

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

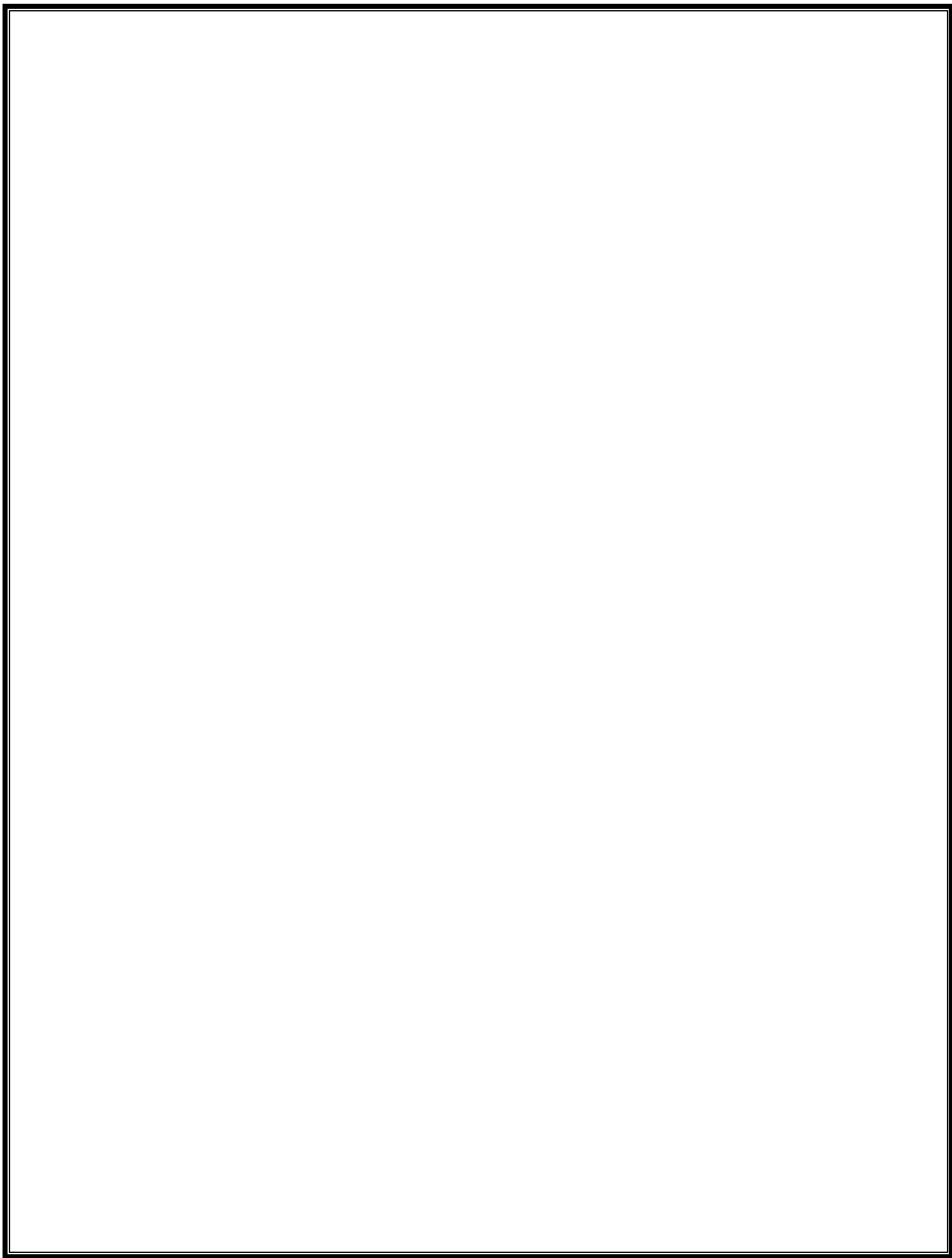
Month Day, Year (01 06, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Table of Contents

1. Project Report: ERP System.....	4
1. Project Title: ERP System.....	4
2. Group Members:.....	4
3. Objective:.....	4
4. Technologies Used:.....	4
5. System Architecture & Component Structure:.....	5
1. Frontend Structure:.....	5
2. Backend Structure (Laravel):.....	5
6. Module Descriptions with Work Done:.....	5
1. Employee Module:.....	5
2. Category & Product Modules:.....	5
3. Supplier Module:.....	6
4. Sales Module:.....	6
5. Attendance Module:.....	6
6. Leave Module:.....	6
7. API, Security & Token Handling:.....	6
8. Testing & Validation Strategy:.....	6
9. Documentation & Source Control:.....	7
10. Deployment on EC2 on AWS Cloud.....	7
11. Submission Summary:.....	14
12. Screen Shots:.....	14
1. Admin Login:.....	14
2. Dashboard:.....	15
13. Conceptual schema :.....	20
1. Finalized Conceptual Schema (Entity-Level Description):.....	20
1. Product.....	20
2. Category.....	21
3. Supplier.....	21
4. Purchase.....	21
5. Employee.....	21
6. Department.....	22
7. Attendance.....	22
8. Customer.....	22
9. Sale.....	22
10. Leave.....	22
11. Expense.....	23
14. Relationships Overview (Diagrammatically Inferred).....	23
15. Finalized Normalized Relations :.....	23
1. Normalization Summary.....	26
16. SQL Database Tables and Queries:.....	26
1. Database Creation:.....	26
2. Tables:.....	27
17. Conclusion:.....	28



1. Project Report: ERP System

1. Project Title: ERP System

2. Group Members:

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

3. 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
-

4. Technologies Used:

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

5. System Architecture & Component Structure:

1. 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
 - ProductList.js, PurchaseList.js, SalesList.js
 - AttendanceList.js, LeaveList.js
- **Forms:** Dynamic controlled forms for input validation
- **Reusable Components:** Table, Dropdown, Button, Modal

