**ERP SYSTEM**



Spring 2025

CSE-403L Database Management System Lab

Submitted by:

Fiza Nihar Roll No: 22pwcse2100

Afia Shah Roll No: 22pwcse2115

Hafsa Javed Roll No: 22pwcse2171

Class Section: B

“On my honor, as student of University of Engineering and Technology, I have neither given

nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Sumayyea Salahuddin**

Month Day, Year (01 06, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Table of Contents**

[1. Project Report: ERP System 4](#__RefHeading___Toc3006_3299793794)

[1. Project Title: ERP System 4](#__RefHeading___Toc3008_3299793794)

[2. Group Members: 4](#__RefHeading___Toc3010_3299793794)

[3. Objective: 4](#__RefHeading___Toc3012_3299793794)

[4. Technologies Used: 4](#__RefHeading___Toc3014_3299793794)

[5. System Architecture & Component Structure: 5](#__RefHeading___Toc3016_3299793794)

[1. Frontend Structure: 5](#__RefHeading___Toc3018_3299793794)

[2. Backend Structure (Laravel): 5](#__RefHeading___Toc3020_3299793794)

[6. Module Descriptions with Work Done: 5](#__RefHeading___Toc3022_3299793794)

[1. Employee Module: 5](#__RefHeading___Toc3024_3299793794)

[2. Category & Product Modules: 5](#__RefHeading___Toc3026_3299793794)

[3. Supplier Module: 6](#__RefHeading___Toc3028_3299793794)

[4. Sales Module: 6](#__RefHeading___Toc3030_3299793794)

[5. Attendance Module: 6](#__RefHeading___Toc3032_3299793794)

[6. Leave Module: 6](#__RefHeading___Toc3034_3299793794)

[7. API, Security & Token Handling: 6](#__RefHeading___Toc3036_3299793794)

[8. Testing & Validation Strategy: 6](#__RefHeading___Toc3038_3299793794)

[9. Documentation & Source Control: 7](#__RefHeading___Toc3040_3299793794)

[10. Deployment on EC2 on AWS Cloud 7](#__RefHeading___Toc3042_3299793794)

[11. Submission Summary: 14](#__RefHeading___Toc3044_3299793794)

[12. Screen Shots: 14](#__RefHeading___Toc3046_3299793794)

[1. Admin Login: 14](#__RefHeading___Toc3048_3299793794)

[2. Dashboard: 15](#__RefHeading___Toc3050_3299793794)

[13. Conceptual schema : 20](#__RefHeading___Toc3052_3299793794)

[1. Finalized Conceptual Schema (Entity-Level Description): 20](#__RefHeading___Toc3054_3299793794)

[1. Product 20](#__RefHeading___Toc3056_3299793794)

[2. Category 21](#__RefHeading___Toc3058_3299793794)

[3. Supplier 21](#__RefHeading___Toc3060_3299793794)

[4. Purchase 21](#__RefHeading___Toc3062_3299793794)

[5. Employee 21](#__RefHeading___Toc3064_3299793794)

[6. Department 22](#__RefHeading___Toc3066_3299793794)

[7. Attendance 22](#__RefHeading___Toc3068_3299793794)

[8. Customer 22](#__RefHeading___Toc3070_3299793794)

[9. Sale 22](#__RefHeading___Toc3072_3299793794)

[10. Leave 22](#__RefHeading___Toc3074_3299793794)

[11. Expense 23](#__RefHeading___Toc3076_3299793794)

[14. Relationships Overview (Diagrammatically Inferred) 23](#__RefHeading___Toc3078_3299793794)

[15. Finalized Normalized Relations : 23](#__RefHeading___Toc3080_3299793794)

[1. Normalization Summary 26](#__RefHeading___Toc3082_3299793794)

[16. SQL Database Tables and Queries: 26](#__RefHeading___Toc3084_3299793794)

[1. Database Creation: 26](#__RefHeading___Toc3086_3299793794)

[2. Tables: 27](#__RefHeading___Toc3088_3299793794)

[17. Conclusion: 28](#__RefHeading___Toc3090_3299793794)

# **Project Report: ERP System**

## **Project Title:** ERP System

## **Group Members:**

* Fiza Nihar Roll No: 22pwcse2100
* Afia Shah Roll No: 22pwcse2115
* Hafsa Javed Roll No: 22pwcse2171

## **Objective:**

The objective of this ERP System project is to design and implement a comprehensive, real-time, and modular web-based ERP application capable of managing all core aspects of an enterprise’s internal operations. It integrates various modules like HR Management,

Inventory Control, Product & Supplier Management, Purchase & Sales Recording, Attendance, Leave Tracking, and a visually insightful Admin Dashboard. The ERP system ensures that manual processes are digitized, data is consistent across all modules, and the entire workflow of a business can be monitored from a single platform.

Key goals achieved:

* + Designed modular architecture with separation of concerns
  + Built secure, JWT-based login and authentication system
  + Implemented 10+ interlinked modules using REST APIs
  + Managed relational data with MySQL and enforced foreign key constraints
  + Deployed and tested the application live using Hostinger

## **Technologies Used:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Component** | | Frontend | | API Communication | | Styling | | Backend | | Database | | Authentication | | Dev Tools | | Hosting | | |  | | --- | | **Technology** | | React.js (JSX, Hooks, Functional Components) | | Axios (with Auth Headers) | | Tailwind CSS, Flex/Grid Layouts | | Laravel 10, PHP 8+ | | MySQL 8+ with indexed foreign keys | | JWT Tokens, Laravel Middleware | | Postman, Composer, VS Code, Git | | Hostinger Shared Hosting, cPanel | |
| |  | | --- | | Deployment Tools | | Documentation | | |  | | --- | | File Manager, phpMyAdmin, GitHub | | README.md, Manual.txt, Screenshots | |

## **System Architecture & Component Structure:**

### **Frontend Structure:**

* + **Authentication:** LoginPage.js stores token in localStorage
  + **Navigation:** Navbar.js, Sidebar.js
  + **Dashboard:** Dashboard.js displays summaries from dashboard API  **Modules:**
    - EmployeeList.js, DepartmentList.js o ProductList.js, PurchaseList.js, SalesList.js o AttendanceList.js, LeaveList.js
  + **Forms:** Dynamic controlled forms for input validation
  + **Reusable Components:** Table, Dropdown, Button, Modal

### **Backend Structure (Laravel):**

* + **Routes:** api.php defines REST endpoints with auth:api middleware
  + **Controllers:** CRUD logic for each module (e.g., EmployeeController.php)
  + **Models:** Mapped to MySQL tables (e.g., Product, Attendance)
  + **Migrations:** 2024\_create\_products\_table.php for schema
  + **Middleware:** Verifies tokens, protects routes
  + **Resources:** Formats JSON for client consumption **Database Schema:**
  + Normalized structure with referential integrity  Sample relations:
    - employees.department\_id → departments.id o products.category\_id → categories.id o purchases.product\_id, purchases.supplier\_id

## **Module Descriptions with Work Done:**

### **Employee Module:**

* + React form for name, email, department
  + Displays list in a styled table with edit/delete
  + Data fetched via /api/employees **Department Module:**
  + Manage departments (CRUD)
  + Filters employees by department

### **Category & Product Modules:**

* + Add categories like Electronics, Stationery, etc.
  + Each product has name, category, price, quantity
  + Low stock alerts considered for future

### **Supplier Module:**

* + Stores vendor info
  + Used in Purchase module
  + Linked via foreign key **Customer Module:**
  + Registers customer details
  + Used when recording sales **Purchases Module:**
  + Records inventory inflow
  + Quantity is added to product stock
  + Total cost auto-calculated in React form

### **Sales Module:**

* + Records inventory outflow
  + Reduces product quantity
  + Warns on insufficient stock

### **Attendance Module:**

* + Employee attendance recorded with status
  + Dropdown selection per employee

### **Leave Module:**

* + Employees can request leave (future: approval system)
  + Stored with reason, duration **Dashboard Module:**
  + Dynamic API-powered cards:
    - Total employees, stock levels, purchases, sales o Attendance today, pending leaves

## **API, Security & Token Handling:**

* + All API endpoints secured with middleware
  + JWT token added in Axios headers
  + Invalid/expired tokens redirect to login
  + Laravel guards configured in auth.php

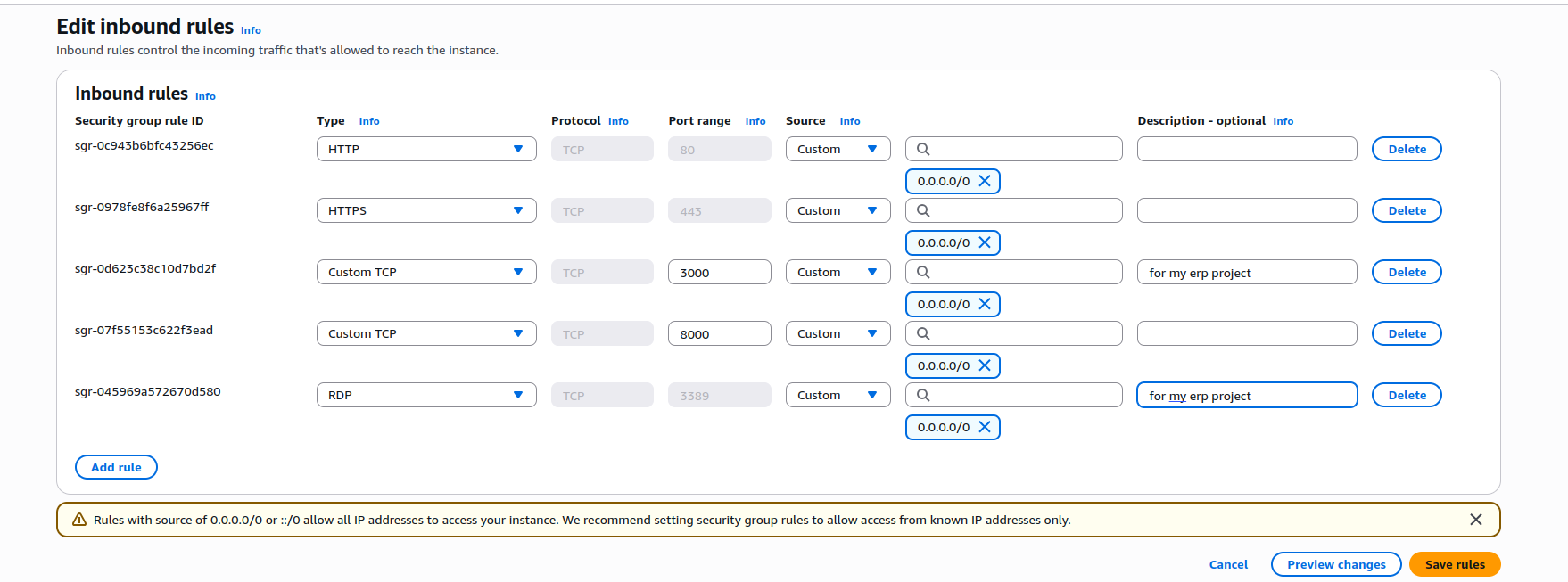
## **Testing & Validation Strategy:**

* + **Postman:** Verified all endpoints (GET, POST, PUT, DELETE)
  + **Frontend Validation:** Required fields, email format  **Integration Checks:** Inventory changes on purchase/sale  **Edge Case Testing:**
    - Sales > stock → error o Missing fields → alert o Unique email per employee

