PROJECT REPORT

ON

ASSET MANAGEMENT SYSTEM

By

PATEL VEDANT	**(2302250210288)**
PRANAY BHARTIYA	**(2302250210314)**
RANA DEV	**(2302250210320)**
SHAH HARIT	**(2302250210349)**
SHAH MOKSH	**(2302250210354)**
SHAH PARSHVA	**(2302250210356)**

DEPARTMENT OF COMPUTER ENGINEERING

LJ POLYTECHNIC, AHMEDABAD

2025-26

DEPARTMENT OF COMPUTER ENGINEERING
LJ POLYTECHNIC, AHMEDABAD

2025-26

CERTIFICATE

Date: ____/____/_____

This is to certify that ***Mr. Patel Vedant, Mr. Pranay Bhartiya, Mr. Rana Dev, Mr. Shah Harit, Mr. Shah Moksh and Mr. Shah Parshva*** from LJ POLYTECHNIC having Enrollment No. ***23012250210288, 23012250210314, 23012250210320, 23012250210349, 23012250210354 and 23012250210356*** have completed project documentation and partial development on the problem definition of semester V during the academic year 2025-26 having Title ***Asset Management System*** in a group consisting of 6.

Institute Guide **Head of the Department**

ACKNOWLEDGEMENT

We would thank God for being able to complete this project with success.Then We would like to thank our faculty whose valuable guidance has been the ones that helped us patch this project and make it full proof success his suggestions and his instructions has served as the major contributor towards the completion of the project. Secondly, we would also like to thank our parents who supported in our project.

PATEL VEDANT (2302250210288)
PRANAY BHARTIYA (2302250210314)
RANA DEV (2302250210320)
SHAH HARIT (2302250210349)
SHAH MOKSH (2302250210354)
SHAH PARSHVA (2302250210356)

Table of Contents

TABLE INDEX

FIGURE INDEX

ABSTRACT

Managing assets like laptops, desktops, and printers is crucial for any organization, and a web-based Asset Management System (AMS) offers a more efficient and accurate alternative to Excel or manual records. The AMS centralizes asset tracking by recording details such as type, location, condition, status, and assigned employee, with assets organized into categories like laptops or accessories for easier management. Role-based access control ensures users have appropriate permissions—some can only view records, while others can add, edit, or delete—maintaining data security and integrity. Accessible from any internet-connected device and requiring no technical expertise, the system eliminates the risk of data loss, reduces spreadsheet dependency, and supports better tracking, maintenance, and decision-making.

CHAPTER-1
INTRODUCTION

1.1 Need of the system

1.1.1 Challenges in Manual Tracking

- Tracking assets like laptops, desktops, printers, and accessories manually is time-consuming.
- Spreadsheets or manual methods often lack accuracy and accessibility.

1.1.2 Advantages of the Web-Based System

- Centralized storage and management of asset details.
- Supports asset assignment, condition updates, and attaching documents (bills, warranty cards).

1.1.3 Enhanced Monitoring

- Tracks current status and ownership of assets.
- Provides visual tools like graphs and charts for usage, distribution, and life-cycle trends.

1.2 Detailed problem definition

1.2.1 Limitations of Manual Management

- Manual tracking is inefficient and leads to missing or incorrect data.
- Asset-related documents are scattered, making retrieval difficult.

1.2.2 Lack of Analytical Tools

- Absence of visual reports makes it hard to analysis asset growth and ownership history.
- No consolidated view of asset condition and life-cycle.

1.2.3 Need for an Integrated Solution

- A secure system is required for asset tracking, document storage, and reporting.
- Simplifies asset management and ensures reliability of information.

1.3 Viability of the system

1.3.1 Applicability

- Suitable for both individuals and organizations to improve asset control.
- Useful for managing IT assets such as laptops, desktops, printers, and accessories.

1.3.2 Features

- Assets can be assigned to responsible persons with condition updates.
- Provides reports and graphs on asset value, changes over time, and category distribution.

1.4 Presently available system

1.4.1 Existing Platforms

- https://www.assetpanda.com/
- https://www.zoho.com/

1.5 Future prospects

1.5.1 Predictive Maintenance & Automation

- Alerts for equipment that may require servicing or replacement.
- Automated reminders for warranty expiration's, license renewals, and updates.

1.5.2 Integration & Accessibility

- Integration with bank platforms and inventory systems for automatic updates.
- Mobile access with QR/bar-code scanning for quick identification and tracking.

CHAPTER-2
ANALYSIS

2.1 Requirement Analysis

2.1.1 Types of Users

- Admin
- IT Supervisor
- Employee (Asset User)
- Vendor (Supplier)
- Maintenance Staff

2.1.2 Admin

Manage all aspects of the system

- **Manage Users, Roles, and Permissions -** Add, edit, or remove users. Assign roles and set access levels.
- **Add/Edit/Delete Assets -** Maintain asset records including IT hardware and software.
- **Assign/Unassign Assets -** Allocate assets to employees and track returns.
- **Track Asset History and Reports -** Monitor usage, condition, and generate analytics.
- **Manage Maintenance -** Schedule and track asset maintenance activities.
- **Approve/Reject Vendor Quotes -** Review incoming quotes and decide on procurement.
- **View Dashboard and Analytics -** Get an overview of all assets, reports, and system health.

2.1.3 IT Supervisor:

Manage departmental assets and activities

- **View and Manage Assigned Assets –** Monitor assigned assets within departments.
- **Assign Assets to Employees –** Allocate IT resources according to requirements.
- **Manage Locations and Rooms –** Keep records of where assets are located.
- **Track Usage and Activity Logs –** Ensure proper usage and monitor activity.
- **Request Maintenance Support –** Identify faulty assets and coordinate repairs.
- **Generate Departmental Reports –** Summarize asset usage and condition for management.

2.1.4 Employee

Use and report on assigned assets

- **View Assigned Assets –** Check the list of assets allocated to them.
- **Report Asset Condition or Issues –** Notify if any asset is damaged or not working properly.
- **Request New Assets or Software Access –** Submit requests for additional resources.
- **View Request Status –** Track progress of submitted requests.
- **Edit Profile –** Update personal information and contact details.

2.1.5 Vendor

Supply and maintain assets

- **Provide Assets** – Supply IT hardware or software as needed.

- **Replace or Upgrade** – Handle faulty or outdated equipment.
- **Upload Documents** – Share bills, warranties, and supply details.
- **Update Status** – Track and update delivery progress.