2. 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
 - products.category_id → categories.id
 - purchases.product_id, purchases.supplier_id

6. Module Descriptions with Work Done:

1. 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

2. Category & Product Modules:

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

3. **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

4. **Sales Module:**

- Records inventory outflow
- Reduces product quantity
- Warns on insufficient stock

5. **Attendance Module:**

- Employee attendance recorded with status
- Dropdown selection per employee

6. **Leave Module:**

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

7. **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`

8. **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:**
 - o Sales > stock → error o Missing fields
 - alert o Unique email per employee

9. Documentation & Source Control:

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

10. Deployment on EC2 on AWS Cloud

Edit inbound rules [Info](#)
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
Security group rule ID sgr-0c943b6bfc43256ec	HTTP	TCP	80	Custom	<input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>
sgr-0978fe0f6a25967ff	HTTPS	TCP	443	Custom	<input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>
sgr-0d623c38c10d7bd2f	Custom TCP	TCP	3000	Custom	<input type="text" value="0.0.0.0/0"/> for my erp project	<input type="button" value="Delete"/>
sgr-07f55153c622f5ead	Custom TCP	TCP	8000	Custom	<input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>
sgr-045969a572670d580	RDP	TCP	3389	Custom	<input type="text" value="0.0.0.0/0"/> for my erp project	<input type="button" value="Delete"/>

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

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

```
PS C:\Office_Ease\frontend> New-NetFirewallRule -DisplayName "Allow TCP 8000" `
>> -Direction Inbound -Protocol TCP -LocalPort 8000 -Action Allow -Profile Any

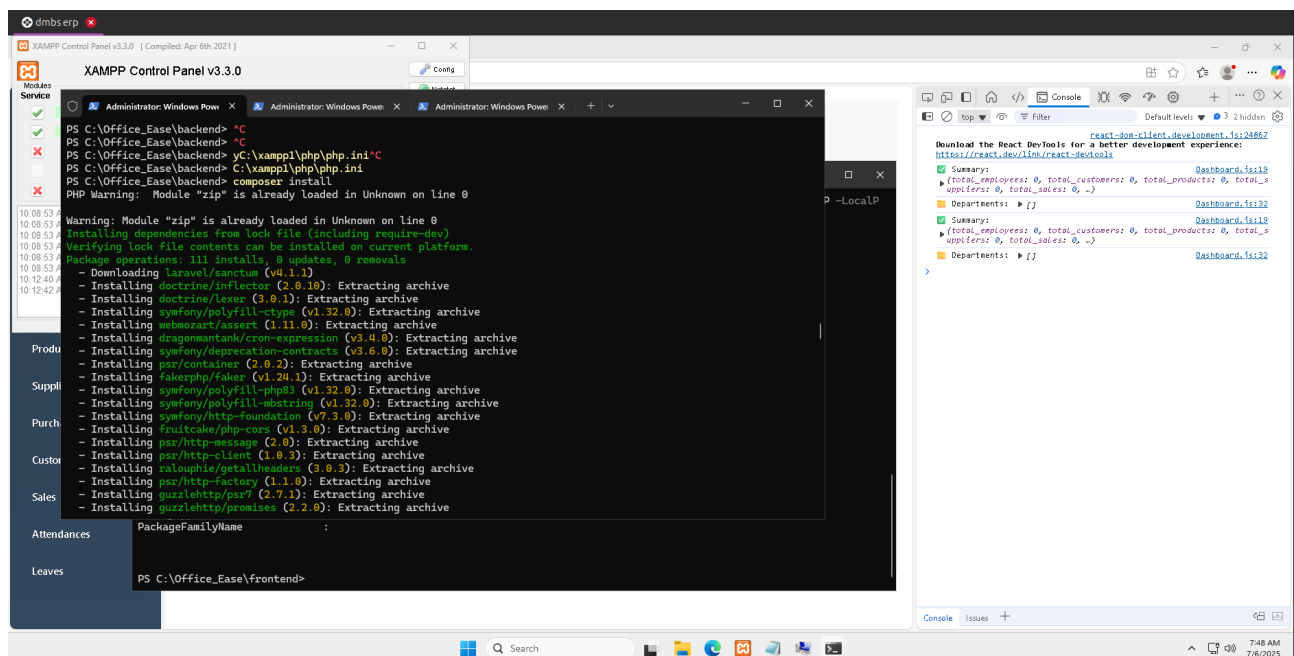
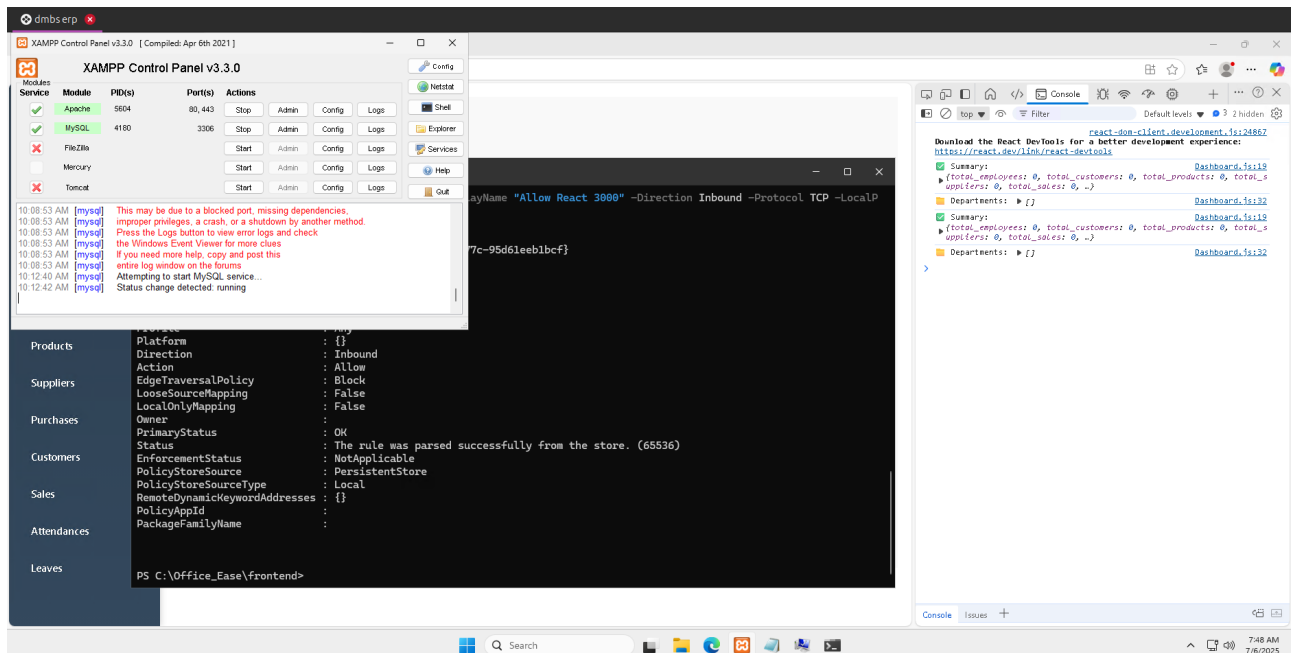
Name                          : {aca8cd3e-868b-4cd9-ac0a-286bd7a20750}
DisplayName                   : Allow TCP 8000
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                     : {}
Direction                    : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
```

```
Administrator: Windows Powe X + -
PS C:\Office_Ease\frontend> New-NetFirewallRule -DisplayName "Allow React 3000" -Direction Inbound -Protocol TCP -LocalP
ort 3000 -Action Allow -Profile Any

Name                          : {ab7945d2-d890-410e-b77c-95d61eeb1bcf}
DisplayName                   : Allow React 3000
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                     : {}
Direction                    : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId                   :
PackageFamilyName             :

PS C:\Office_Ease\frontend>
```