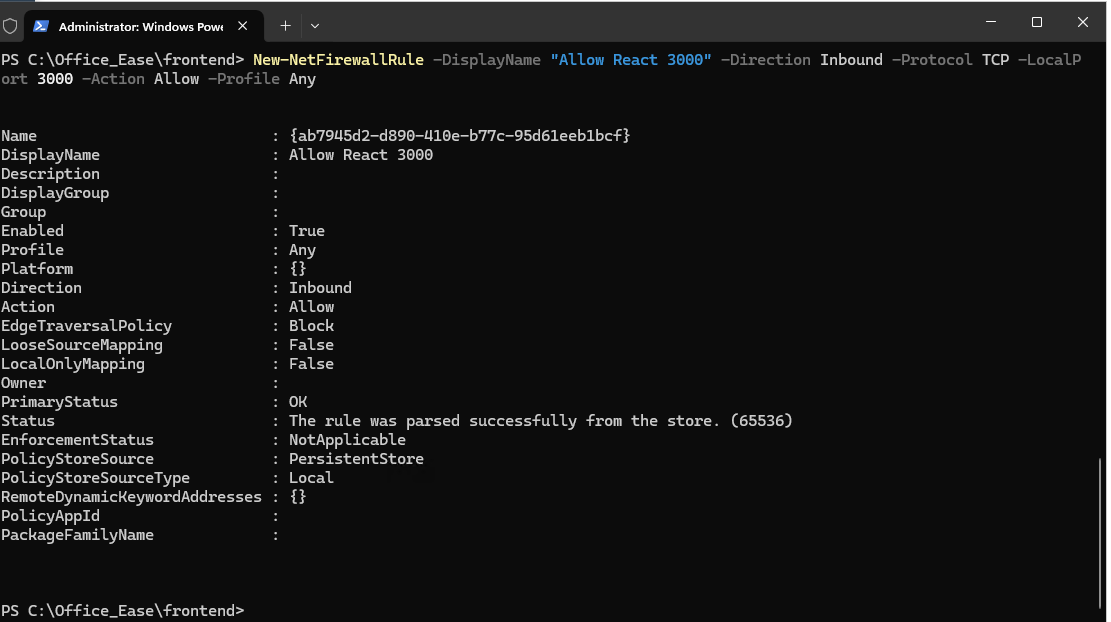
## **Documentation & Source Control:**

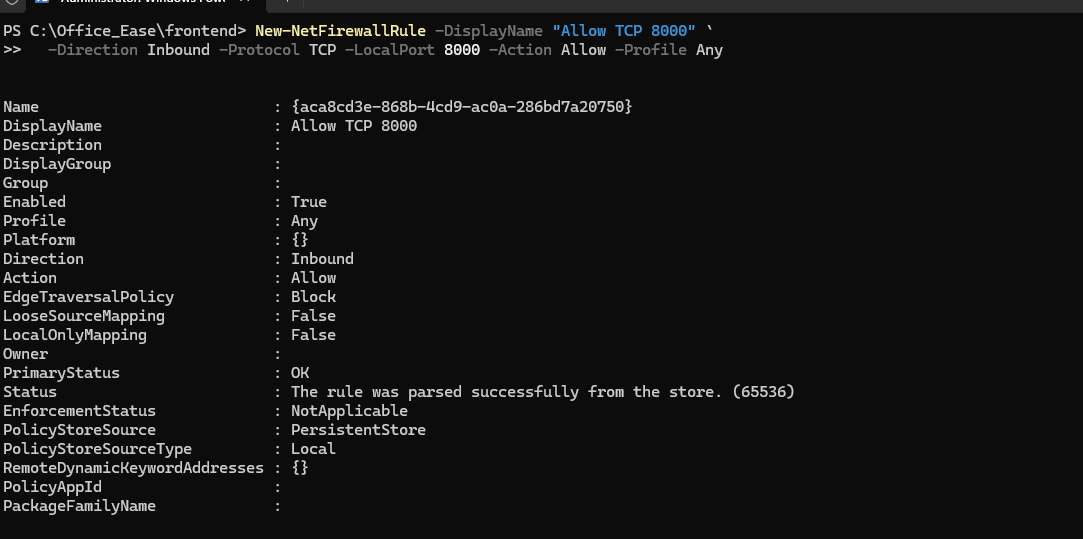
* + GitHub repository maintained with commits  README includes:
    - Setup instructions (local & Hostinger)
    - API reference o Screenshots
  + Project zip includes:
    - frontend/ o backend/ o database.sql
    - README.md

## Deployment on EC2 on AWS Cloud

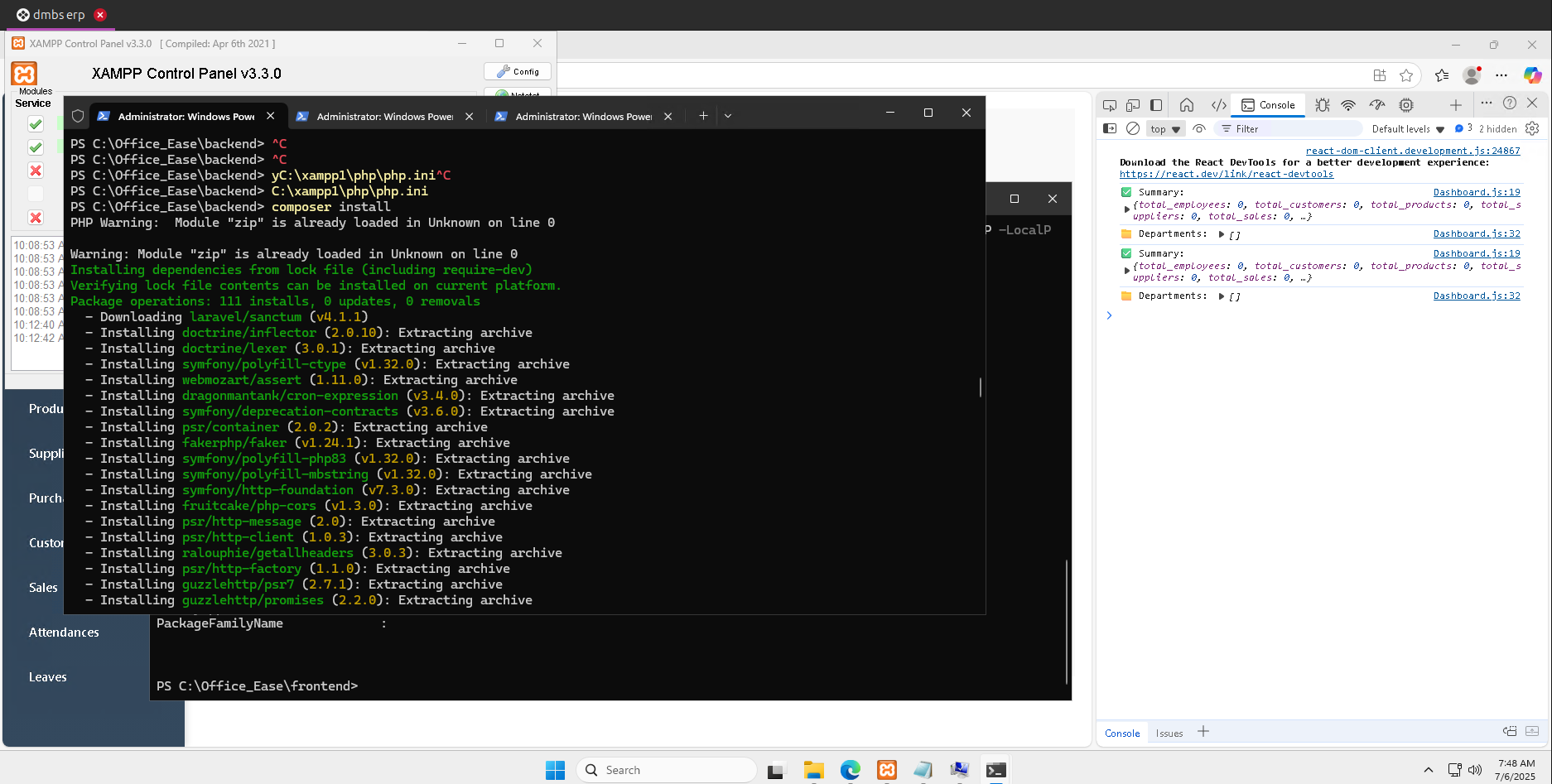
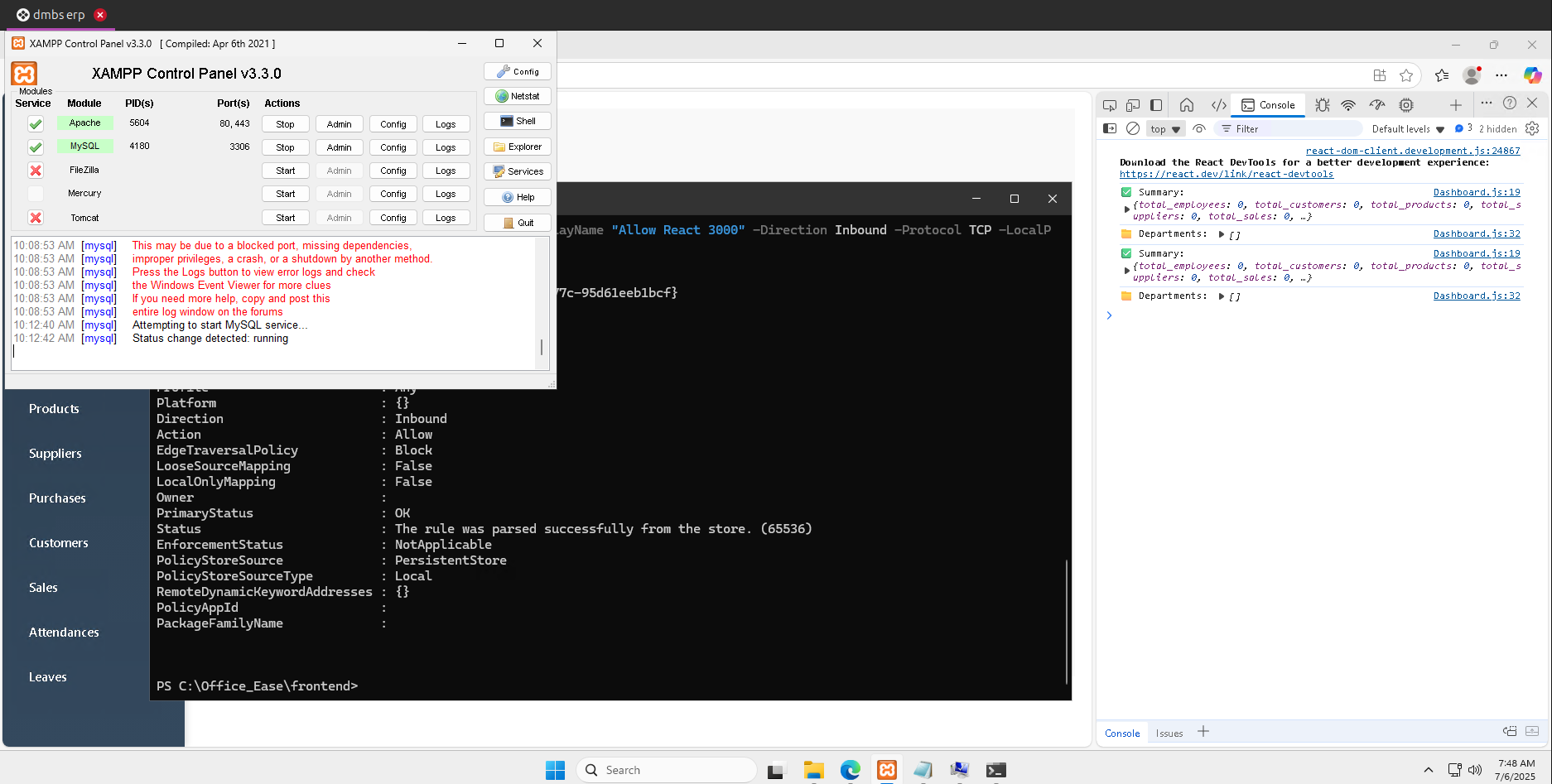


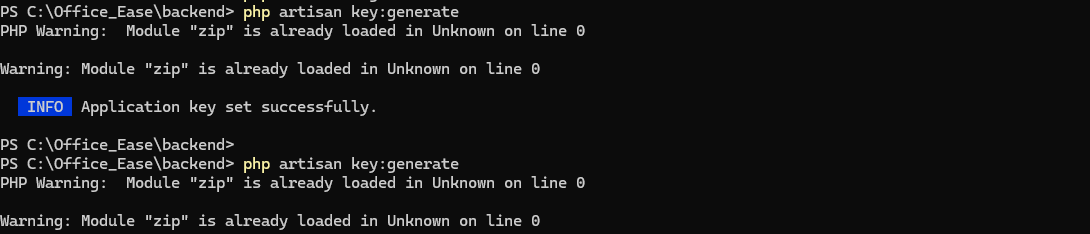
Added firewall rules to allow inbound tcp traffic from port 3000 and 8000