2.1.6 Maintenance Staff

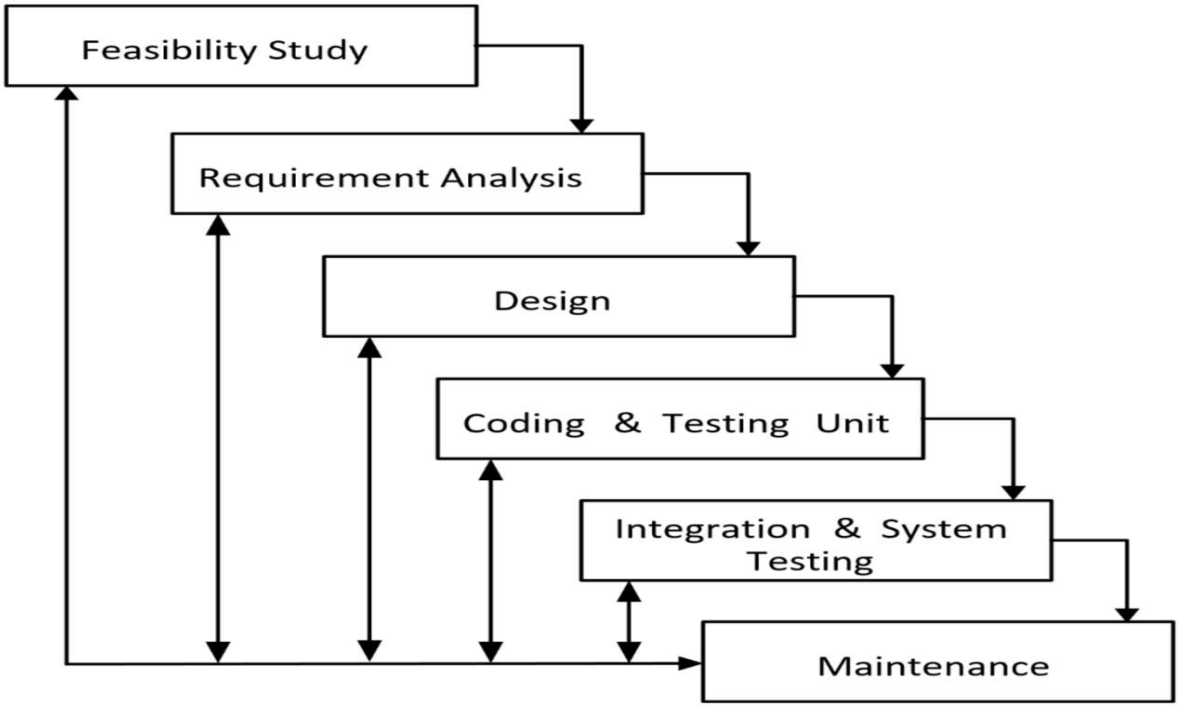
Configure, repair, and maintain assets

- **Configure Assets** – Install and set up new systems.
- **Repair or Replace** – Fix or change damaged assets.
- **Maintenance Check** – Perform regular inspections.
- **Update Records** – Log repair and maintenance details.

2.2 Project Model

2.2.1 Iterative Waterfall Model

- Combines structured planning with flexibility to update after feedback.
- Centralizes hardware/software management securely and at scale.
- Ensures role-based access for different users (Admin, Supervisor, Employee, Vendor, Maintenance).
- Improves accountability with transparent workflows, approvals, and reporting.



[Figure 1: Iterative Waterfall Model]

2.2.2 Advantages of Iterative Waterfall Model

- Provides a clear, structured, and sequential development approach.
- Allows changes in later phases without restarting the whole project.
- Incorporates user feedback at each stage to improve usability.
- Early detection and fixing of issues due to iterative testing.
- Supports scalability by allowing addition of new modules or features

2.3 Schedule Representation

2.3.1 Techniques Used: PERT and CPM

- PERT (Program Evaluation & Review Technique) – Estimates time and effort for tasks, helps in planning realistic timelines.
- CPM (Critical Path Method) – Identifies critical tasks and ensures the project finishes on time.

2.3.2 Benefits of Schedule Techniques

- Breaks project into smaller, manageable tasks.

- Organizes tasks step-by-step for smooth execution.

[Table 1: Schedule Representation]

ACTIVITY	START DATE	FINISH DATE
Requirement Analysis		
System Analysis		
System Design		
Coding		
Testing		

2.4 Feasibility Study

2.4.1 Technical Feasibility

- Frontend: HTML, CSS, JavaScript, React.js for fast and user-friendly interfaces.
- Backend: Node.js with Express.js ensures scalable and efficient processing.
- Database: MySQL for secure, structured, and reliable data storage.
- Open-source stack reduces dependency on costly licensed software.

2.4.2 Economical Feasibility

- Uses free, open-source tools → cost-effective with no licensing fees.
- Saves resources by replacing manual record-keeping with automation.
- Reduces paperwork, errors, and delays, improving efficiency.

2.4.3 Operational Feasibility

- Easy adoption by Admin, Supervisors, and Employees through role-based dashboards.

- Improves overall workflow by automating repetitive tasks like maintenance reminders.

2.4.4 Schedule Feasibility

- Estimated completion within 3 months using PERT/CPM planning.
- Allows buffer time for testing and incorporating user feedback.

CHAPTER-3
DESIGN

3.1 Data Flow Diagram

3.1.1 Definition and Purpose

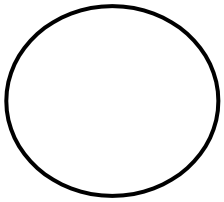
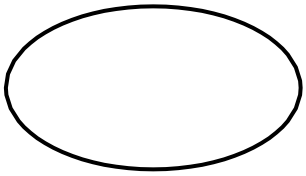
- What is a DFD – Also known as a bubble chart or data flow graph, represents how data moves through a system.
- Importance – Helps understand system functionality and is effective during analysis.

3.1.2 Features of DFD

- Hierarchical graphical model showing processing activities/functions and data interchange.
- Views a system as a function that transforms inputs into desired outputs.
- Each process consumes input data and produces output data.
- Represents input data, processing activities, and output data in a clear visual format.
- Functional model can be represented using flow diagrams for easy understanding.

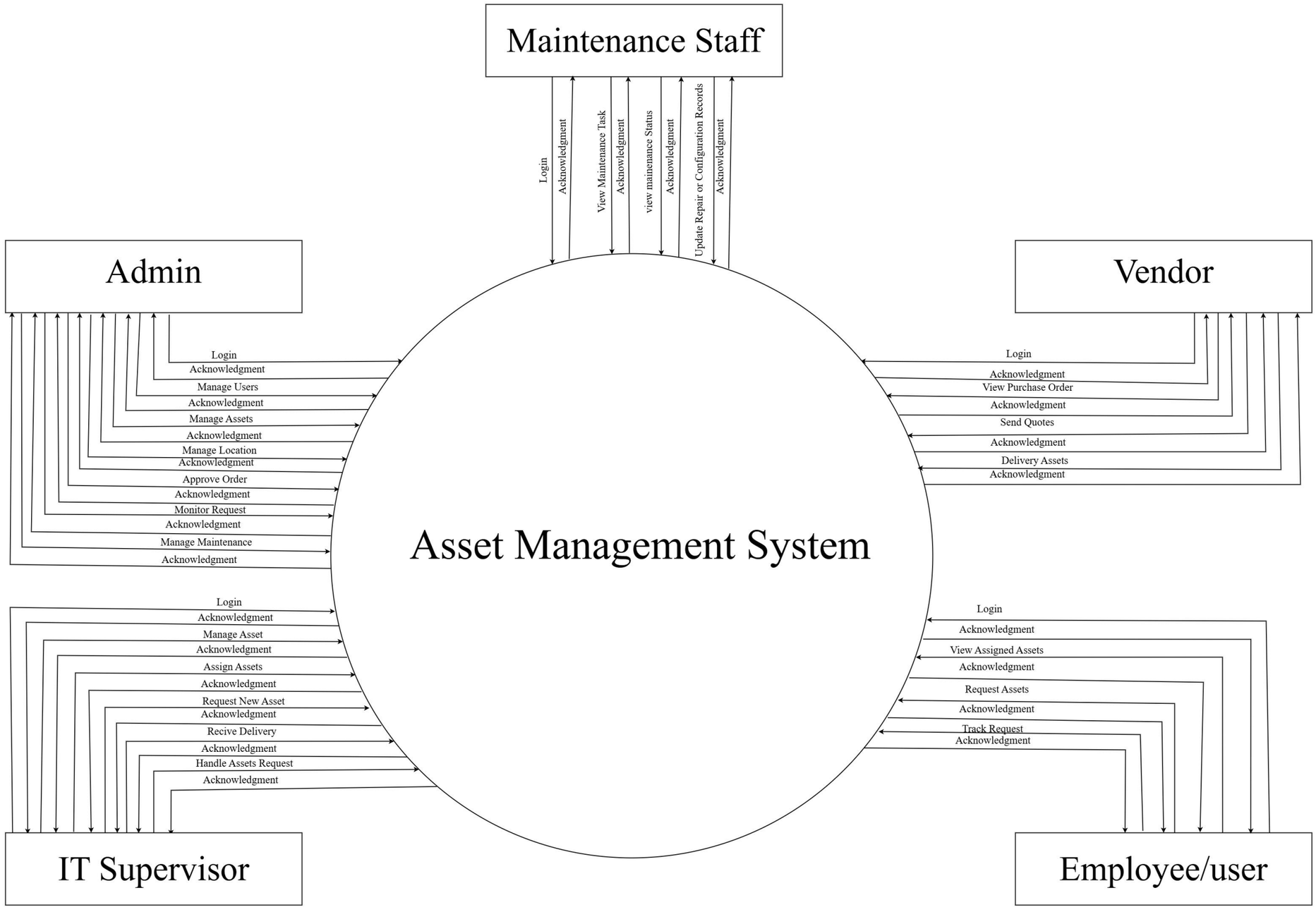
3.1.3 DFD Symbols

[Table 2: Data Flow Diagram Symbols]

Symbols	Description
	**Entity:** Entities are external to the system which interacts by inputting the data.
	**System:** It shows the system name.
	**Process:** It shows the part of the system that transforms into outputs.
	Data Flow: It passes the data from one part to another.
	**Data Store:** Data store is represented by two parallel lines. It is generally logical file or database.