started apache and mysql service using xaamp



```
PS C:\Office_Ease\backend> php artisan key:generate
PHP Warning: Module "zip" is already loaded in Unknown on line 0

Warning: Module "zip" is already loaded in Unknown on line 0

[INFO] Application key set successfully.

PS C:\Office_Ease\backend>
PS C:\Office_Ease\backend> php artisan key:generate
PHP Warning: Module "zip" is already loaded in Unknown on line 0

Warning: Module "zip" is already loaded in Unknown on line 0
```

```
PS C:\Office_Ease\backend> php artisan migrate
PHP Warning: Module "zip" is already loaded in Unknown on line 0

Warning: Module "zip" is already loaded in Unknown on line 0

WARN The database 'minierp' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [yes]
> yes

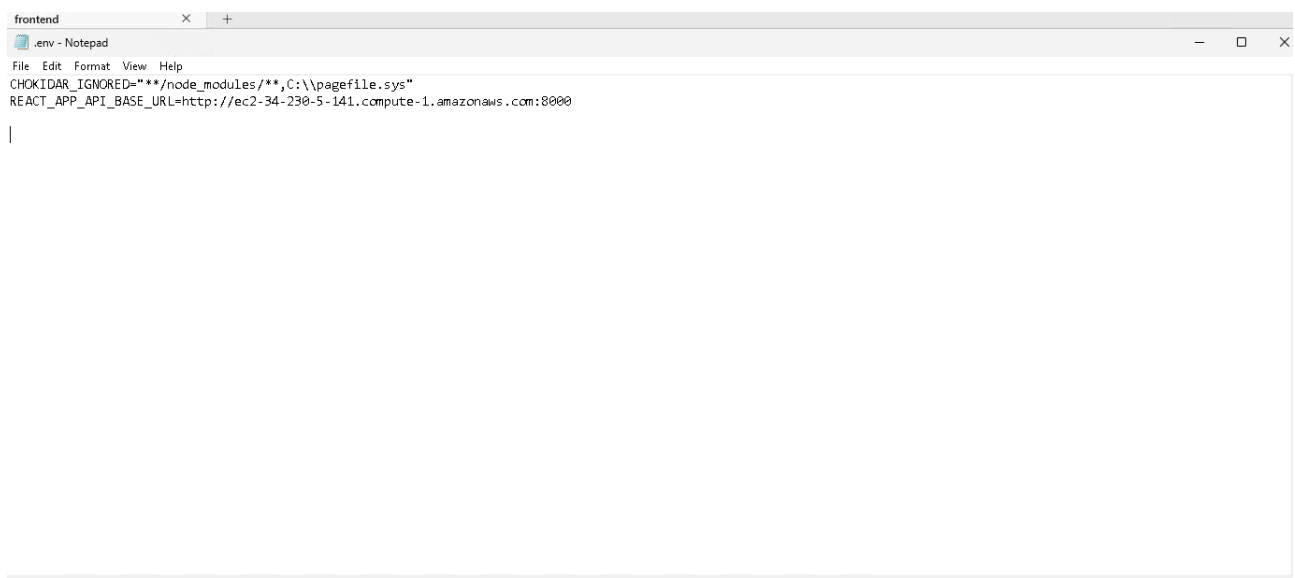
INFO Preparing database.
```

```
2025-07-04 11:05:32 /api/admin/login ..... ~ 518.23ms
2025-07-04 11:05:32 /api/admin/login ..... ~ 524.89ms
2025-07-04 11:05:32 /api/admin/login ..... ~ 511.87ms
2025-07-04 11:14:29 /api/admin/login ..... ~ 3s
2025-07-04 11:14:32 /api/admin/login ..... ~ 2s
PS C:\Office_Ease\backend> php artisan db:seed --class=AdminSeeder
PHP Warning: Module "zip" is already loaded in Unknown on line 0

Warning: Module "zip" is already loaded in Unknown on line 0

INFO Seeding database.
```

