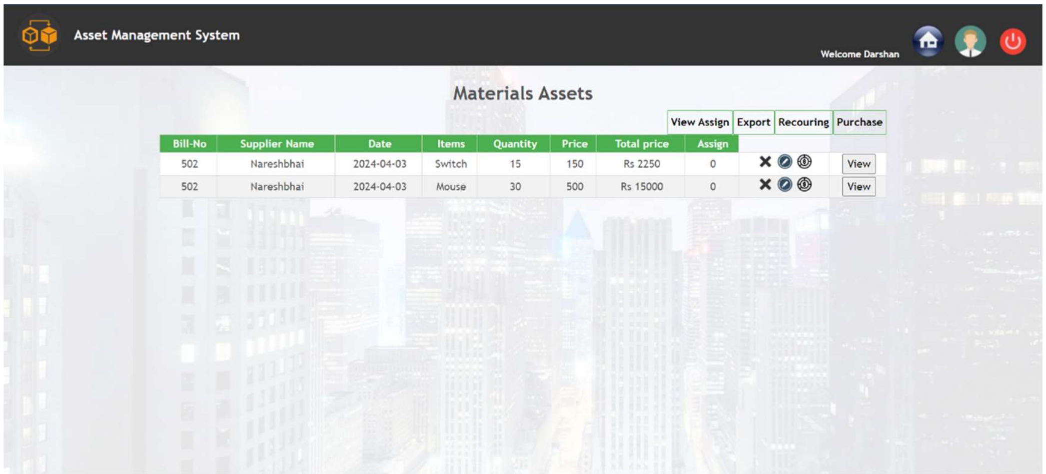
- **Assign Asset**



Dell

SW1	SW2	SW3	SW4	SW5	SW6
10	5	10	10	5	5
Faculty1	**Faculty2**	**Faculty3**	**Faculty4**	**Faculty5**	**Faculty6**
SKV	DCV	VD	HBP	SRS	RAV

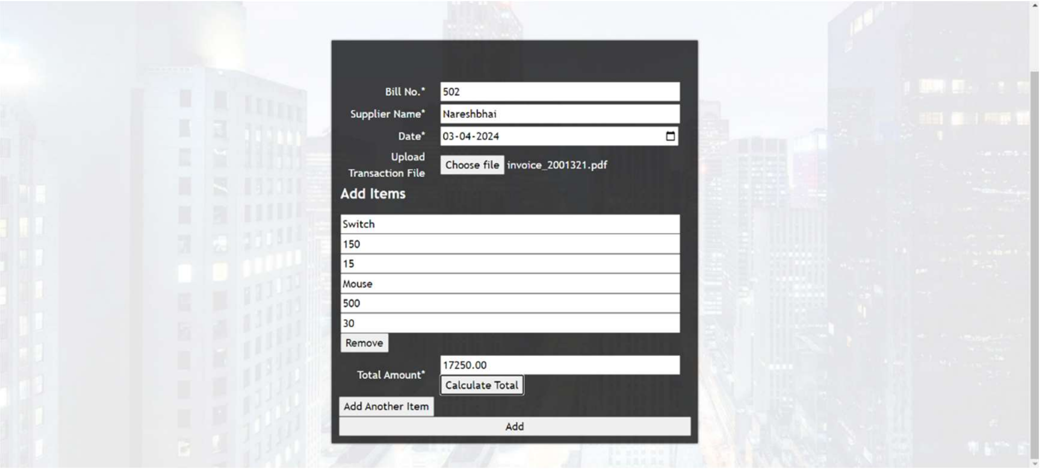
- **Materials Assets Page**



Materials Assets

Bill-No	Supplier Name	Date	Items	Quantity	Price	Total price	Assign		
502	Nareshbhai	2024-04-03	Switch	15	150	Rs 2250	0	✕	View
502	Nareshbhai	2024-04-03	Mouse	30	500	Rs 15000	0	✕	View