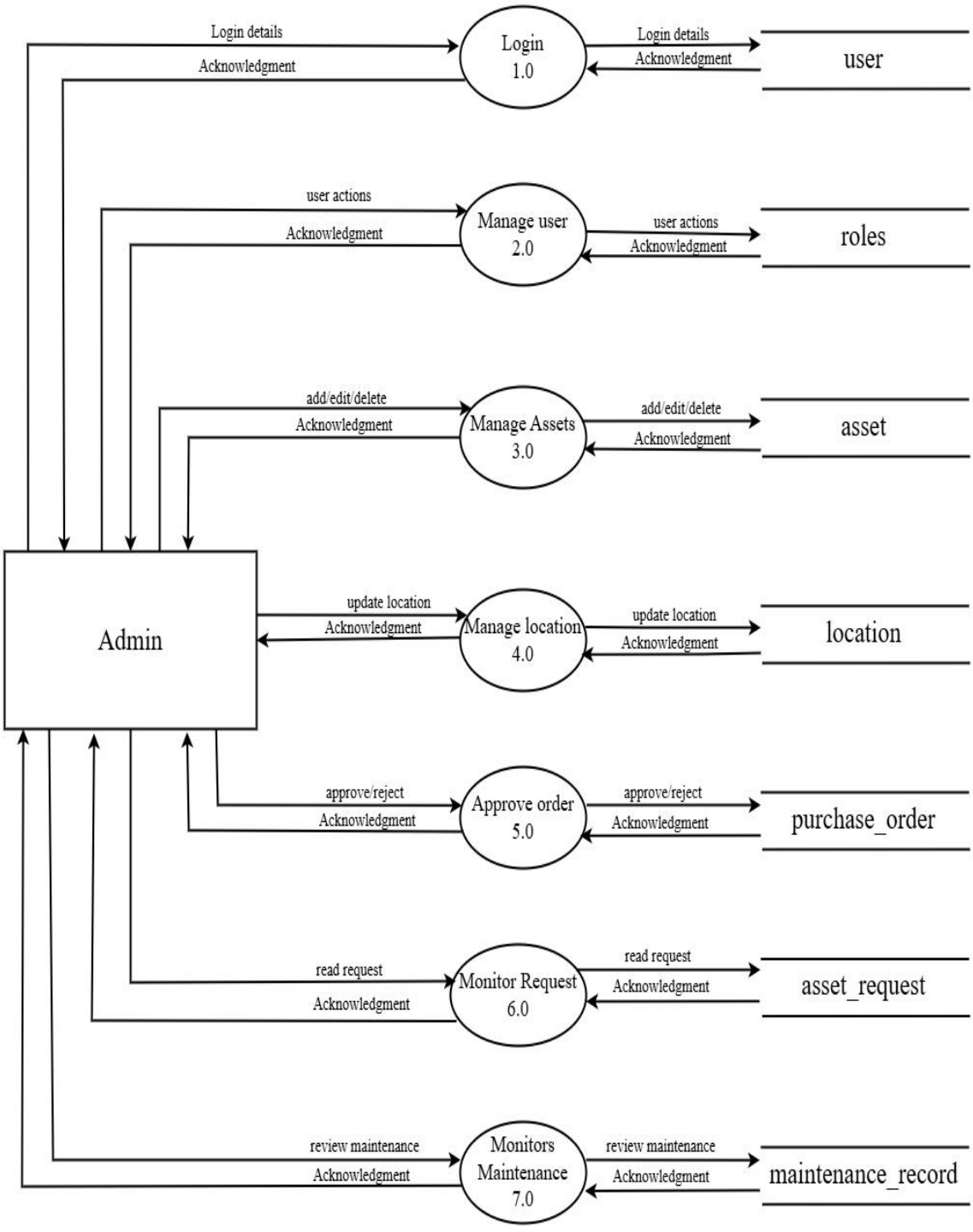
3.1.4 Levels of DFD

Level 0: Context



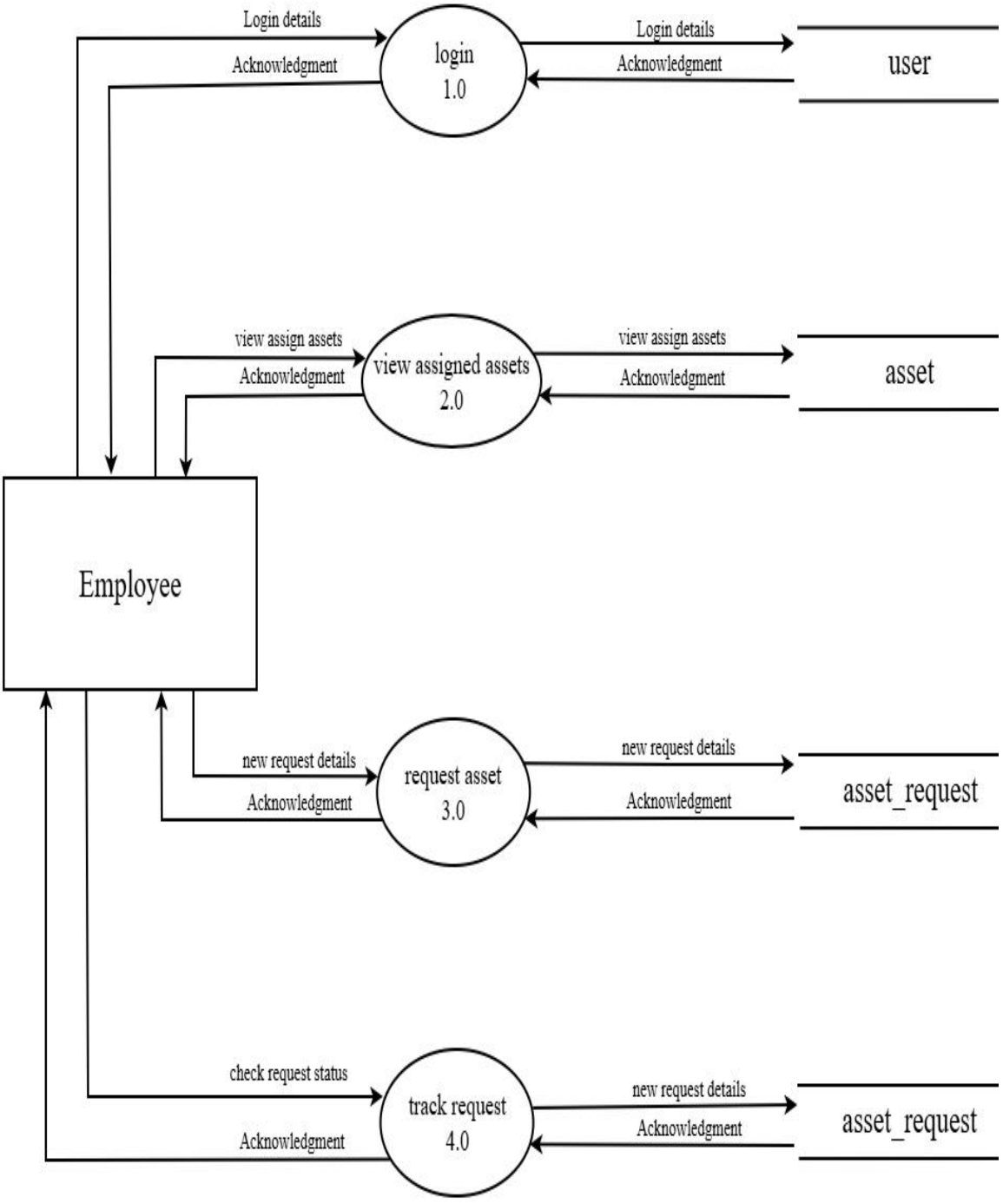
[Figure 2: DFD LEVEL 0: Context Level]