Added in frontend/.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

```
frontend
AdminLogin.js - Notepad
File Edit Format View Help
import React, { useState } from 'react';
import axios from 'axios';

function AdminLogin() {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');

  // read from env, fallback to local
  const API_BASE = process.env.REACT_APP_API_BASE || 'http://127.0.0.1:8000';

  const handleLogin = async () => {
    try {
      const res = await axios.post(
        `${API_BASE}/api/admin/login`,
        { username, password },
        { withCredentials: true } // if you need cookies / sanctum
      );

      const { token } = res.data;
      localStorage.setItem('adminToken', token);
      setError('');
      window.location.href = '/admin/dashboard';
    } catch (err) {
      setError('Invalid credentials');
    }
  };

  return (
    <div style={styles.container}>

```

added in backend/.env

```
# APP_MAINTENANCE_STORE=database

PHP_CLI_SERVER_WORKERS=4

BCRYPT_ROUNDS=12
CORS_ALLOWED_ORIGINS=http://localhost:3000,http://ec2-34-230-5-141.compute-1.amazonaws.com:3000
SANCTUM_STATEFUL_DOMAINS=localhost:3000,ec2-34-230-5-141.compute-1.amazonaws.com:3000

LOG_CHANNEL=stack
LOG_STACK=single
LOG_DEPRECATIONS_CHANNEL=null
```

public dns added in backend/config/cors.php

```
cors.php - Notepad
File Edit Format View Help
<?php

return [

    'paths' => ['api/**'],

    'allowed_methods' => ['*'],

    // Allow React frontend on localhost:3000
    'allowed_origins' => ['http://localhost:3000', 'http://ec2-34-230-5-141.compute-1.amazonaws.com:3000'],

    'allowed_origins_patterns' => [],

    'allowed_headers' => ['*'],

    'exposed_headers' => [],

cors.php - Notepad
File Edit Format View Help
<?php

return [

    'paths' => ['api/**'],

    'allowed_methods' => ['*'],

    // Allow React frontend on localhost:3000
    'allowed_origins' => ['http://ec2-34-230-5-141.compute-1.amazonaws.com:3000', 'http://localhost:3000'],

    'allowed_origins_patterns' => [],

    'allowed_headers' => ['*'],

    'exposed_headers' => [],

    'max_age' => 0,

    'supports_credentials' => true

];
```

in frontend/src/utlis/apiClient.js

```
utils  x  config  x  +
apiClientjs - Notepad
File Edit Format View Help
Import axios from 'axios';

const API_BASE = 'http://ec2-34-230-5-141.compute-1.amazonaws.com:8000' || 'http://127.0.0.1:8000';

const apiClient = axios.create({
  baseURL: `${API_BASE}/api`,
});

apiClient.interceptors.request.use(config => {
  const token = localStorage.getItem('adminToken');
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});

export default apiClient;
```

in frontend/src/App.js

```
Appjs - Notepad
File Edit Format View Help

<Router>
  <div style={{ display: 'flex' }}>
    {!hideNavbar && <Navbar />}
    <div
      style={{
        marginLeft: !hideNavbar ? '220px' : '0',
        padding: '20px',
        width: '100%',
      }}
    >
      <Routes>
        <Route path="/admin/login" element={<AdminLogin />} />
        <Route path="/admin/dashboard" element={<Dashboard />} />
        <Route path="/employees" element={<EmployeeList />} />
        <Route path="/departments" element={<DepartmentList />} />
        <Route path="/categories" element={<CategoryList />} />
        <Route path="/products" element={<ProductList />} />
        <Route path="/suppliers" element={<SupplierList />} />
        <Route path="/purchases" element={<PurchaseList />} />
        <Route path="/customers" element={<CustomerList />} />
        <Route path="/sales" element={<SalesList />} />
        <Route path="/attendances" element={<AttendanceList />} />
        <Route path="/leaves" element={<LeaveList />} />
      </Routes>
    </div>
  </div>
```

starting php server to listen on all Ips from port 8000

```
vendor\laravel\framework\src\Illuminate\Database\mysqlconnection.php:53
PDOStatement::execute()

C:\Office_Ease\backend> php artisan serve --host=0.0.0.0 --port=8000
P Warning: Module "zip" is already loaded in Unknown on line 0

arning: Module "zip" is already loaded in Unknown on line 0
PHP Warning: Module "zip" is already loaded in Unknown on line 0

INFO Server running on [http://0.0.0.0:8000].

Press Ctrl+C to stop the server

2025-07-06 04:46:50 /api/admin/login ..... ~ 1s
2025-07-06 04:46:53 /api/admin/login ..... ~ 501.76ms
2025-07-06 04:54:09 /api/admin/Login ..... ~ 3s
2025-07-06 04:54:48 /api/admin/Login ..... ~ 2s
2025-07-06 05:09:50 /api/reports/summary ..... ~ 7s
```