- **Add Materials Asset**



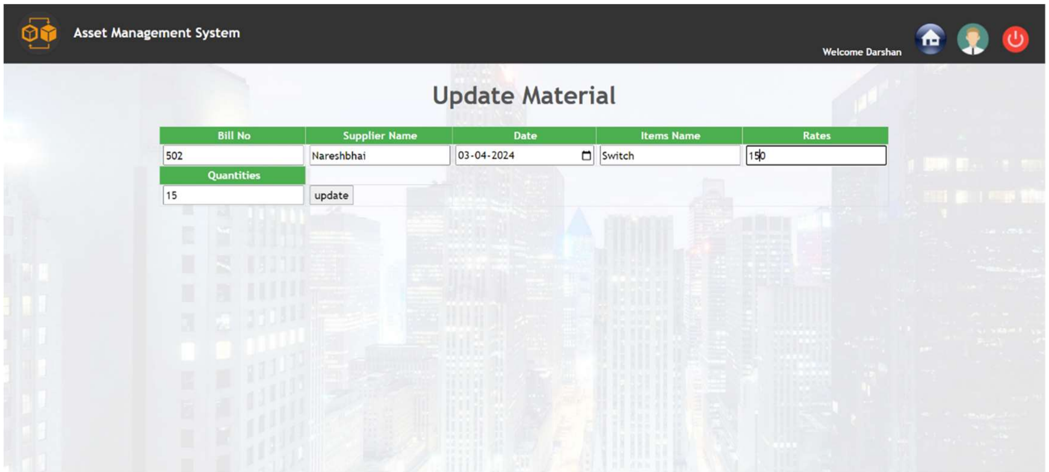
Bill No.*	502
Supplier Name*	Nareshbhai
Date*	03-04-2024
Upload Transaction File	Choose file invoice_2001321.pdf