Level 1: Admin



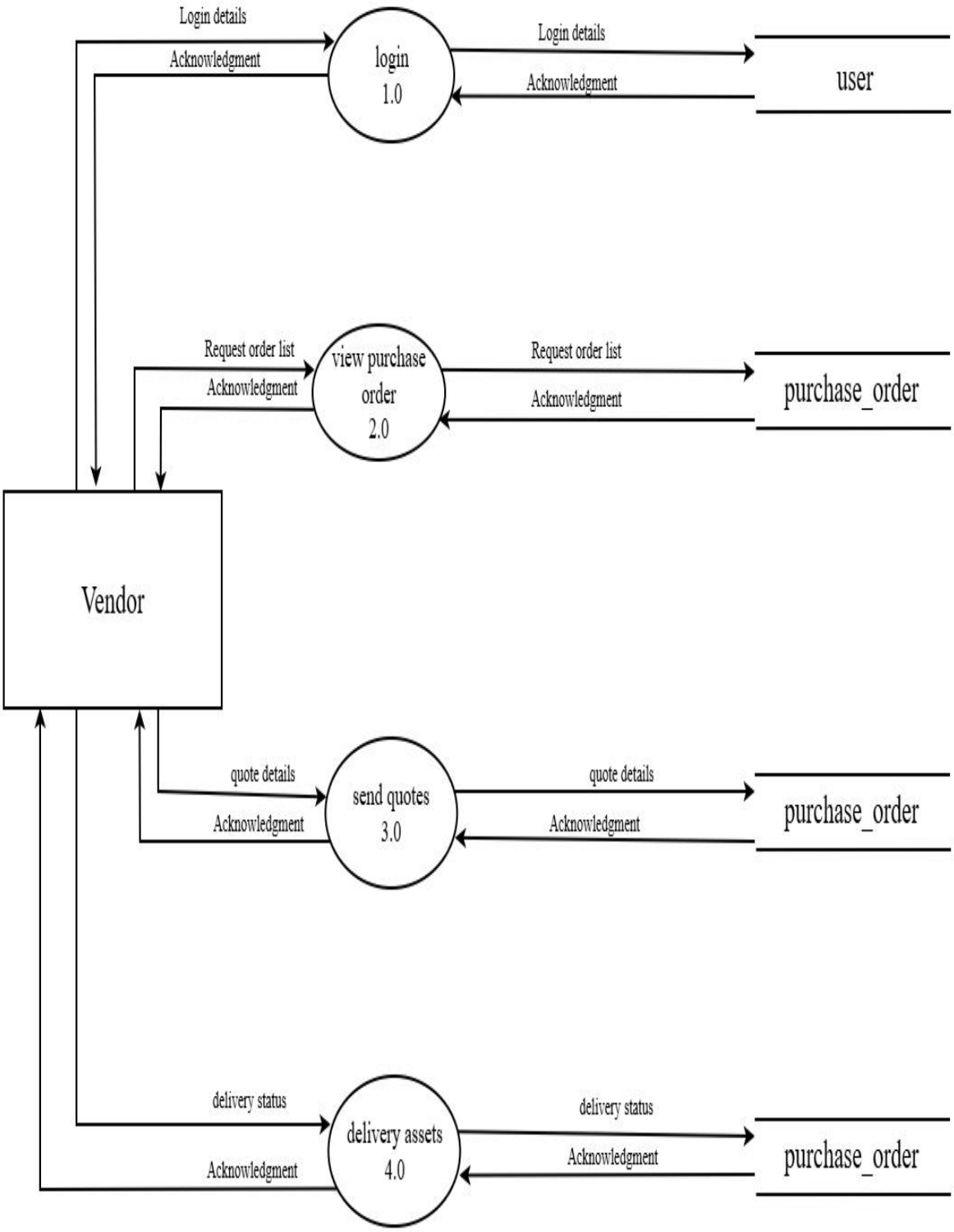
[Figure 3: DFD Level 1: Admin]

Level 1: Employee/User



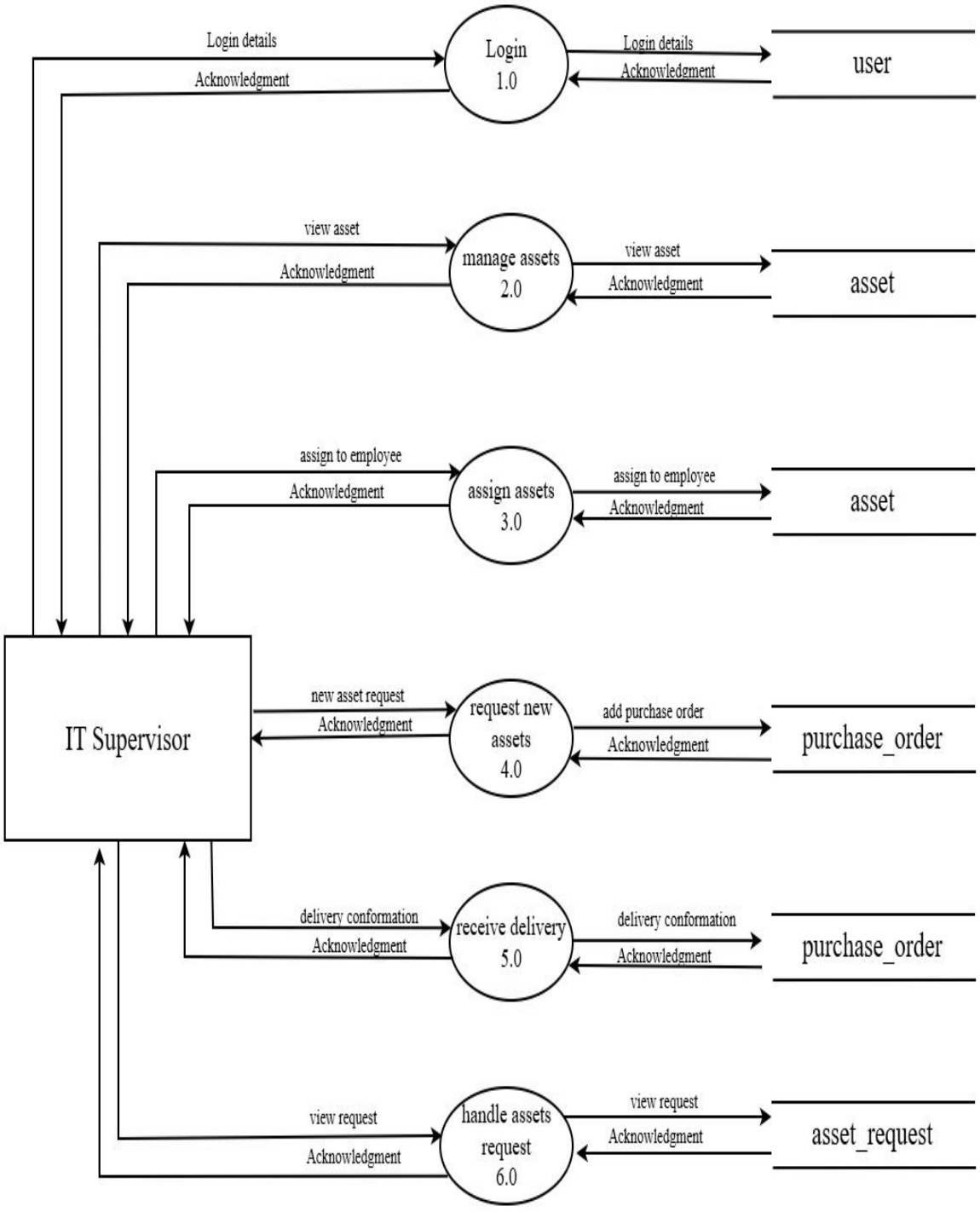
[Figure 4: DFD Level 1: Employee/User]