started npm server

```
Install the latest Powershell for new features and improvements: https://aka.ms/PSWindows

PS C:\Users\Administrator>
PS C:\Users\Administrator> cd C:\Office_Ease\frontend\
PS C:\Office_Ease\frontend> npm start

> frontend@0.1.0 start
> react-scripts start

(node:1984) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is
precatd. Please use the 'setupMiddlewares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:1984) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option
deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view frontend in the browser.

Local:            http://localhost:3000
On Your Network:  http://172.31.16.125:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

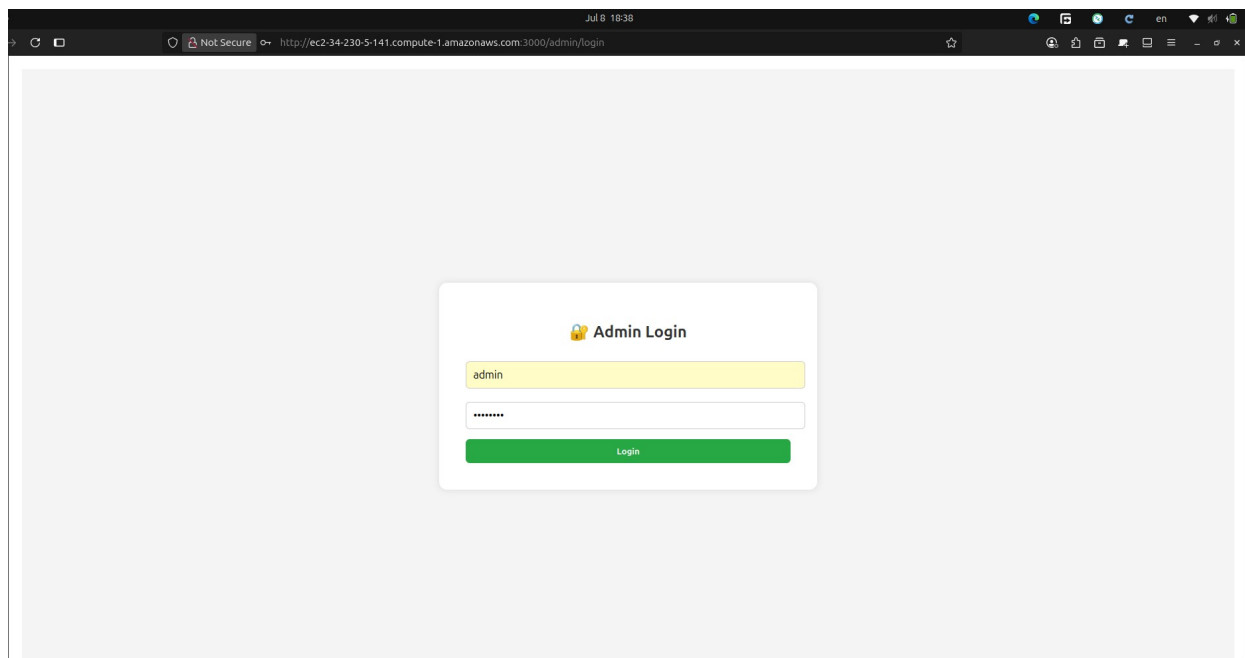
webpack compiled successfully
```