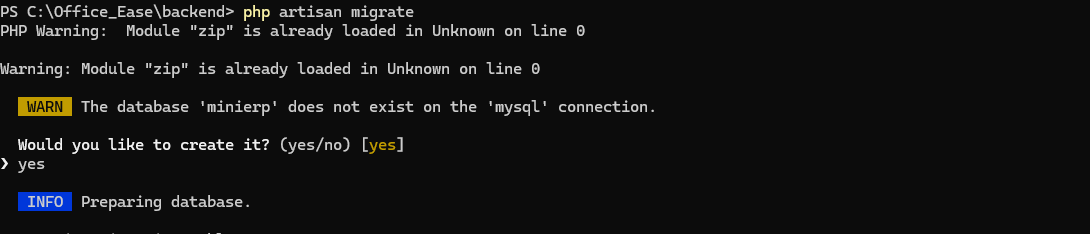


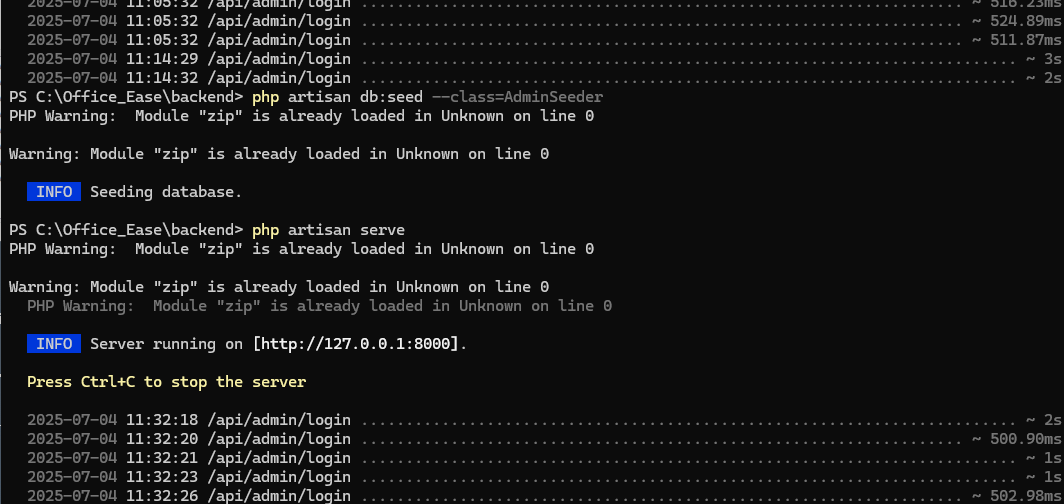


started apache and mysql service using xaamp

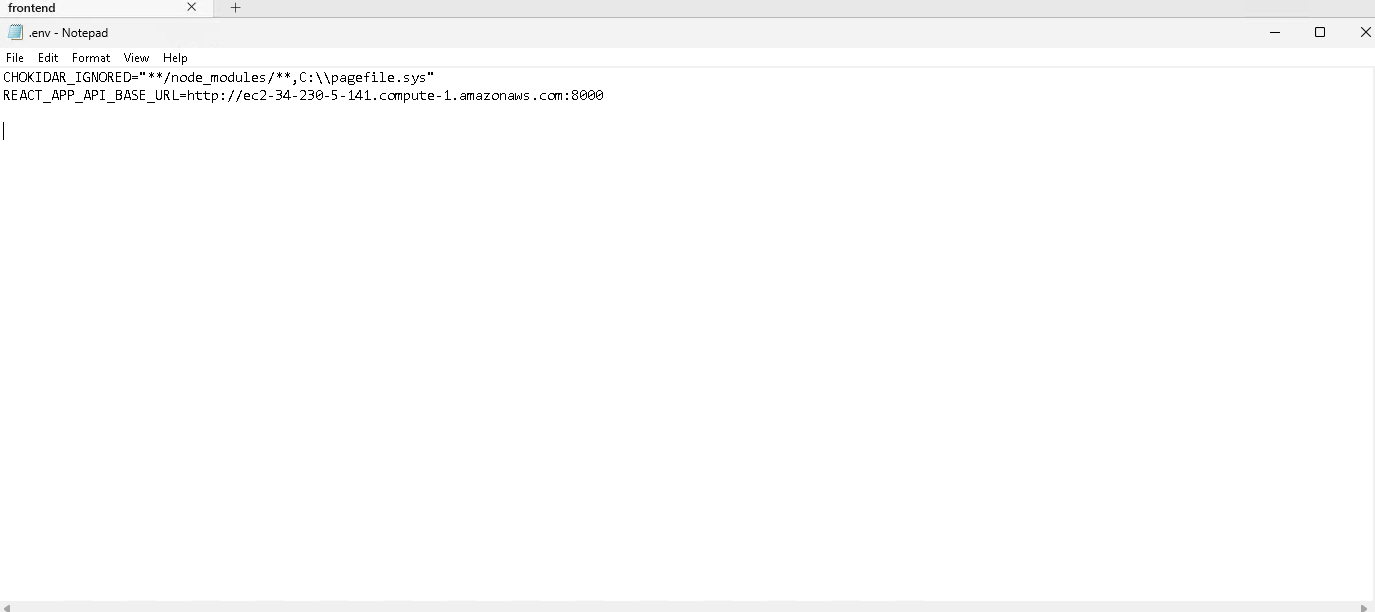






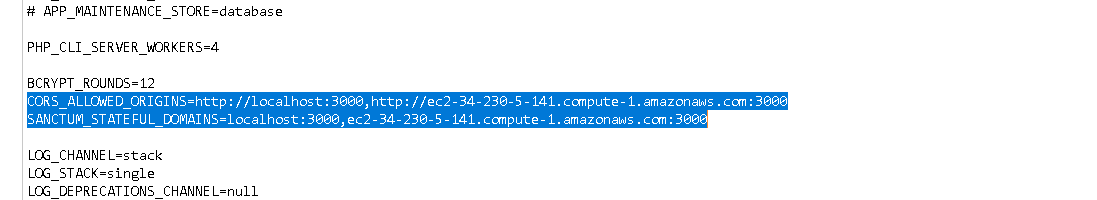


Added in fronted/.env to include the public DNS from ec2 instance

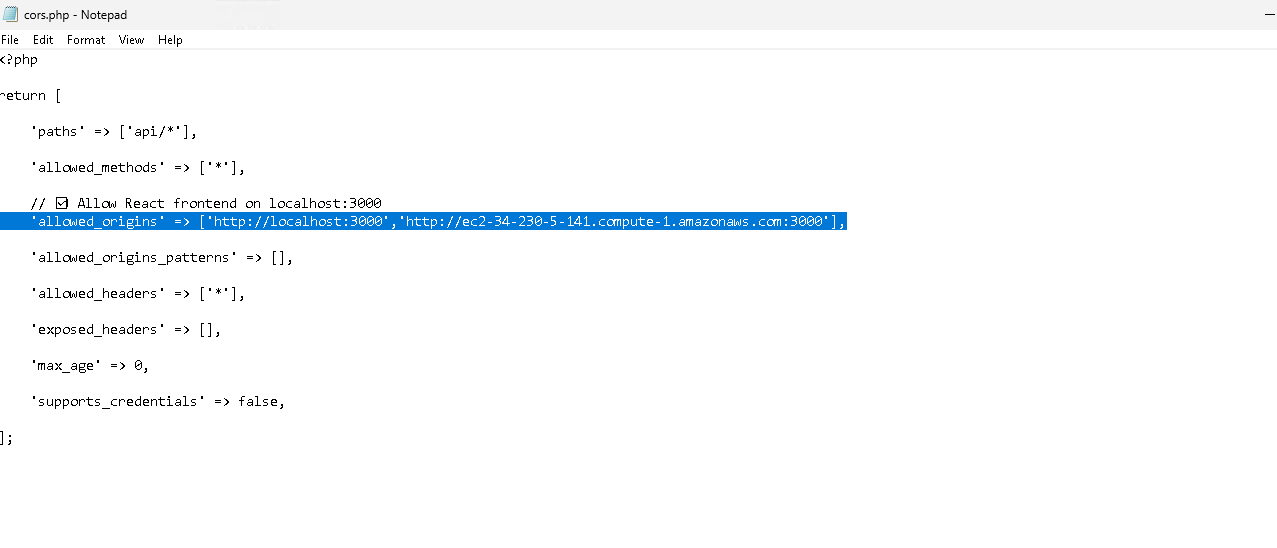


added support from response from public dns in frontend/src/components/AdminLogin.js and all other components as well

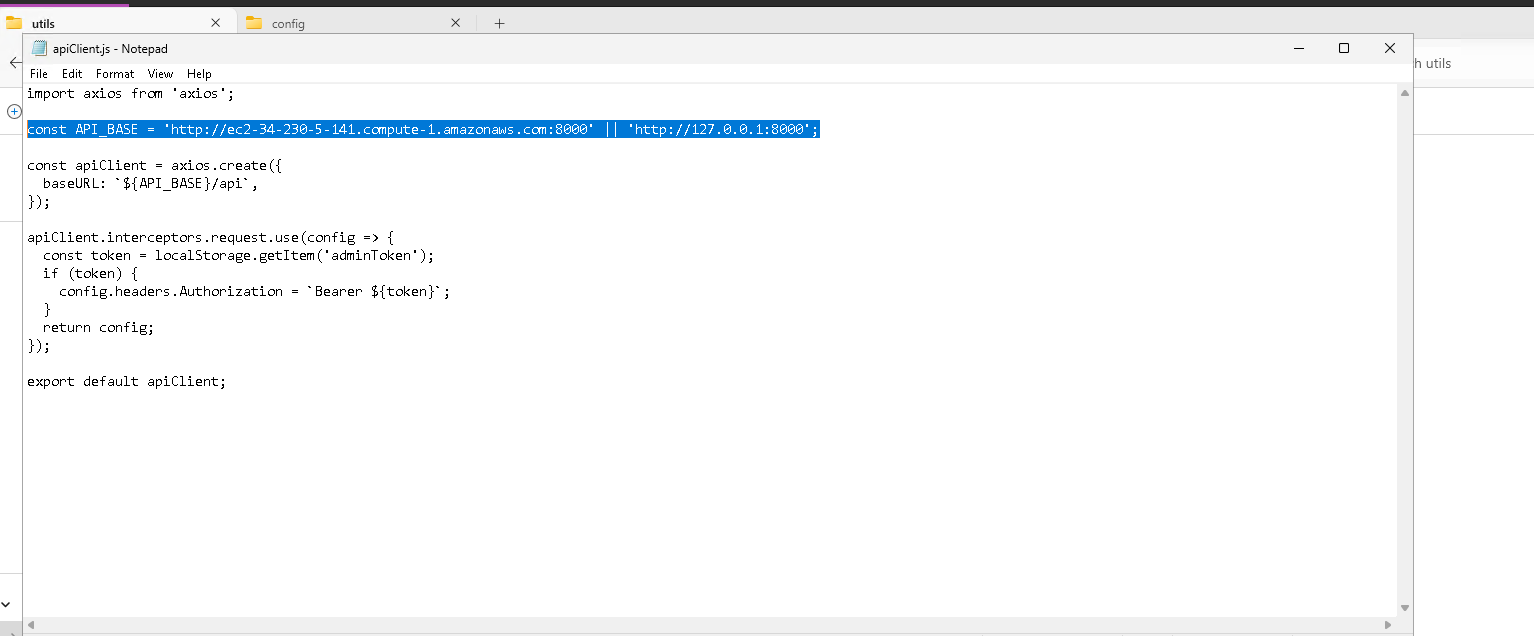
added in backend/.nev



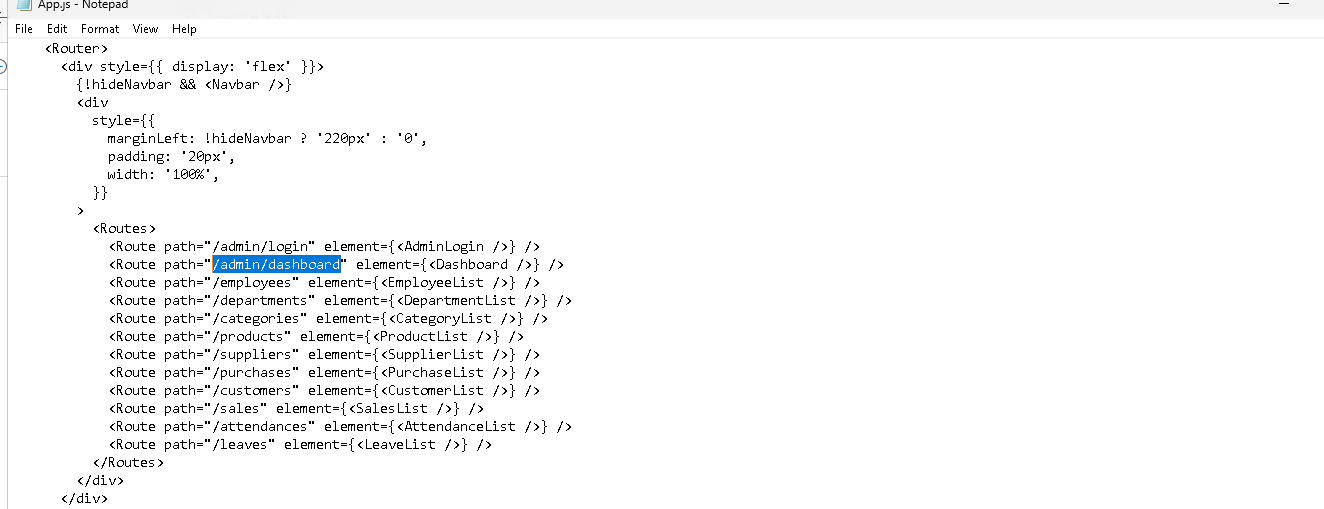
public dns added in backend/config/cors.php