Level 1: Vendor



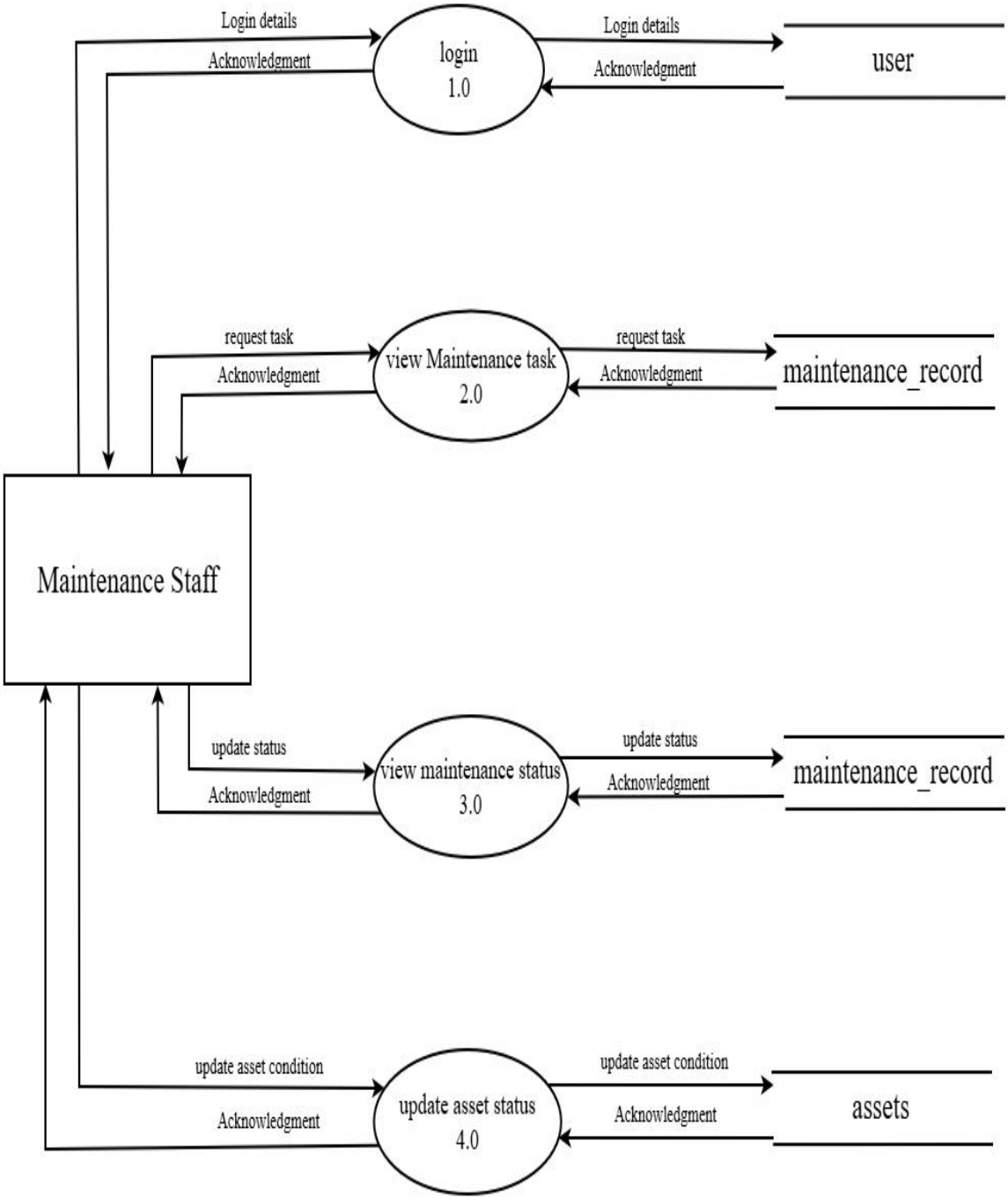
[Figure 5: DFD Level 1: Vendor]

Level 1: IT Supervisor



[Figure 6: DFD Level 1: IT Supervisor]

Level 1: Maintenance Staff



[Figure 7: DFD Level 1: Maintenance Staff]

3.2 ER-Diagram

3.2.1 Definition and Purpose

- What is an ER Diagram – A visual representation of the data structure in a database.
- Importance – Models entities, their attributes, and relationships helps in database design.

3.2.2 Features of ER-Diagram

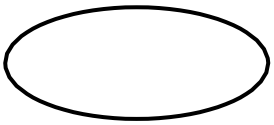
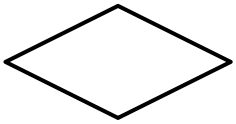
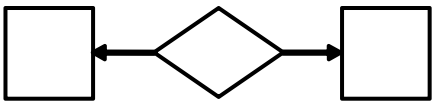
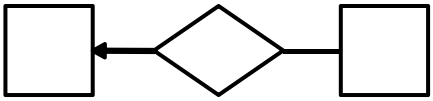
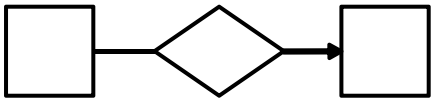
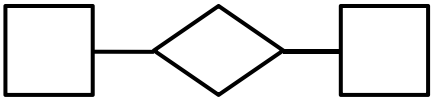
- Provides a clear, structured view of how components interact.
- Illustrates relationships between entities in the system.
- Structured analysis technique – shows logical data design that can be converted to tables.
- ERD is a snapshot of data structure and a detailed logical representation of the system.
- Model's data and relationships but does not describe how the data is processed.

3.2.3 Main Elements of ER-Diagram

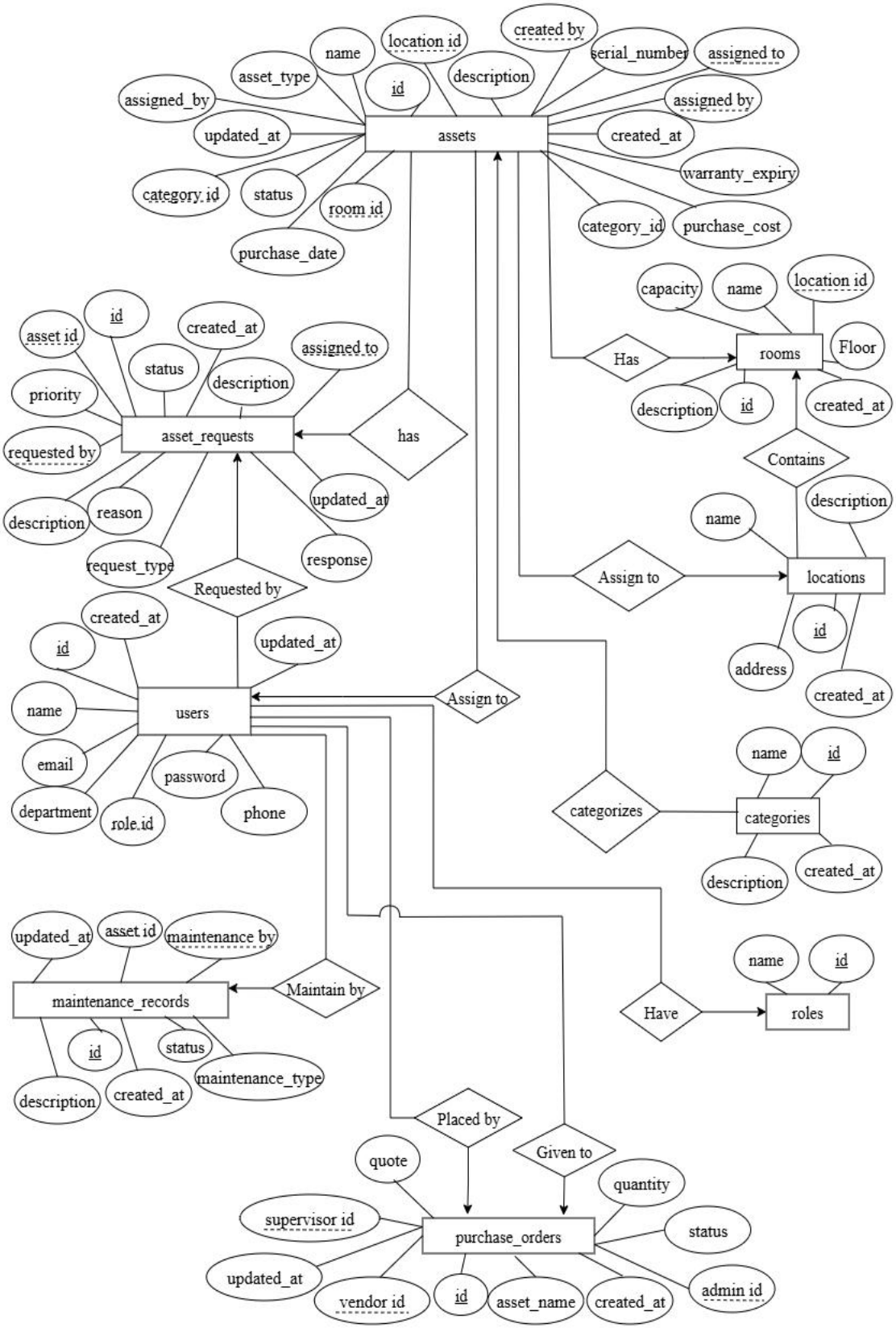
- Entity (Data Object) – Represents a real-world object or concept, e.g., User, Asset.
- Attributes – Properties of entities, e.g., Asset ID, Name, Condition.
- Relationships – How entities are linked, e.g., Employee "assigned" Asset, Vendor "supplies" Asset.