final view


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

username: admin

password: admin123

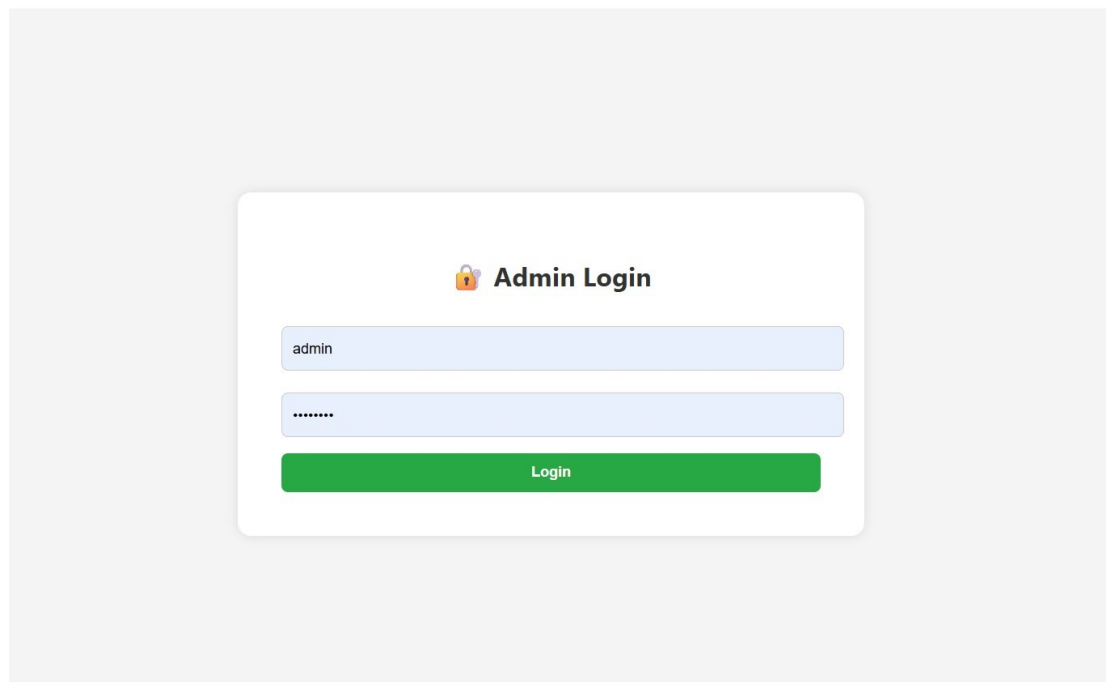


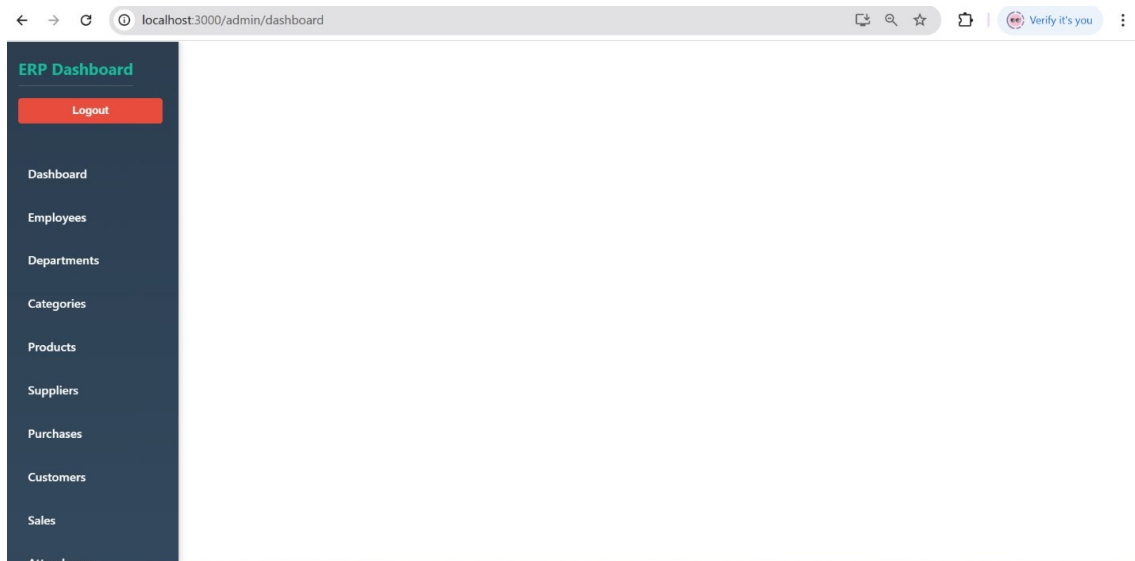
11. Submission Summary:

- **GitHub Repo:** https://github.com/niharfiza125/Office_Ease  **Included Files:**
 - o Frontend code o Backend code o SQL database o Screenshots
 - o README
 - **ZIP File Name:** ERP_1980_1983_2059.zip
-

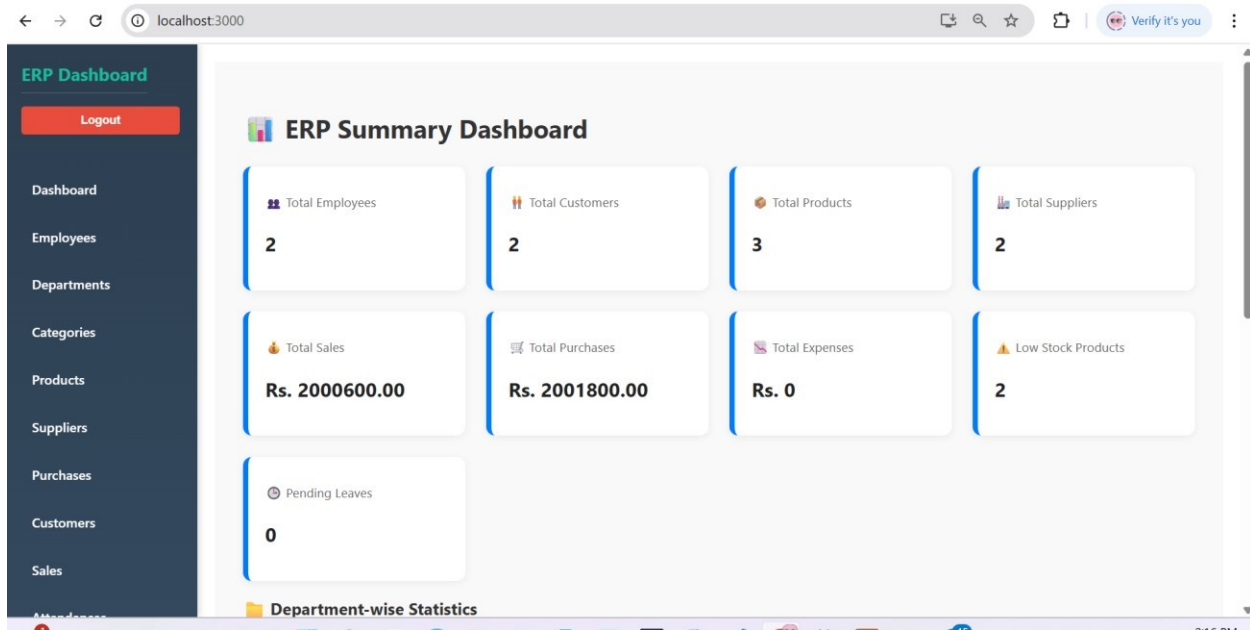
12. Screen Shots:

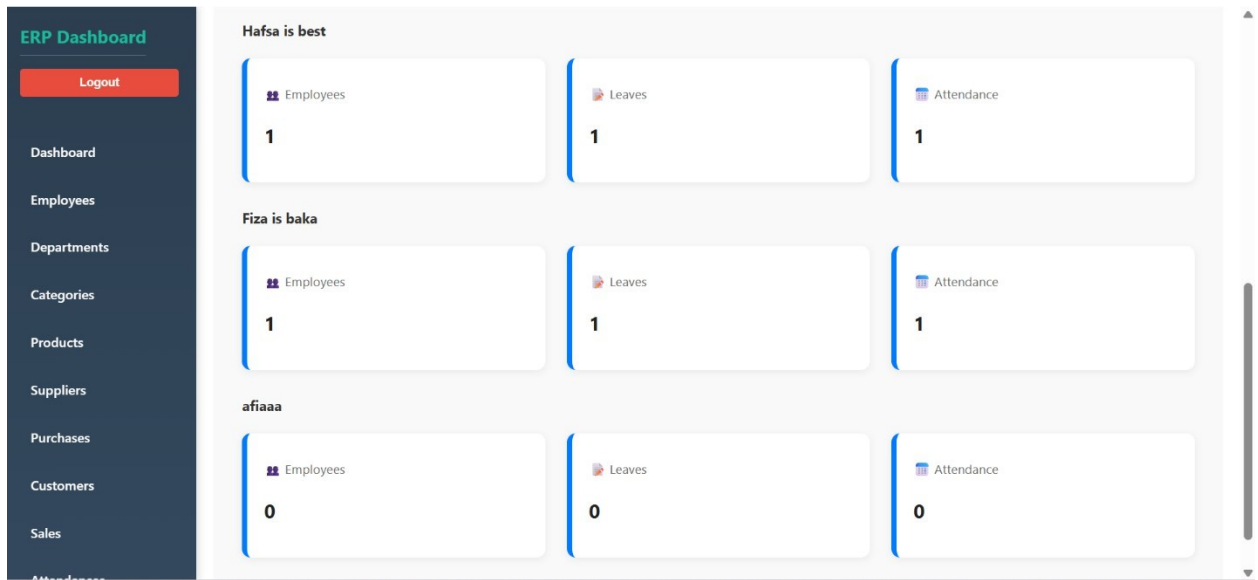
1. Admin Login:





2. Dashboard:





ERP Dashboard

Logout

Dashboard

Employees

Departments

Categories

Products

Suppliers

Purchases

Customers

Sales

Attendances

Leaves

Employee List

Name

Email

Select Department

Add Employee

Fiza — fizanihar125@gmail.com (Dept ID: 2)

Edit

Delete

hafsa — hafsajaved@gmail.com (Dept ID: 1)

Edit

Delete

ERP Dashboard

[Logout](#)[Dashboard](#)[Employees](#)[Departments](#)[Categories](#)[Products](#)[Suppliers](#)[Purchases](#)[Customers](#)[Sales](#)[Attendances](#)[Leaves](#)

Department

[Add Department](#)

Hafsa is best (ID: 1)

Fiza is baka (ID: 2)

afiaaa (ID: 3)

ERP Dashboard

[Logout](#)[Dashboard](#)[Employees](#)[Departments](#)[Categories](#)[Products](#)[Suppliers](#)[Purchases](#)[Customers](#)[Sales](#)[Attendances](#)[Leaves](#)

Category List

[Add Category](#)

best [Edit](#) [Delete](#)

good [Edit](#) [Delete](#)

ERP Dashboard

[Logout](#)[Dashboard](#)[Employees](#)[Departments](#)[Categories](#)[Products](#)[Suppliers](#)[Purchases](#)[Customers](#)[Sales](#)[Attendances](#)[Leaves](#)

Product List

[Add Product](#)

Fiza — SKU: gkb, Qty: 6, Price: Rs.300.00
Category ID: 1, Supplier ID: 1

[Edit](#) [Delete](#)

hafsaaa — SKU: s, Qty: 2, Price: Rs.1000000.00
Category ID: 2, Supplier ID: 1

[Edit](#) [Delete](#)

afiaaa — SKU: sddd, Qty: 123, Price: Rs.2.00
Category ID: 1, Supplier ID: 1

[Edit](#) [Delete](#)

ERP Dashboard

Logout

Dashboard

Employees

Departments

Categories

Products

Suppliers

Purchases

Customers

Sales

Attendances

Leaves

Supplier List

Name Email Add

Fizza — fizanihar125@gmail.com Edit Delete

hafsaaa — hafsa@gmail.com Edit Delete

ERP Dashboard

Logout

Dashboard

Employees

Departments

Categories

Products

Suppliers

Purchases

Customers

Sales

Attendances

Leaves

Purchase

Product ID

Quantity

Supplier ID

Add Purchase

Product: Fiza, Supplier: Fizza, Quantity: 2, Total Cost: Rs.600.00

Edit

Delete

Product: Fiza, Supplier: Fizza, Quantity: 2, Total Cost: Rs.600.00

Edit

Delete

Product: Fiza, Supplier: Fizza, Quantity: 2, Total Cost: Rs.600.00

Edit

Delete

Product: hafsaana, Supplier: hafsaana, Quantity: 2, Total Cost: Rs.2000000.00

Edit

Delete

ERP Dashboard

Logout

Dashboard

Employees

Departments

Categories

Products

Suppliers

Purchases

Customers

Sales

Attendances

Leaves

Customers

Name

Email

Phone

Address

Add Customer

Fiza — fizanihar01@gmail.com | 03420970416

uet

Edit

Delete

afiaana — fizanihar125@gmail.com | 03420970414

uethsotel

Edit

Delete

ERP Dashboard

Logout

Dashboard

Employees

Departments

Categories

Products

Suppliers

Purchases

Customers

Sales

Attendances

Leaves

Sales

Product ID

Customer ID

Quantity

Add Sale

Product: Fiza, Customer: Fiza, Quantity: 2

Delete

Product: hafsaana, Customer: afiaana, Quantity: 2

Delete

ERP Dashboard

Logout

Dashboard

Employees

Departments

Categories

Products

Suppliers

Purchases

Customers

Sales

Attendances

Leaves

Attendance

Employee ID

mm/dd/yyyy

Status (e.g., Present/Absen)

Add Attendance

Employee ID: 1, Date: 2025-07-01, Status: Present

Edit Delete

Employee ID: 2, Date: 2025-07-08, Status: Present

Edit Delete

ERP Dashboard

Logout

Dashboard

Employees

Departments

Categories

Products

Suppliers

Purchases

Customers

Sales

Attendances

Leaves

Leave

Employee ID

mm/dd/yyyy

mm/dd/yyyy

Leave Type (e.g., Sick, Cas)

Pending

Add Leave

Employee ID: 1, Type: Sick, From: 2025-07-15, To: 2025-07-25, Status: approved

Edit Delete

Employee ID: 2, Type: Sick, From: 2025-07-01, To: 2025-07-18, Status: rejected

Edit Delete

13. Conceptual schema :

1. Finalized Conceptual Schema (Entity-Level Description):

1. *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.
-

2. **Category**

Represents classifications for products.

- **category_id** (*Primary Key*): Unique identifier for a category. □ **name:** Unique category name (e.g., Electronics, Stationery).
-

3. **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.
-

4. **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.
-

5. **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.
-

6. **Department**

Units within the company.