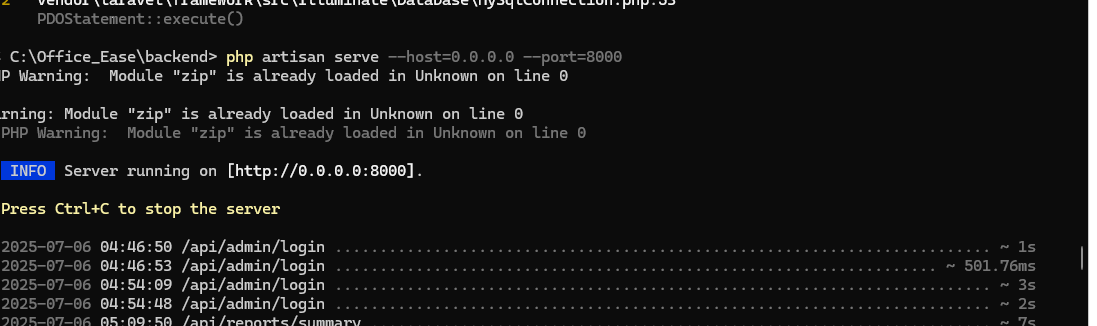
in frontend/src/utils/apiClient.js



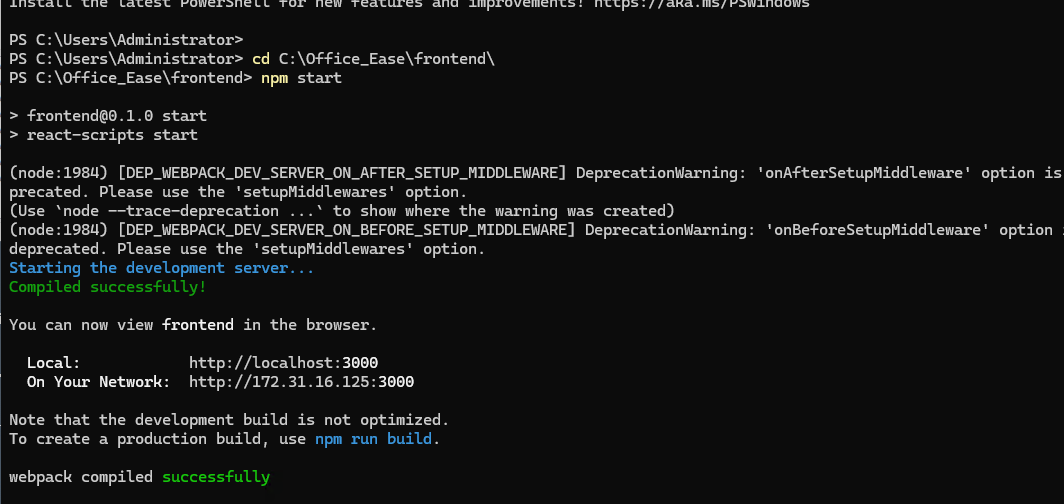
in frontend/src/App.js



starting php server to listen on all Ips from port 8000



started npm server

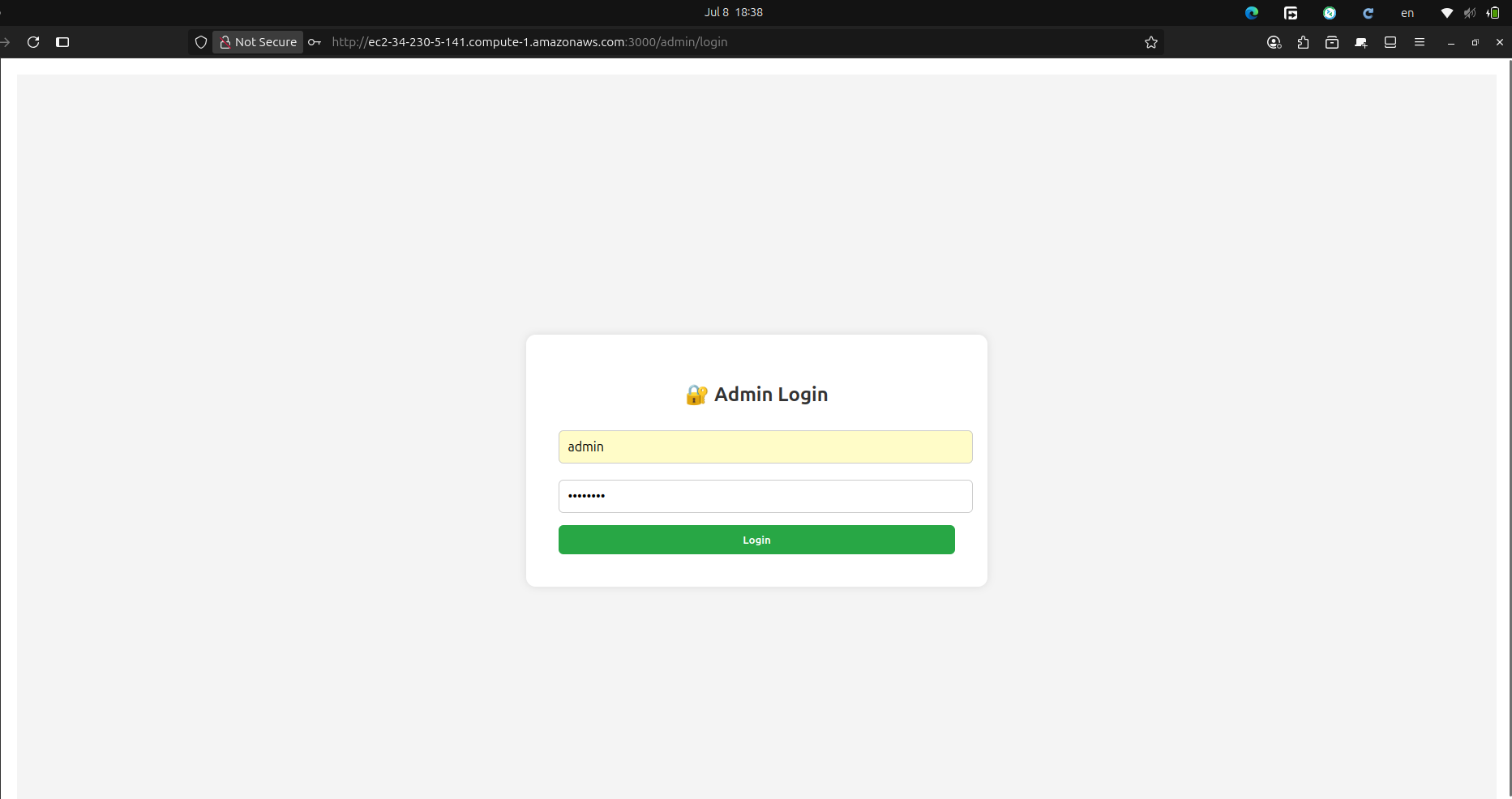


final view

link: http://ec2-34-230-5-141.compute-1.amazonaws.com:3000/

username: admin

password:admin123

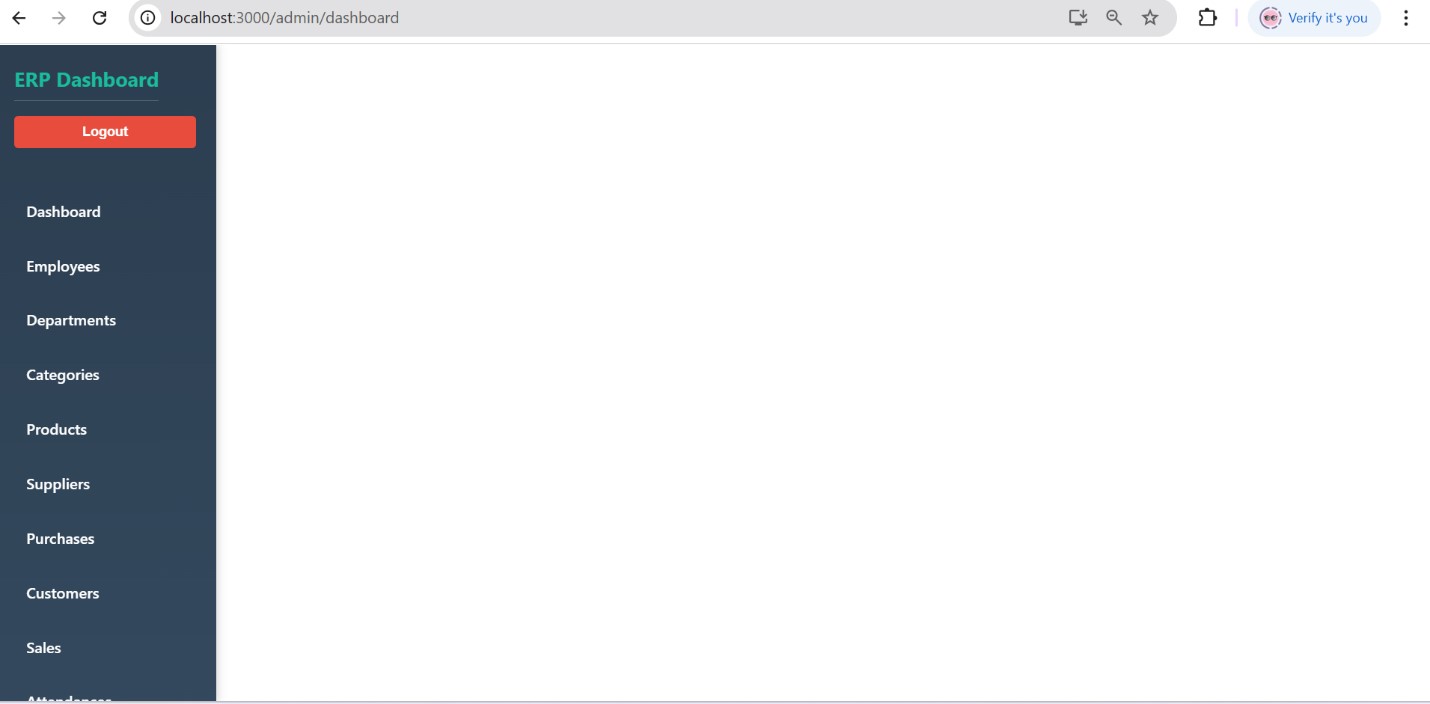
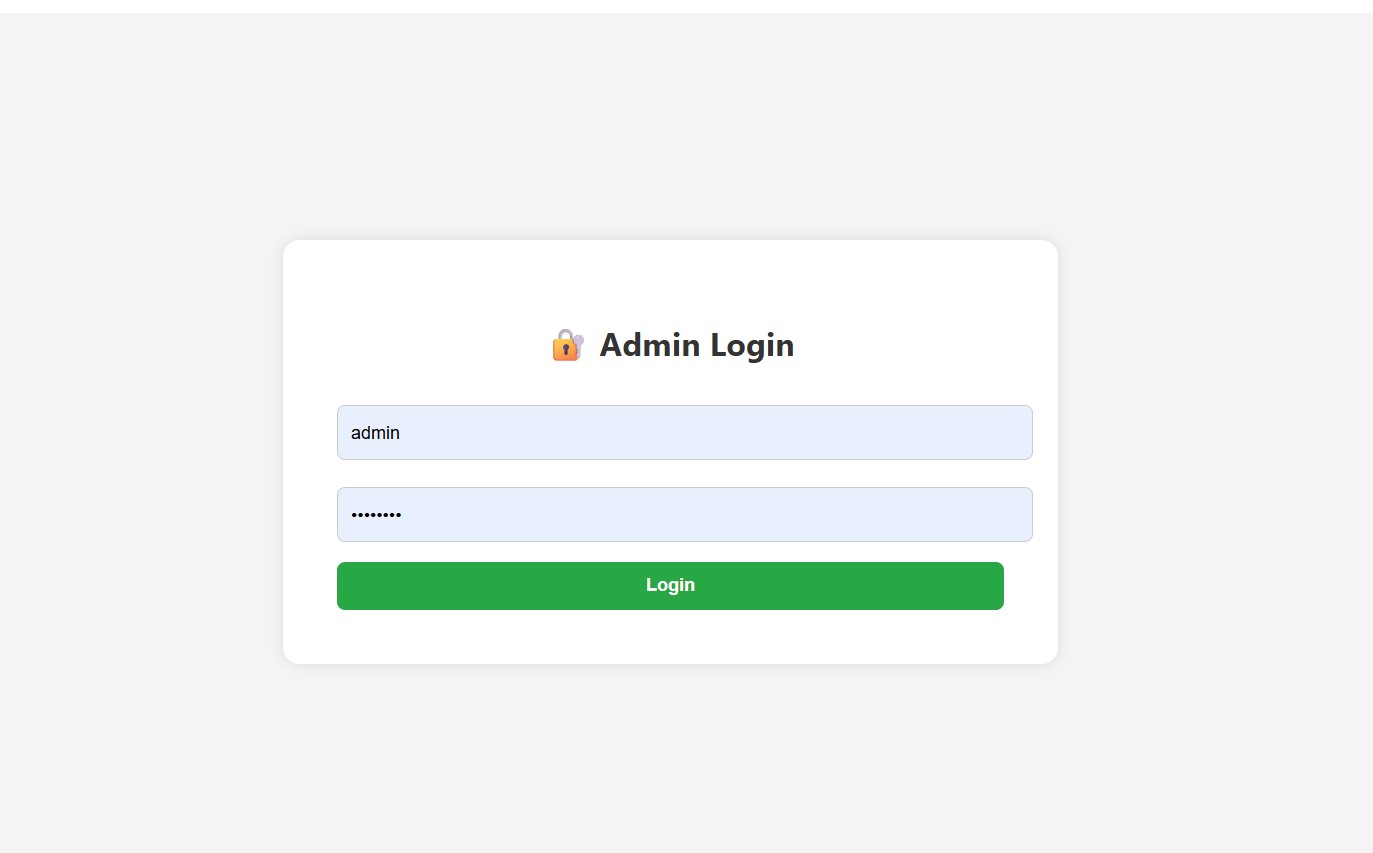