3.2.4 ER-Diagram Symbols

[Table 3: ER-Diagram Symbols]

Symbols	Description
	**Entity:** Data object is real world entity or thing. It is represented by a rectangle shape. An entity is an object or concept about which you want to store information.
	**Attributes:** An attribute is property of characteristic of an entity. It is represented by oval shape.
	**Relationship:** Entity are connected each other via relations. Generally, relationships in binary because there are two entities are related to each other.
	**Cardinality (One to One):** An instance of entity A can relate to one instances of entity B.
	**Cardinality (One to Many):** An instance of entity A can relate to one or many instances of B but we can only relate one instance of A.
	**Cardinality (Many to One):** One or more instances of entity A can relate to one instances of B.
	**Cardinality (Many to Many):** One or more instances of entity A can relate to one more instance of entity B.

3.2.5 ER-Diagram:



[Figure 8: ER-Diagram]

CHAPTER-4
SYSTEM MODELING

4.1 Database Dictionary

4.1.1 Table Name: Roles

Primary Key: id

[Table 4: Roles]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies the role id.
2	name	Varchar(50)	Unique, Not Null	It specifies the role name.

4.1.2 Table Name: Users

Primary Key: id

Foreign Key: role id

[Table 5: Users]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies user id.
2	name	varchar(100)	Not Null	It specifies user's full name.
3	email	varchar(100)	Unique, Not Null	It specifies user's email address.
4	password	varchar(100)	Not Null	It specifies user's password.
5	role id	int(10)	Foreign Key	It specifies role id.
6	department	varchar(50)	Null	It specifies department.
7	phone	varchar(25)	Null	It specifies phone number.
8	created_at	timestamp	Default	It shows the time it is created.
9	updated_at	timestamp	Default	It shows the time it is updated.

4.1.3 Table Name: Categories

Primary Key: id

[Table: 6 Categories]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies category id.
2	name	varchar(100)	Not Null	It specifies category name.
3	description	varchar(100)	Null	It specifies description.
4	created_at	timestamp	Default	It shows the time it is created.

4.1.4 Table Name: Locations

Primary Key: id

[Table: 7 Locations]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies location id.
2	name	varchar(100)	Not Null	It specifies location name.
3	address	varchar(100)	Null	It specifies location address.
4	description	varchar(100)	Null	It specifies description.
5	created_at	timestamp	Default	It shows the time it is created.

4.1.5 Table Name: Rooms

Primary Key: id

Foreign Key: location id

[Table: 8 Rooms]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies room id.
2	name	varchar(100)	Not Null	It specifies room name.
3	floor	varchar(25)	Null	It specifies floor.
4	capacity	int(10)	Null	It specifies capacity.
5	description	varchar(100)	Null	It specifies description.
6	location id	int(10)	Foreign Key	It specifies location id.
7	created_at	timestamp	Default	It shows the time it is created.

4.1.6 Table Name: Assets

Primary Key: id

Foreign Key: category id, location id, room id, assigned to, assigned by, created by

[Table: 9 Assets]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies asset id.
2	name	varchar(100)	Not Null	It specifies asset name.
3	description	varchar(100)	Null	It specifies description.
4	serial_number	varchar(100)	Unique	It specifies serial number.
5	category id	int(10)	Foreign Key	It specifies category id.
6	location id	int(10)	Foreign Key	It specifies location id.
7	room id	int(10)	Foreign Key	It specifies room id.
8	status	enum	Default 'Available'	It specifies asset status.
9	asset_type	enum	Not Null	It specifies asset type.
10	purchase_date	date	Null	It specifies purchase date.
11	warranty_expiry	date	Null	It specifies warranty expiry date.
12	purchase_cost	decimal(10,2)	Null	It specifies purchase cost.
13	assigned to	int(10)	Foreign Key	It specifies assigned to user id.

14	assigned by	int(10)	Foreign Key	It specifies assigned by user id.
15	created by	int(10)	Foreign Key	It specifies created by user id.
16	created_at	timestamp	Default	It shows the time it is created.
17	updated_at	timestamp	Default	It shows the time it is updated.

4.1.7 Table Name: Asset_requests

Primary Key: id

Foreign Key: requested by, asset id, assigned to

[Table: 10 Asset_requests]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies request id.
2	asset id	int(10)	Foreign Key	It specifies asset id.
3	request_type	enum	Not Null	It specifies request type.
4	reason	varchar(100)	Null	It specifies reason.
5	description	varchar(100)	Null	It specifies description.
6	priority	enum	Default 'Medium'	It specifies request priority.
7	status	enum	Default 'Pending'	It specifies request status.
8	requested by	int(10)	Foreign Key	It specifies requested by user id.
9	assigned to	int(10)	Foreign Key	It specifies assigned to user id.
10	response	varchar(100)	Null	It specifies response.
11	created_at	timestamp	Default	It shows the time it is created.
12	updated_at	timestamp	Default	It shows the time it is updated.

4.1.8 Table Name: Purchase_orders

Primary Key: id

Foreign Key: supervisor id, vendor id, admin id

[Table: 11 Purchase_orders]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies purchase order id.
2	supervisor id	int(10)	Foreign Key	It specifies supervisor id.
3	vendor id	int(10)	Foreign Key	It specifies vendor id.
4	asset_name	varchar(100)	Not Null	It specifies asset name.
5	quantity	int(10)	Not Null	It specifies quantity.
6	quote	decimal(10,2)	Null	It specifies quoted cost.
7	status	enum	Default 'Requested'	It specifies order status.
8	admin id	int(10)	Foreign Key	It specifies admin id.
9	created_at	timestamp	Default	It shows the time it is created.
10	updated_at	timestamp	Default	It shows the time it is updated.

4.1.9 Table Name: Maintenance_records

Primary Key: id

Foreign Key: asset id, maintenance by

[Table: 12 Maintenance_records]

SR.NO	FIELD NAME	DATATYPE (SIZE)	CONSTRAINTS	DESCRIPTION
1	id	int(10)	Primary Key, Auto Increment	It specifies record id.
2	asset id	int(10)	Foreign Key	It specifies asset id.
3	maintenance by	int(10)	Foreign Key	It specifies user id.
4	maintenance_type	enum	Not Null	It specifies maintenance type.
5	description	varchar(100)	Null	It specifies description.
6	status	enum	Default 'Pending'	It specifies status.
7	created_at	timestamp	Default	It shows the time it is created.
8	updated_at	timestamp	Default	It shows the time it is updated.