- **department_id** (*Primary Key*): Unique identifier for a department. □ **name:** Department name (e.g., Sales, IT).
-

7. **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.
-

8. **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.
-

9. **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.
-

10. **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.

11. *Expense*

Any organizational spending.

- **expense_id** (*Primary Key*): Unique ID.
 - **description**: Nature of the expense.
 - **amount**: Monetary value. **date**: Date of expense.
-

14. 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**

15. 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

2. 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.

3. categories

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

4. 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.

5. suppliers

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

6. 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

7. 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

8. 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.

9. attendances

Attribute	Type	Constraints
id	INT	Primary Key
employee_id	INT	Foreign Key → employees(id)
date	DATE	Not Null
status	ENUM	Values: 'Present', 'Absent'

10. 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'

11. expenses

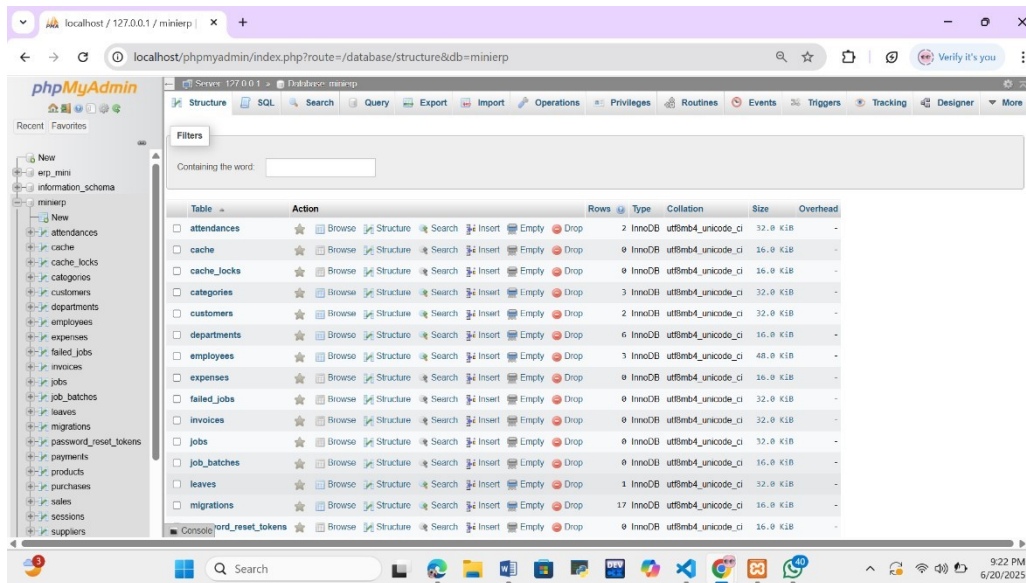
Attribute	Type	Constraints
id	INT	Primary Key
description	TEXT	Not Null
amount	DECIMAL	Not Null
date	DATE	Not Null

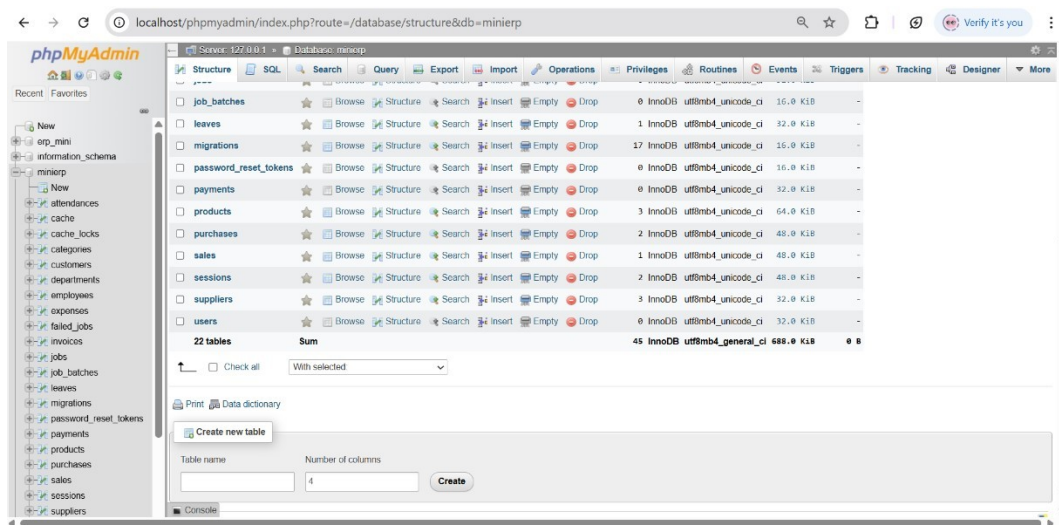
1. Normalization Summary

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

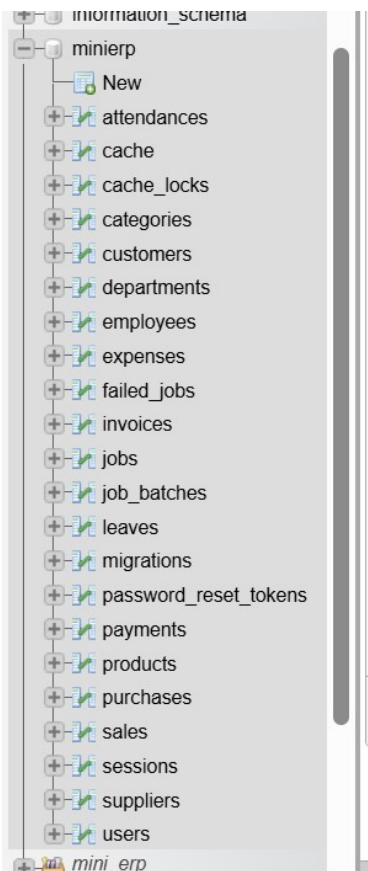
16. SQL Database Tables and Queries:

1. Database Creation:





2. Tables:



17. 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.