Add Items

Switch
150
15
Mouse
500
30

Total Amount* 17250.00

- **Update Material Asset**

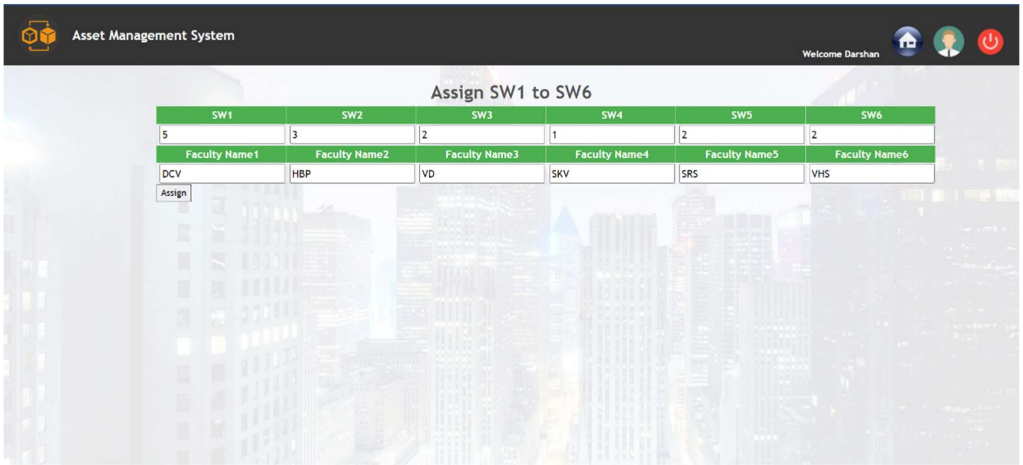


Update Material

Bill No	Supplier Name	Date	Items Name	Rates
502	Nareshbhai	03-04-2024	Switch	150

Quantities	
15	update

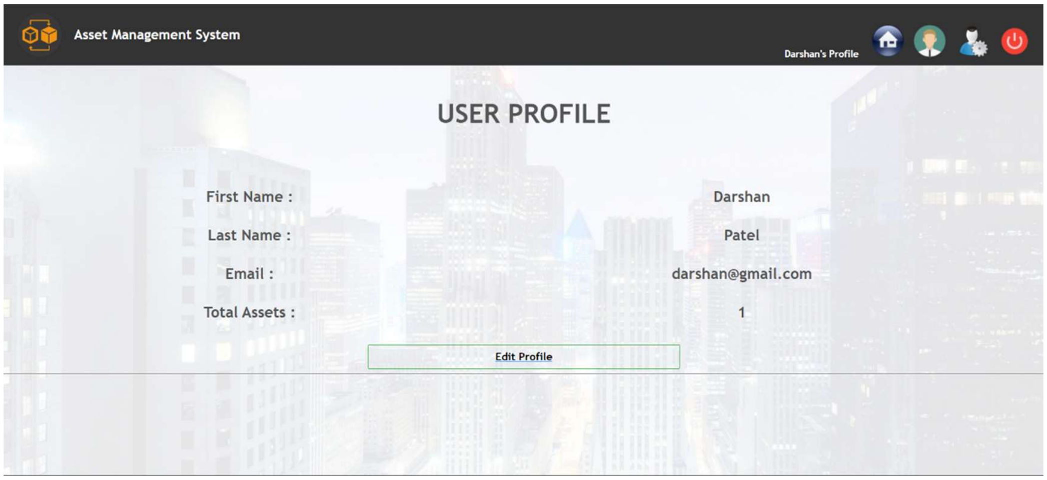
- **Assign Material Asset**



Assign SW1 to SW6

SW1	SW2	SW3	SW4	SW5	SW6
5	3	2	1	2	2
Faculty Name1	Faculty Name2	Faculty Name3	Faculty Name4	Faculty Name5	Faculty Name6
DCV	HBP	VD	SKV	SRS	VHS

- **User Profile**



USER PROFILE

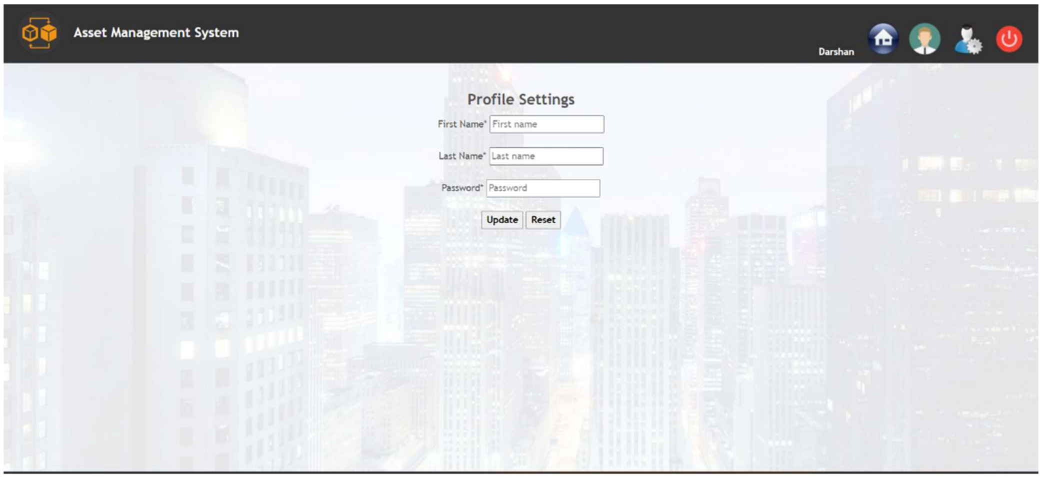
First Name : Darshan

Last Name : Patel

Email : darshan@gmail.com

Total Assets : 1

- **Edit User Profile**



Profile Settings

First Name* First name

Last Name* Last name

Password* Password

9. LIMITATION AND FUTURE ENHANCEMENT

Limitation:

- **Assigning Constraints:** Constraints on assigning assets to labs or users may not accommodate complex scenarios, such as temporary assignments or shared resources, leading to limitations in asset utilization.