CHAPTER-5
TECHNICAL SPECIFICATION

5.1 Hardware Specification

5.1.1 RAM: 8 GB

5.1.2 Hard Drive Storage: 20 GB

5.1.3 Other Hardware Requirements: None

5.2 Platform

5.2.1 Supported Operating System: Windows, Linux, Mac

5.2.2 Programming Server: Node.js (Express.js)

5.2.3 Framework: React.js (Frontend), Express.js (Backend)

5.3 Programming Language Used

5.3.1 Markup Language: HTML 5

5.3.2 Programming Language: CSS 3, JavaScript ES14

5.3.3 Scripting Language: Node.js

5.4 Technical Specification

5.4.1 Front End: HTML 5, CSS 3, JavaScript ES14, React.js

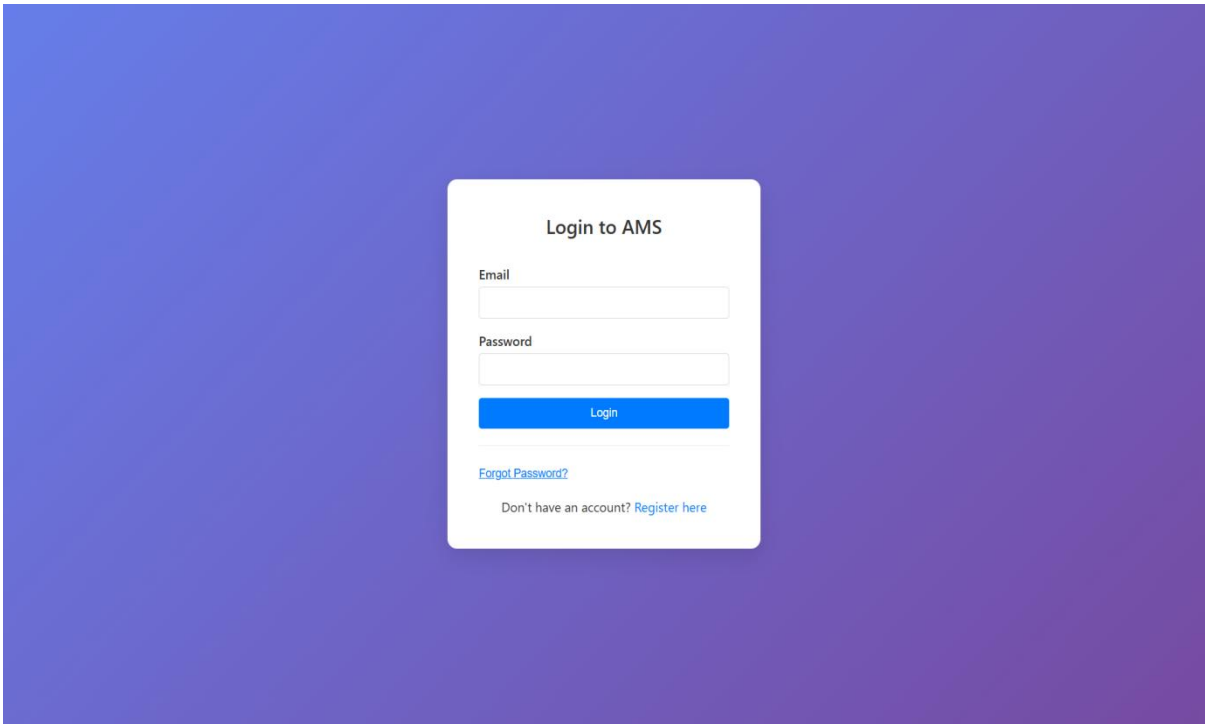
5.4.2 Back End: Node.js, Express.js, MYSQL 9.1.0

