# **ERP System:**

## *****Project Overview*****

This project implements a **ERP (Enterprise Resource Planning) System** designed to manage and automate core business functions such as inventory tracking, employee management, supplier coordination, sales processing, purchase monitoring, leave handling, and customer relations. Built using **Laravel for the backend** and **React.js for the frontend**, the system provides a modular, API-driven architecture with real-time updates and clean UI for seamless user interaction.

The ERP includes the following key modules:

* **Inventory Management**: Categories, Products, Suppliers, Purchases
* **HR Management**: Departments, Employees, Attendances, Leaves
* **Sales Management**: Customers, Sales Records
* **Finance Overview**: Expense tracking and a consolidated dashboard for metrics

It is tailored for small to medium-sized organizations to efficiently handle their day-to-day administrative and operational tasks.

**Project Members:**

|  |  |
| --- | --- |
| ***Fiza Nihar*** |  |
| ***Afia Shah*** |  |
| ***Hafsa Javed*** |  |

***Conceptual schema :***

**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.

**🔹 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 |

**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 |
| 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' |

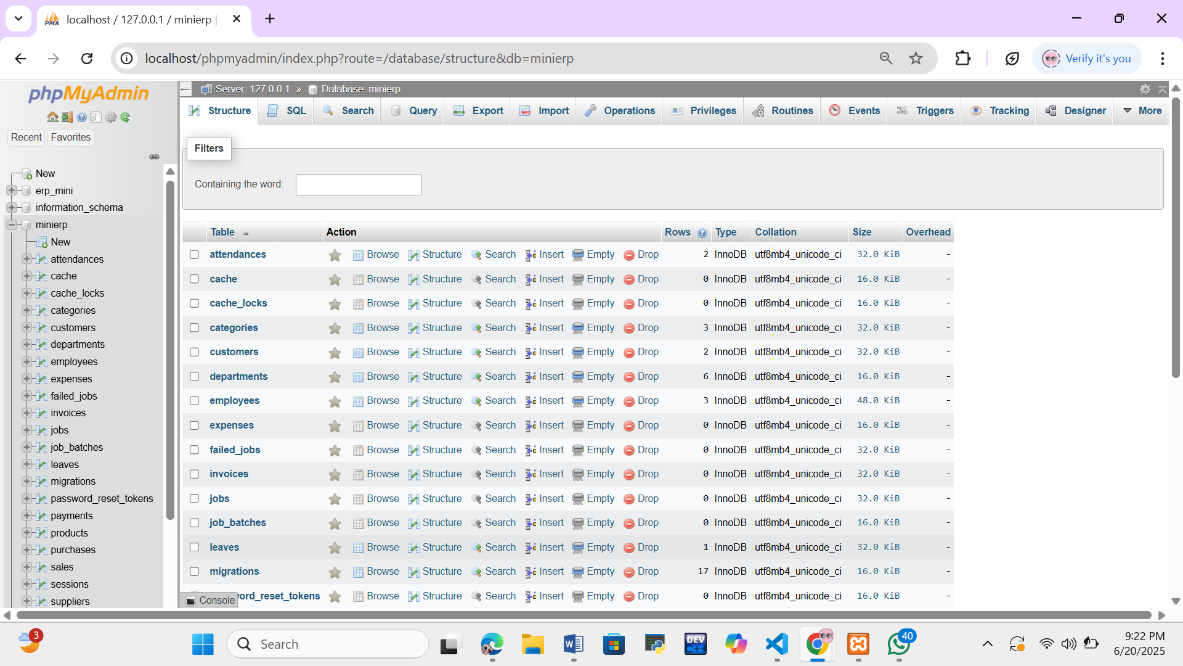
**11. expenses**

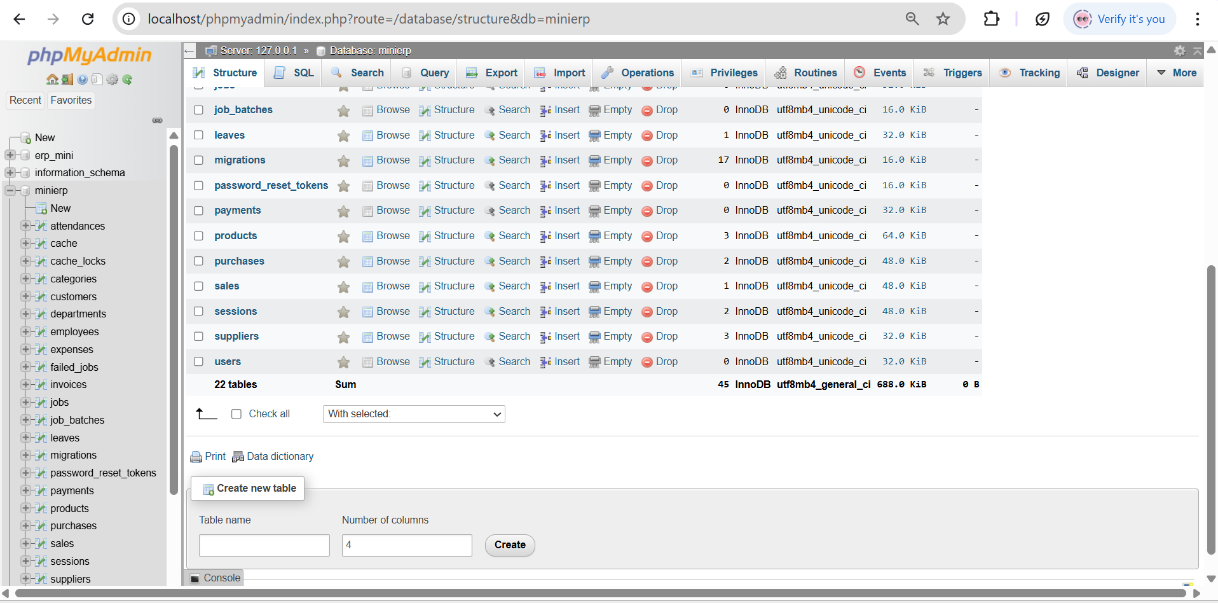
| **Attribute** | **Type** | **Constraints** |
| --- | --- | --- |
| id | INT | Primary Key |
| description | TEXT | Not Null |
| amount | DECIMAL | Not Null |
| date | DATE | Not Null |

**Normalization Summary**

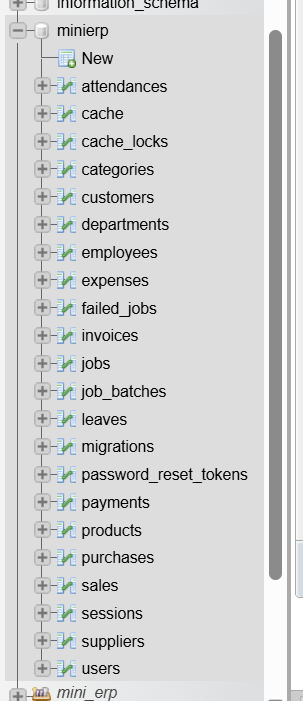
* All tables are in **1NF** (no repeating or multi-valued attributes).
* **2NF** is satisfied (no partial dependencies).
* **3NF** is ensured (no transitive dependencies).