## **Submission Summary:**

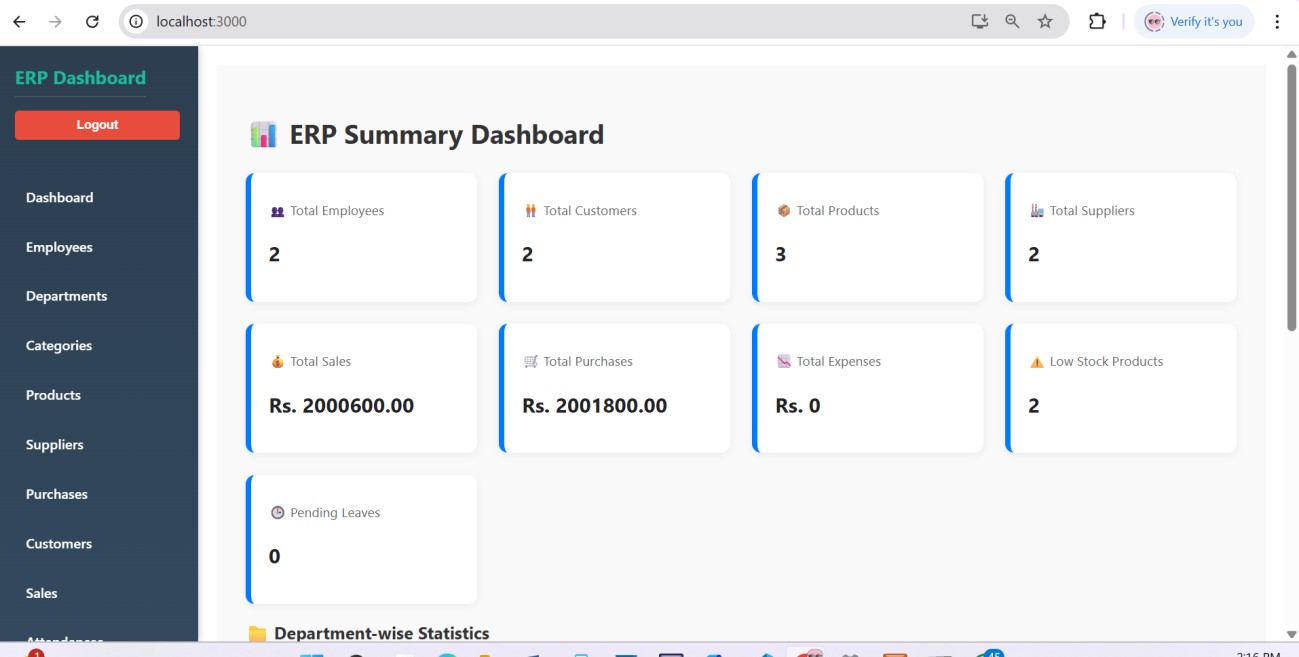
* + **GitHub Repo:** https://github.com/niharfiza125/Office\_Ease  **Included Files:**
    - Frontend code o Backend code o SQL database o Screenshots
    - README
  + **ZIP File Name:** ERP\_1980\_1983\_2059.zip

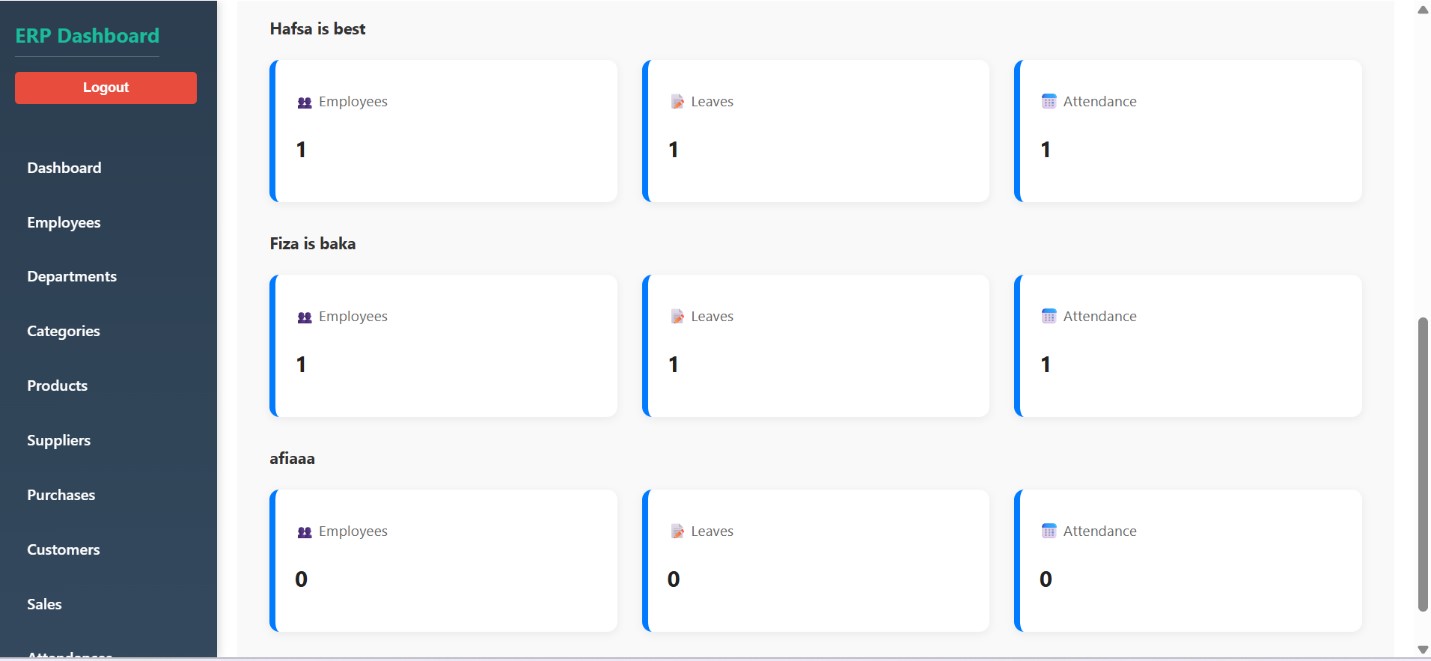
## Screen Shots:

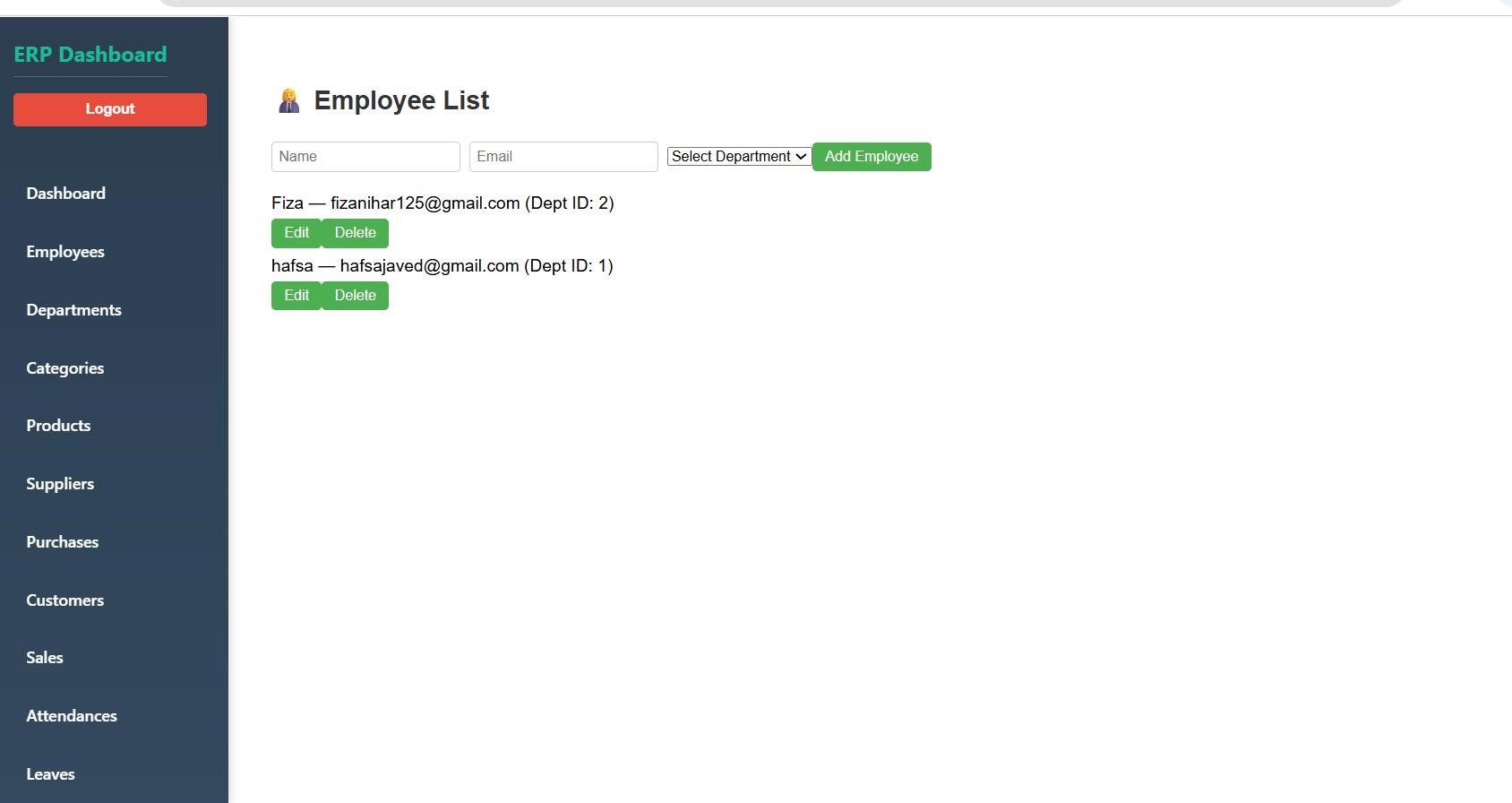
### Admin Login:

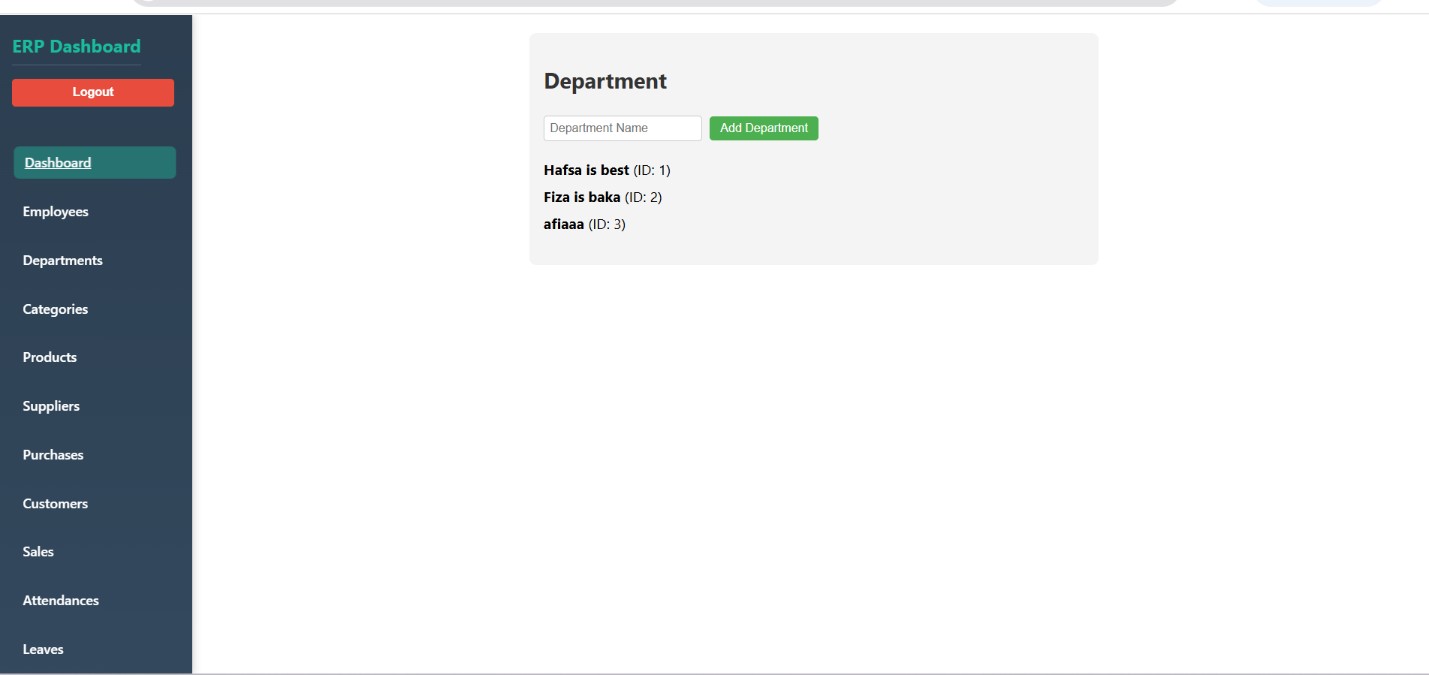


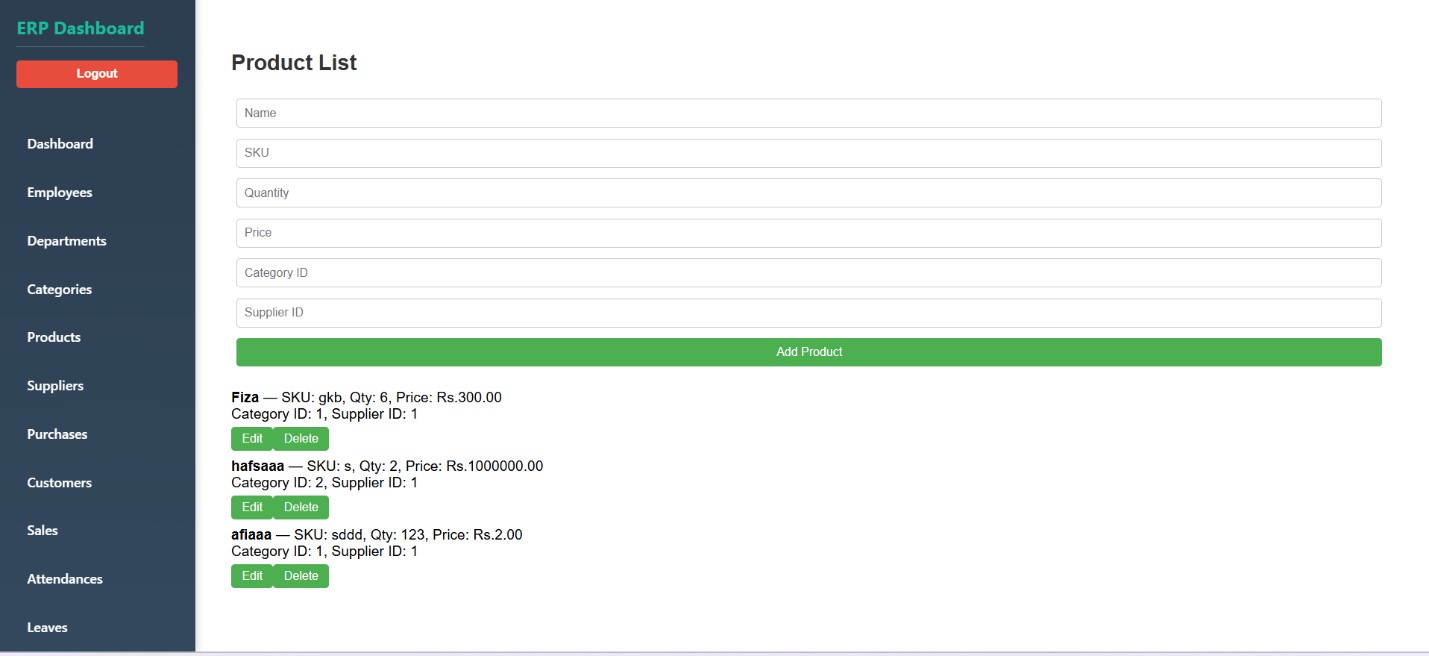
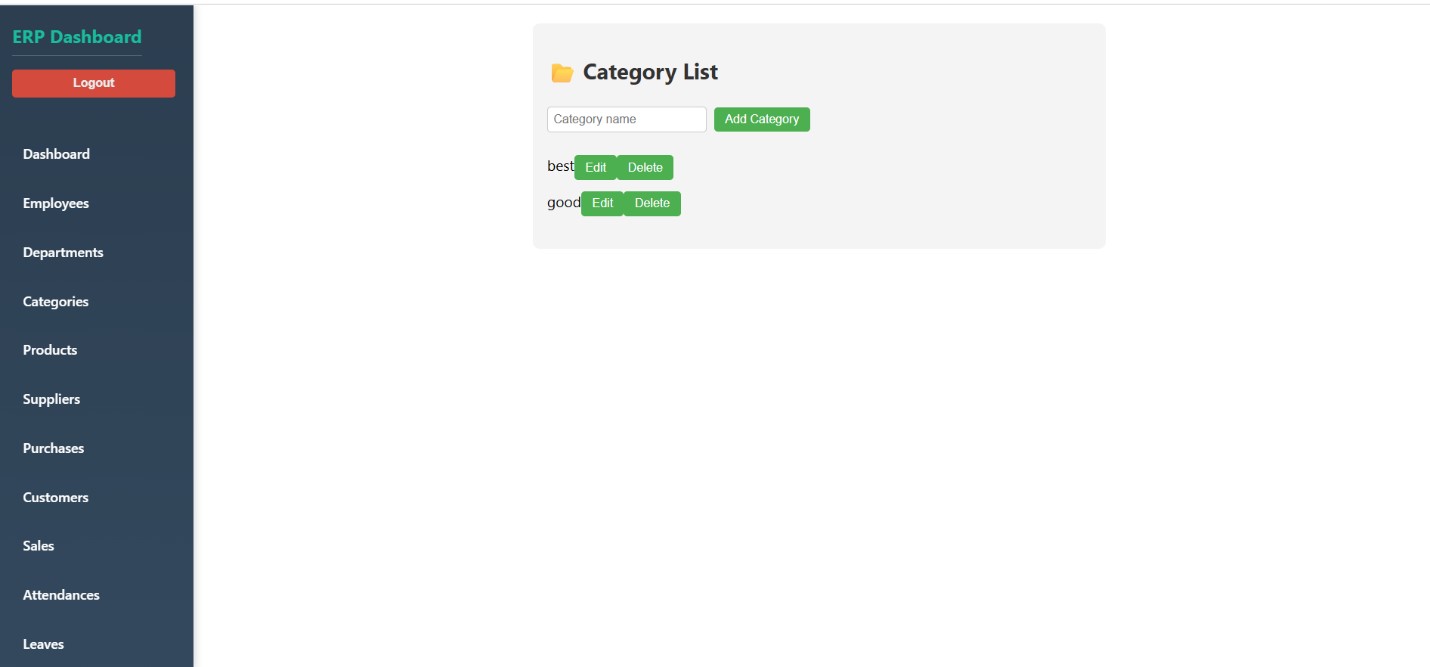
### **Dashboard:**