5.4.3 IDE: VISUAL STUDIO CODE 1.97

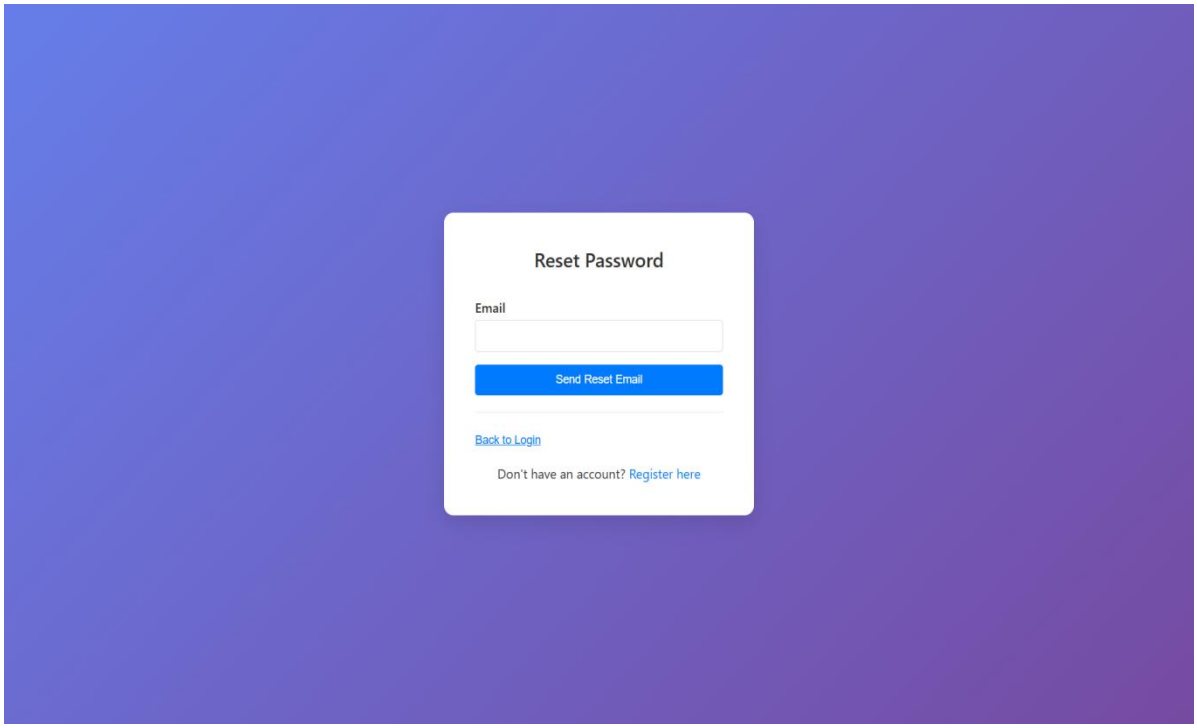
5.4.4 UML Tools: draw.io

5.4.5 SRS Tools: Microsoft Word 2019, WPS Document

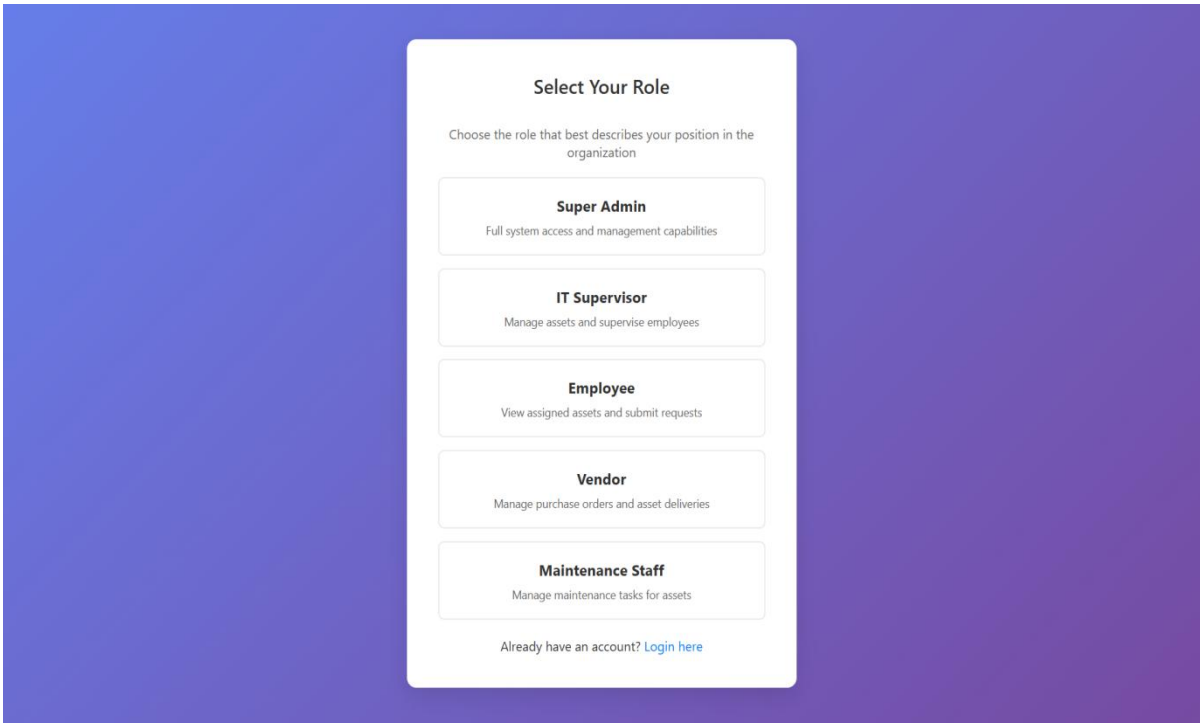
5.5 Design Layout



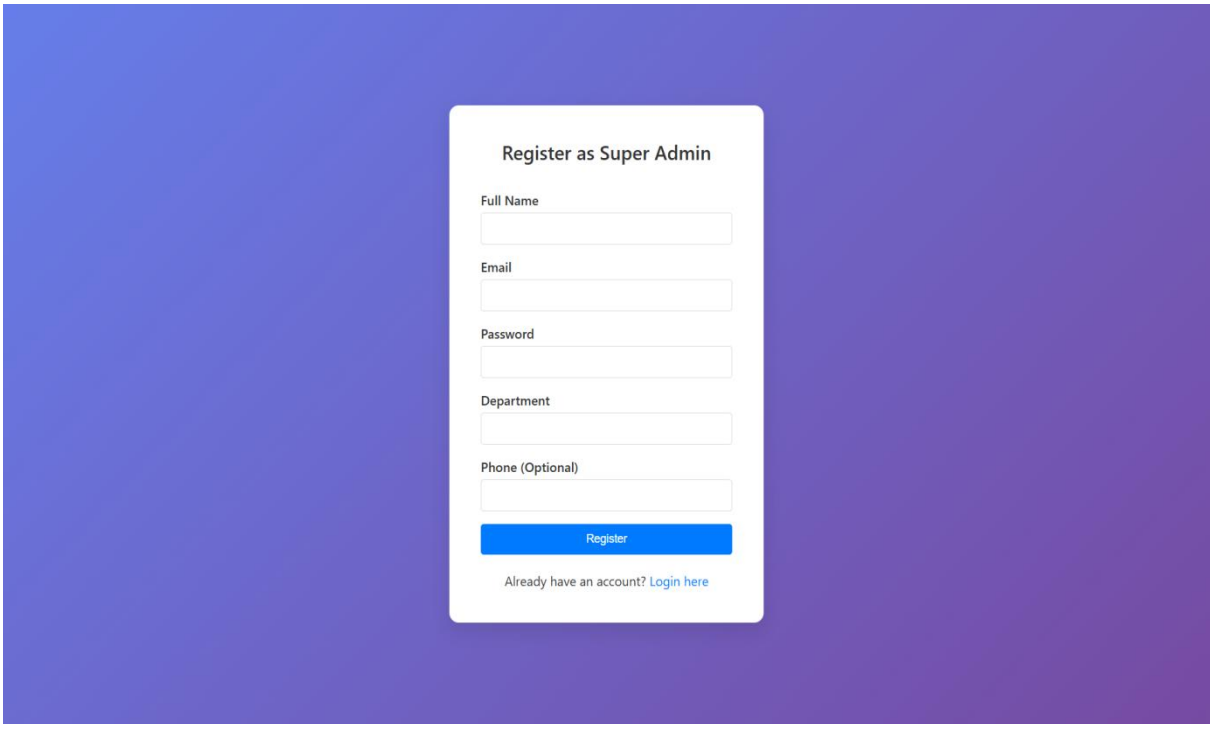
[Figure 9: Login]



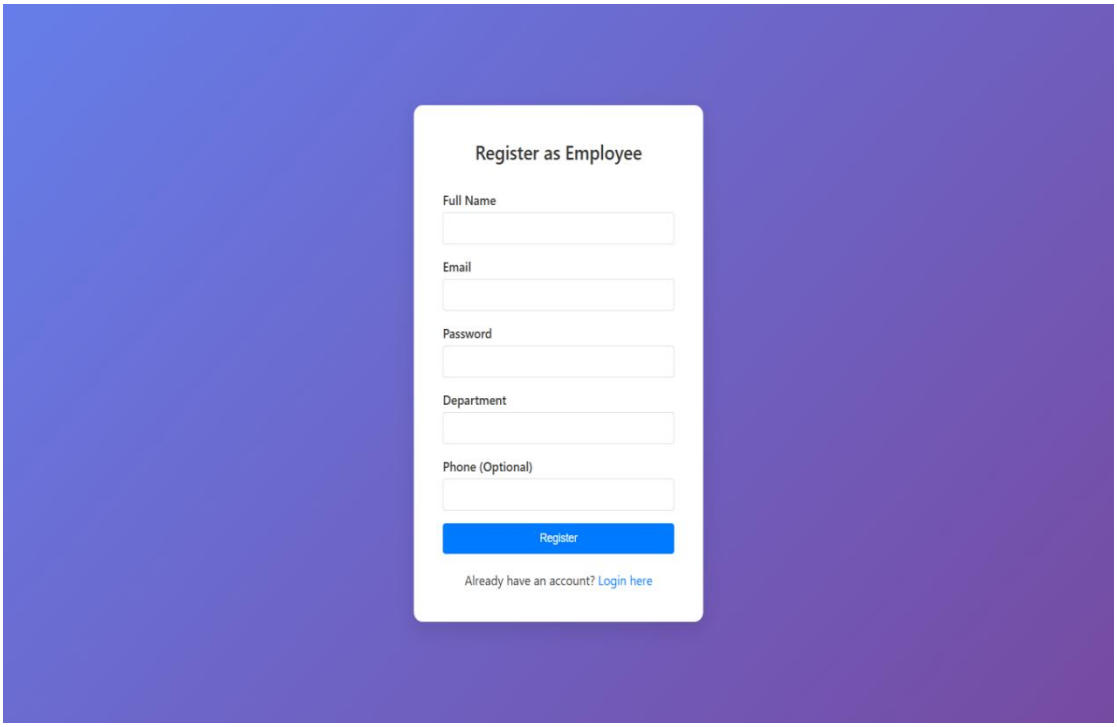
[Figure: Forget Password]



[Figure: Role Selection]



[Figure: Register as Admin]



[Figure: Register as Employee]

CONCLUSION

The Asset Management System is a comprehensive web-based solution that enables organizations to manage both hardware assets, such as laptops, desktops, and printers, and software assets like subscriptions in a secure and efficient way. It centralizes all asset records in one place and provides role-based access for Admins, Supervisors, Employees, Vendors, and Maintenance staff, each with clear responsibilities to ensure smooth operations from allocation and monitoring to ordering, approval, delivery, and maintenance. Supervisors can request assets, vendors provide quotes, admins approve them, and maintenance staff handle setup or repairs, making the workflow transparent and reliable. Key features include asset assignment, condition tracking, order and approval management, maintenance updates, and detailed reporting, all of which reduce errors, prevent data loss, and improve decision-making. With its simple and user-friendly interface, the system allows even non-technical users to manage assets easily while providing graphical insights into usage, distribution, and lifecycle trends. Overall, it balances functionality with simplicity, helping organizations save time, cut costs, improve accountability, and maintain accurate, organized records for both hardware and software assets.

BIBLIOGRAPHY

Book Reference:-

[1] Cantelon, Mike, et al. *Node.js in Action.* 2nd Edition, Manning Publications, 2017.

[2] Brown, Ethan. *Web Development with Node and Express.* O'Reilly Media, 2019.

Web Reference:-

[1] *https://www.geeksforgeeks.org*

[2] *https://expressjs.com*

[3] *https://nodejs.org/en/docs*

[4] *https://react.dev*

[5] *https://developer.mozilla.org/en-US/docs/Web/JavaScript*

[6] *https://dev.mysql.com*