- **Export Data Format:** Exporting data to Excel may be restricted to predefined formats or templates, limiting flexibility in formatting or analyzing exported data.

- **Limited Features:** While the website offers essential features like add asset, update and delete asset, assign asset, generate report, and export data in excel, it may lake advanced functionalities that could enhance user experience and engagement.

- **Scalability Concerns:** As the user base grows or if the website expands to accommodate multiple departments, scalability could become a concern. The website's infrastructure and architecture may need upgrades to handle increased traffic and data processing.

Future Enhancement:

- **Enhanced Asset Tracking:** Implementing barcode technology for automated asset tracking to improve accuracy and efficiency in asset management process.

- **Advanced Reporting Features**: Enhancing reporting capabilities with customizable dashboards trend analysis tools, and predictive analytics to provide deeper insights into asset usage and performance.

- **AI-driven Insights**: Leveraging artificial intelligence and machine learning algorithms to analyze asset data, identity patterns, and make proactive recommendations for optimizing asset utilization and lifecycle management.

- **Mobile Application Development**: Developing a mobile application feature would improve accessibility and efficiency by allowing users to perform asset management tasks on-the-go, facilitating real-time updates, asset assignments, and status checks from anywhere with internet connectivity, enhancing overall workflow flexibility and responsiveness.

- **APIs for Integration**: Providing robust APIs (Application Programing Interfaces) for seamless integration with third-party systems, enabling interoperability and data exchange with other IT tools and platforms.

10. CONCLUSION AND DISCUSSION

10.1 Conclusions and Future Enhancement

According to us, this project gave all of us the confidence to believe in ourselves and a great experience of how to work as a team. It also boosted our requirement gathering, system analysis, designing aspects, technical coding as well as time management skills.

Also, we learned how to work together as a team & collaborate for making ends meet for our website. We also got an insight into how we would have to work in future at job or startup & how we have to contribute for the good of the entity we are working.

10.2 Discussion

10.2.1 Self-Analysis of Project Viabilities

Project Viabilities undergoes self-analysis to assess its feasibility, impact, and sustainability. It's technically feasible with the right technology and expertise, but securing adequate funding and streamlining operational processes are crucial. The project aims to foster mentorship connections, support student development, and build a strong community. To ensure sustainability, sustainable funding sources, regular updates, and strong community engagement are essential. With these considerations, Project Viabilities can successfully foster mentorship connections and support student development in the long term.

10.2.2 Problem Encountered and Possible Solutions

There were so many problems encountered during this project: -

1. Problem to maintain database and to fetch data with complex queries.

2. Need to change some functionality fully which leads to doing the whole work again.

3. For making UI interactive, there were some errors on front-end part.

4. Exporting data to Excel may encounter compatibility issues with different versions of Excel software, resulting in formatting errors or data loss.

5. Some authentication glitches in backend and database part.

10.2.3 Summary of Project Work

Our project, developed through the application of software engineering and system analysis and design methodologies, has been successfully completed. We meticulously adhered to planned schedules, ensuring timely progress in project development while maintaining a focus on delivering tangible results. Through this endeavor, we gained invaluable industrial exposure, enriching our understanding of our field and refining our skills. The experience afforded us extensive learning opportunities, particularly in the realms of new technologies and effective time management practices. Ultimately, we surpassed our targets and goals outlined in the initial requirement specifications, affirming the successful culmination of our efforts.

At last, we extend our sincere gratitude to Prof. Deepak C. Vegda for his invaluable guidance and mentorship throughout this project journey. His expertise, support, and encouragement have been instrumental in guiding us towards successful project completion. Thank you, Prof. Deepak C. Vegda, for generously sharing your knowledge and insights, which have undoubtedly enriched our learning experience.

REFERENCES:

- w3schools: https://www.w3schools.com/
- geeksforgeeks: https://www.geeksforgeeks.org/
- PHP: https://www.php.net/
- stack overflow: https://stackoverflow.com/