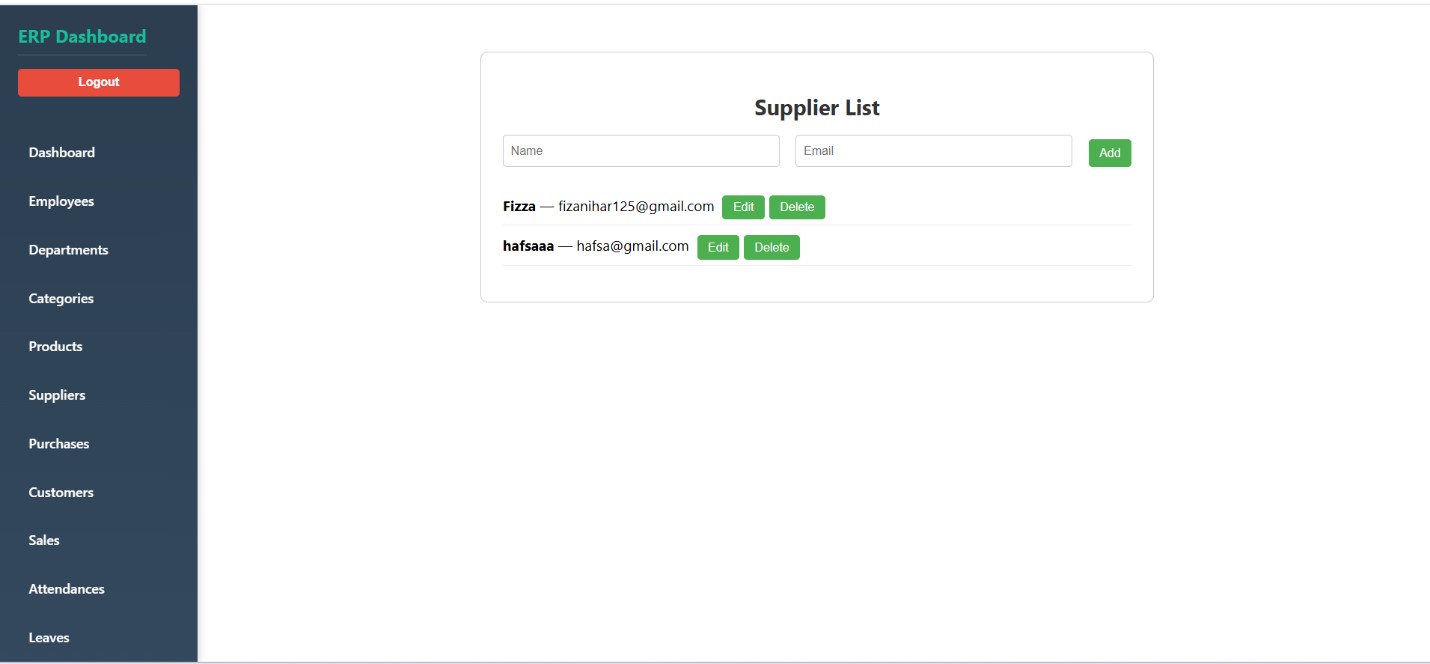


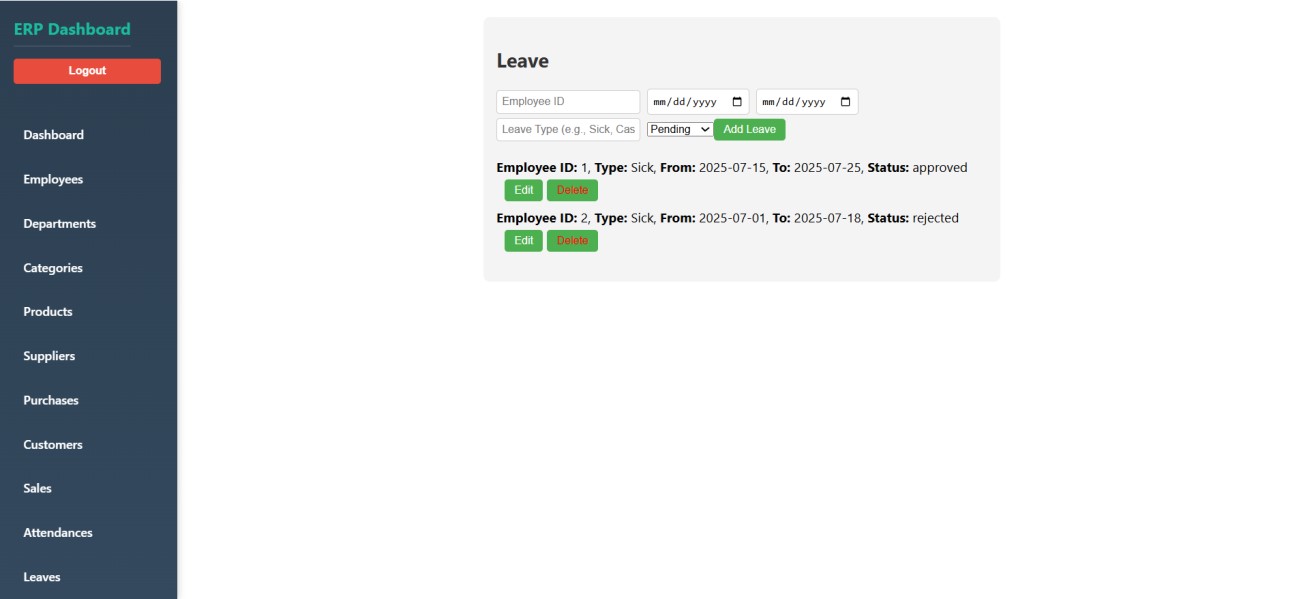
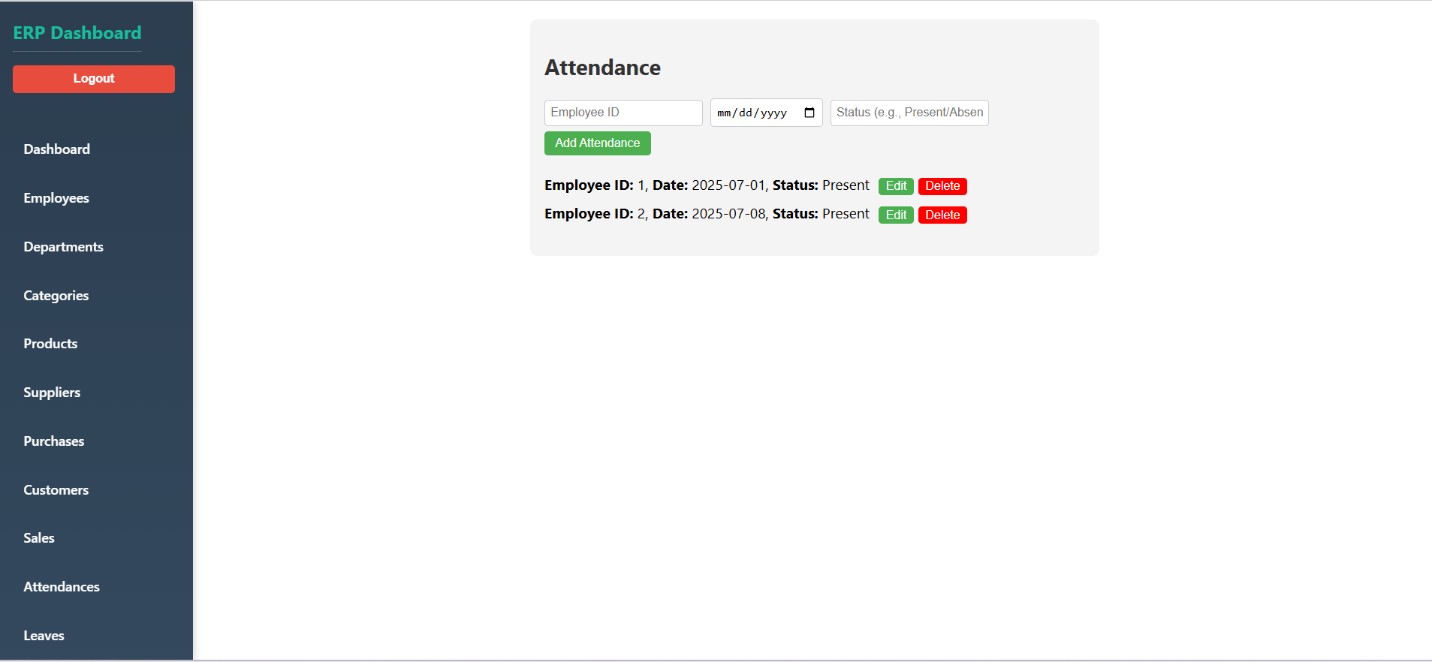
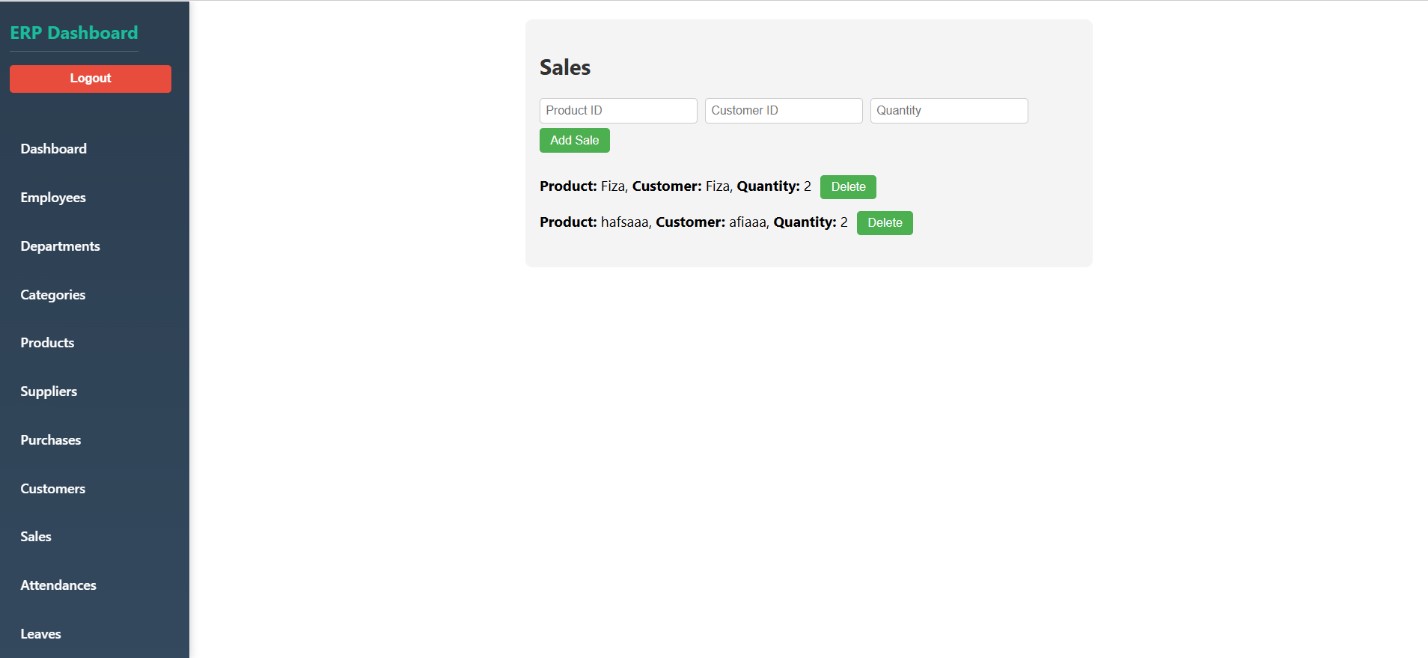
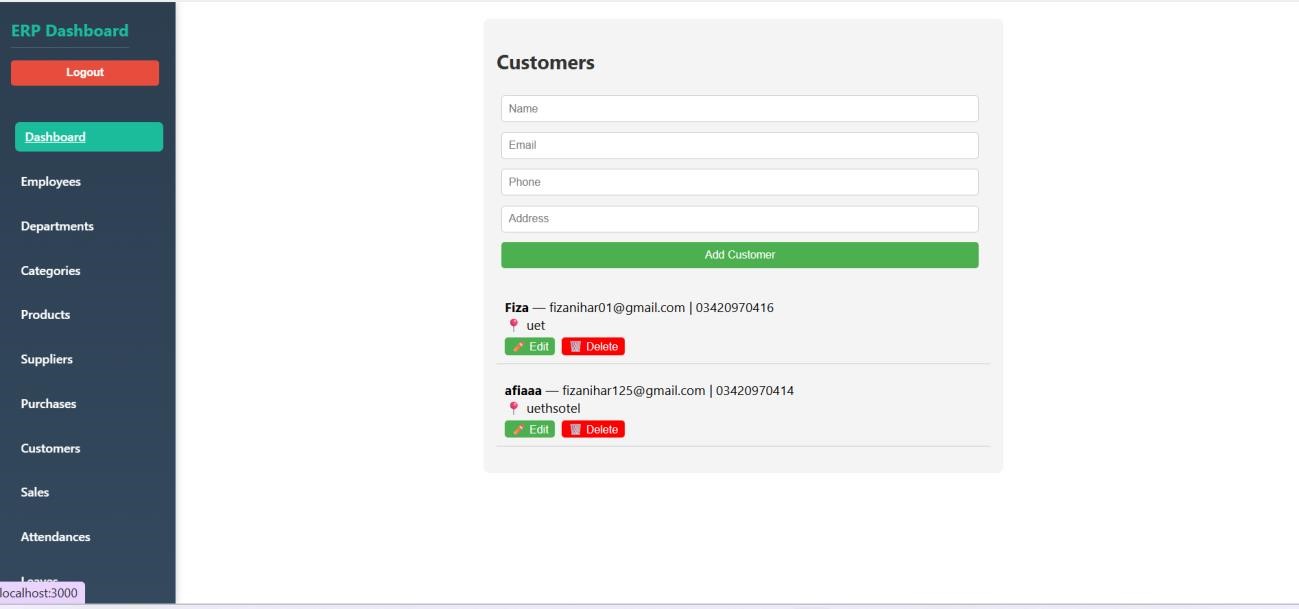
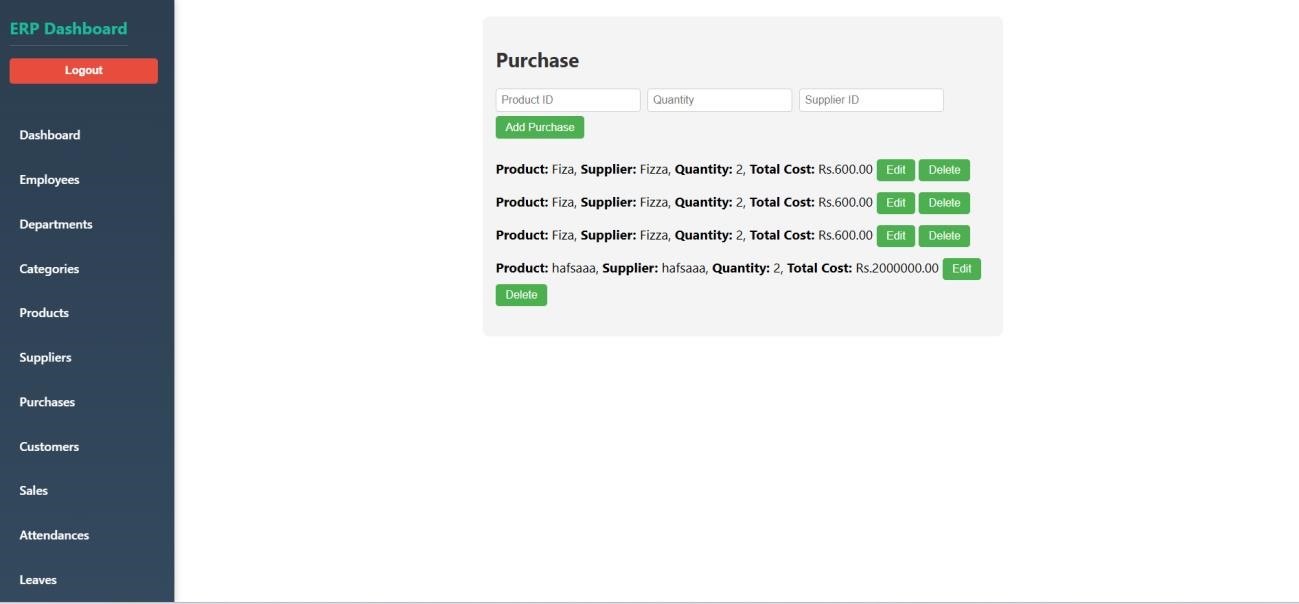












## **Conceptual schema :**

### Finalized Conceptual Schema (Entity-Level Description):

#### Product

Represents items available for sale or purchase in the system.

* + **product\_id** *(Primary Key)*: Unique identifier for each product.
  + **name**: Name of the product.
  + **category\_id** *(Foreign Key → Categories.category\_id)*: Links each product to its category.
  + **price**: Cost per unit of the product.
  + **quantity**: Current inventory stock level.

#### Category

Represents classifications for products.

* + **category\_id** *(Primary Key)*: Unique identifier for a category.  **name**: Unique category name (e.g., Electronics, Stationery).

#### Supplier

Vendors providing products.

* + **supplier\_id** *(Primary Key)*: Unique identifier for each supplier.
  + **name**: Supplier name.
  + **contact**: Phone or email contact.
  + **address**: Physical address of the supplier.

#### Purchase

Records procurement of products.

* + **purchase\_id** *(Primary Key)*: Unique identifier for a purchase transaction.  **product\_id** *(Foreign Key → Products.product\_id)*: Product being purchased.  **supplier\_id** *(Foreign Key → Suppliers.supplier\_id)*: Supplier from whom the product was purchased.
  + **quantity**: Number of units purchased.
  + **date**: Date of the purchase transaction.

#### Employee

Staff members of the organization.

* + **employee\_id** *(Primary Key)*: Unique identifier for each employee.
  + **name**: Full name.
  + **department\_id** *(Foreign Key → Departments.department\_id)*: Which department they belong to.
  + **salary**: Monthly salary.
  + **contact**: Phone or email contact.

#### Department

Units within the company.

* + **department\_id** *(Primary Key)*: Unique identifier for a department.  **name**: Department name (e.g., Sales, IT).

#### Attendance

Tracks employee presence.

* + **attendance\_id** *(Primary Key)*: Unique identifier for an attendance record.
  + **employee\_id** *(Foreign Key → Employees.employee\_id)*: Employee present or absent.
  + **date**: Date of the record.
  + **status** *(ENUM: Present, Absent)*: Attendance status.

#### Customer

Clients who purchase products.

* + **customer\_id** *(Primary Key)*: Unique ID for each customer.
  + **name**: Full name of the customer.
  + **email**: Email address.
  + **phone**: Contact number.
  + **address**: Physical location.

#### Sale

Tracks customer purchases.

* + **sale\_id** *(Primary Key)*: Unique sale transaction ID.
  + **customer\_id** *(Foreign Key → Customers.customer\_id)*: The buyer.
  + **product\_id** *(Foreign Key → Products.product\_id)*: Product sold.  **quantity**: Number of units sold.
  + **total\_price**: Computed total amount (price × quantity).  **date**: Date of sale.

#### Leave

Requests for time off by employees.

* + **leave\_id** *(Primary Key)*: Unique identifier.
  + **employee\_id** *(Foreign Key → Employees.employee\_id)*: Applicant employee.  **type**: Type of leave (e.g., Sick, Casual).
  + **start\_date**: Starting date of leave.
  + **end\_date**: Ending date of leave.
  + **status** *(ENUM: pending, approved, rejected)*: Approval state.

#### Expense

Any organizational spending.

* + **expense\_id** *(Primary Key)*: Unique ID.
  + **description**: Nature of the expense.
  + **amount**: Monetary value.  **date**: Date of expense.

## **Relationships Overview (Diagrammatically Inferred)**

* + One **Category** ➝ Many **Products**
  + One **Supplier** ➝ Many **Purchases**
  + One **Product** ➝ Many **Purchases**
  + One **Department** ➝ Many **Employees**
  + One **Employee** ➝ Many **Attendances**
  + One **Customer** ➝ Many **Sales**
  + One **Product** ➝ Many **Sales**
  + One **Employee** ➝ Many **Leaves**

## Finalized Normalized Relations :

Each relation (table) is in **1NF, 2NF, and 3NF**, with **no repeating groups**, **partial dependencies**, or **transitive dependencies**.

1. **departments**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key, Auto-Increment |
| name | VARCHAR | Unique, Not Null |

1. **employees**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key, Auto-Increment |
| name | VARCHAR | Not Null |
| department\_id | INT | Foreign Key → departments(id) |
| salary | DECIMAL | Not Null |
| contact | VARCHAR | Nullable |

Fully normalized: No repeating or dependent fields; department\_id relates to department table.

1. **categories**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| name | VARCHAR | Unique, Not Null |

1. **products**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| name | VARCHAR | Not Null |
| category\_id | INT | Foreign Key → categories(id) |
| price | DECIMAL | Not Null |
| quantity | INT | Default 0 |

Normalized: No category names here, just a foreign key to categories.

1. **suppliers**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| name | VARCHAR | Not Null |
| contact | VARCHAR | Nullable |
| address | VARCHAR | Nullable |

1. **purchases**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| product\_id | INT | Foreign Key → products(id) |
| supplier\_id | INT | Foreign Key → suppliers(id) |
| quantity | INT | Not Null |
| total\_cost | DECIMAL | Not Null (auto-calculated) |
| date | DATE | Not Null |

1. **customers**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| name | VARCHAR | Not Null |
| email | VARCHAR | Unique, Nullable |
| **Attribute** | **Type** | **Constraints** |
| phone | VARCHAR | Nullable |
| address | VARCHAR | Nullable |

1. **sales**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| customer\_id | INT | Foreign Key → customers(id) |
| product\_id | INT | Foreign Key → products(id) |
| quantity | INT | Not Null |
| total\_price | DECIMAL | Not Null (price × quantity) |
| date | DATE | Not Null |

No redundancy; price is not stored from products directly.

1. **attendances**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| employee\_id | INT | Foreign Key → employees(id) |
| date | DATE | Not Null |
| status | ENUM | Values: 'Present', 'Absent' |

1. **leaves**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| employee\_id | INT | Foreign Key → employees(id) |
| type | VARCHAR | e.g., Sick, Casual |
| start\_date | DATE | Not Null |
| end\_date | DATE | Not Null, >= start\_date |
| status | ENUM | Values: 'pending', 'approved', 'rejected' |

1. **expenses**

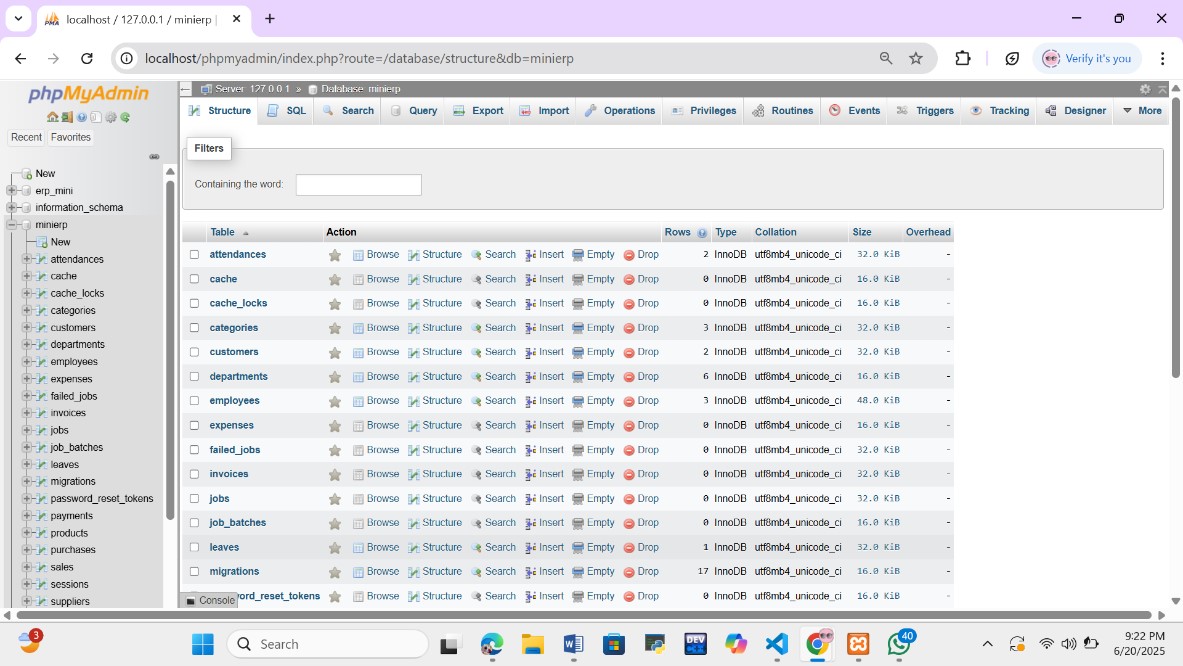
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Constraints** |
| id | INT | Primary Key |
| description | TEXT | Not Null |
| amount | DECIMAL | Not Null |
| date | DATE | Not Null |

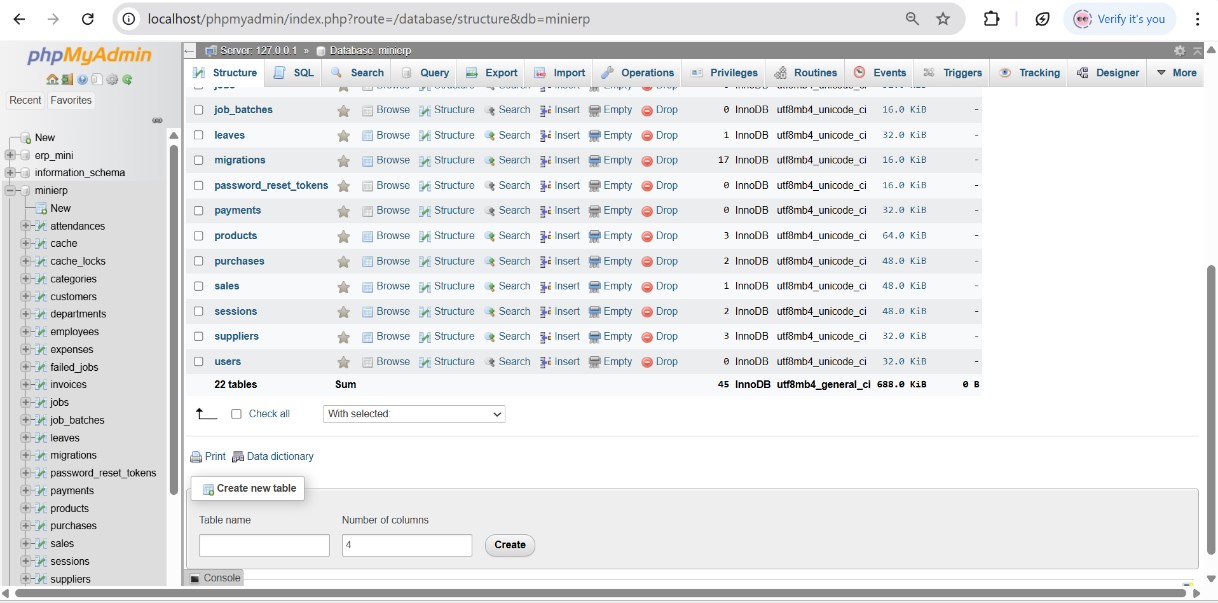
### Normalization Summary

* + All tables are in **1NF** (no repeating or multi-valued attributes).
  + **2NF** is satisfied (no partial dependencies).
  + **3NF** is ensured (no transitive dependencies).

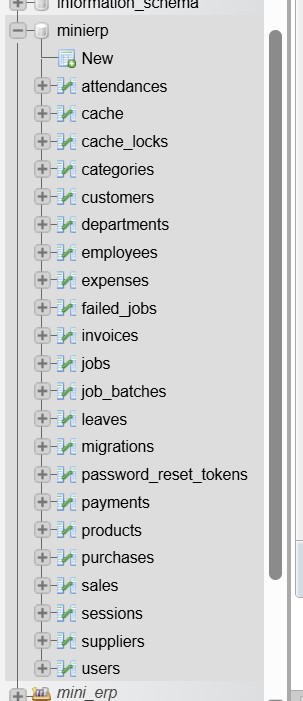
## SQL Database Tables and Queries:

### **Database Creation:**





### **Tables:**



## **Conclusion:**

This ERP system showcases a complete business management application using React, Laravel, and MySQL. It brings together multiple modules into one centralized system. With real deployment on Hostinger, real-time testing, security practices, and modular code, it proves readiness for real-world use. This project enhanced our understanding of full-stack systems, production deployment, and enterprise app structure.