**SQL Database Tables and Queries:**

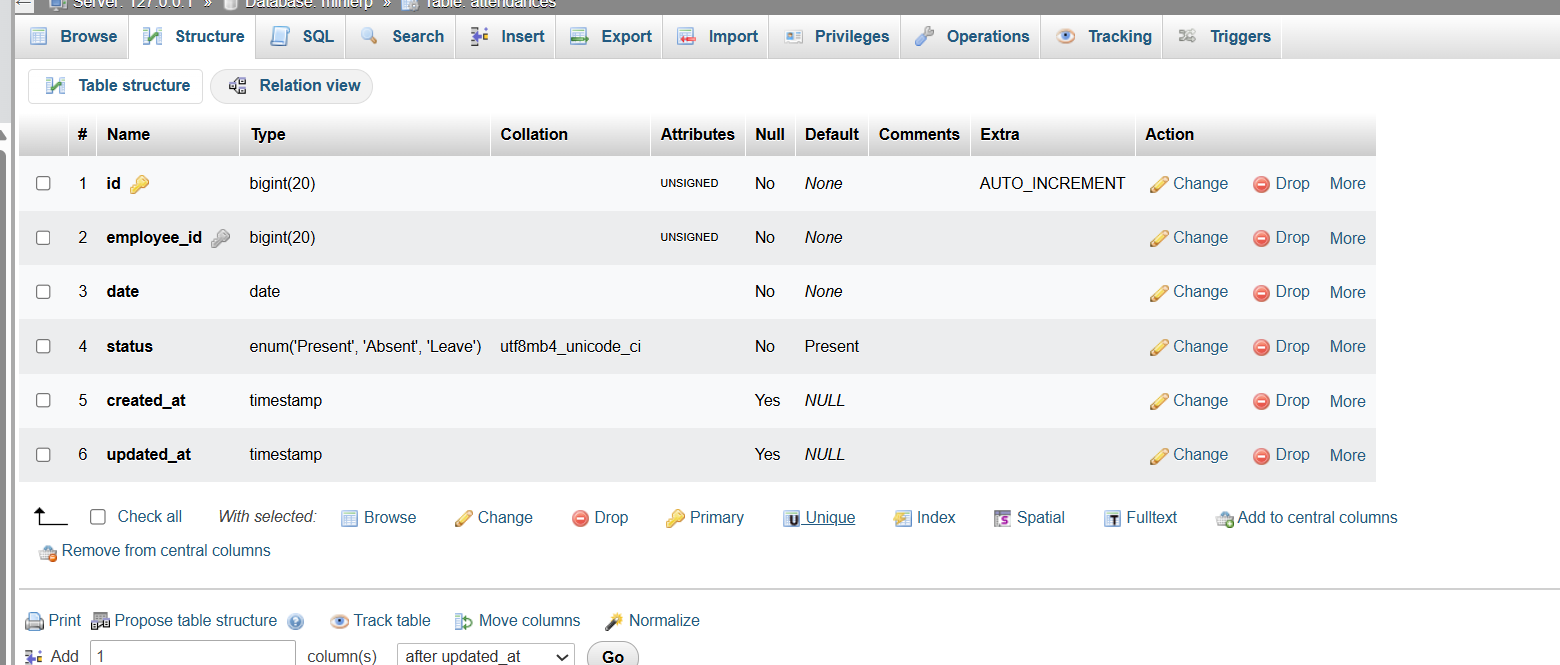
**Database Creation:**

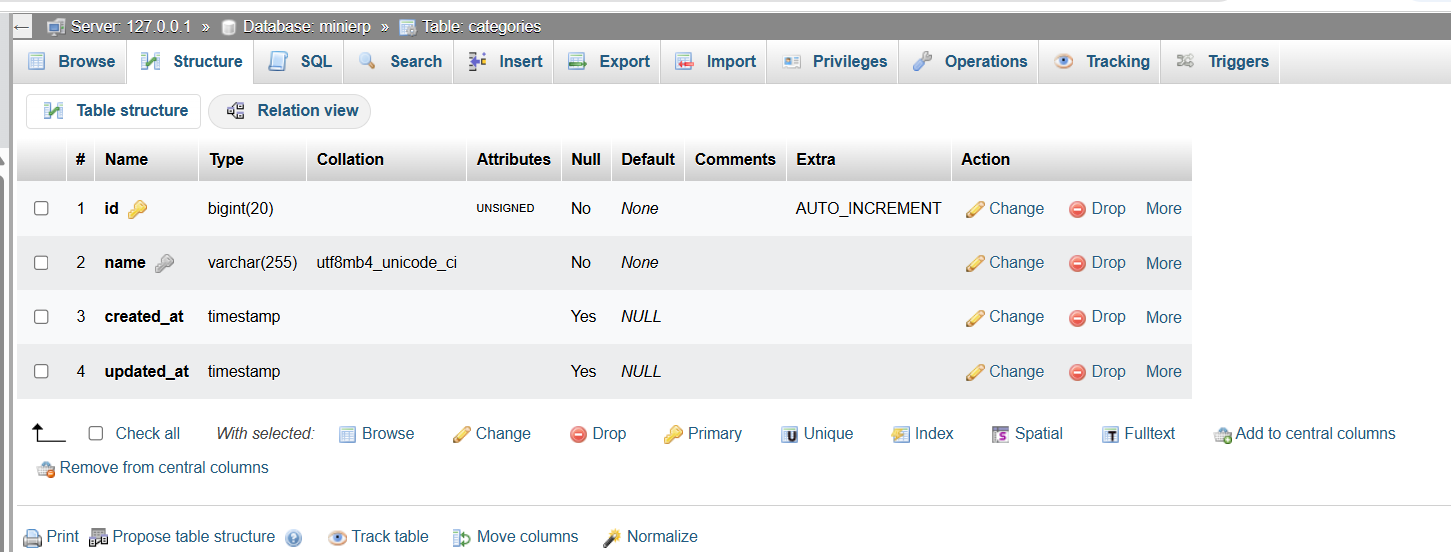


**Tables:**

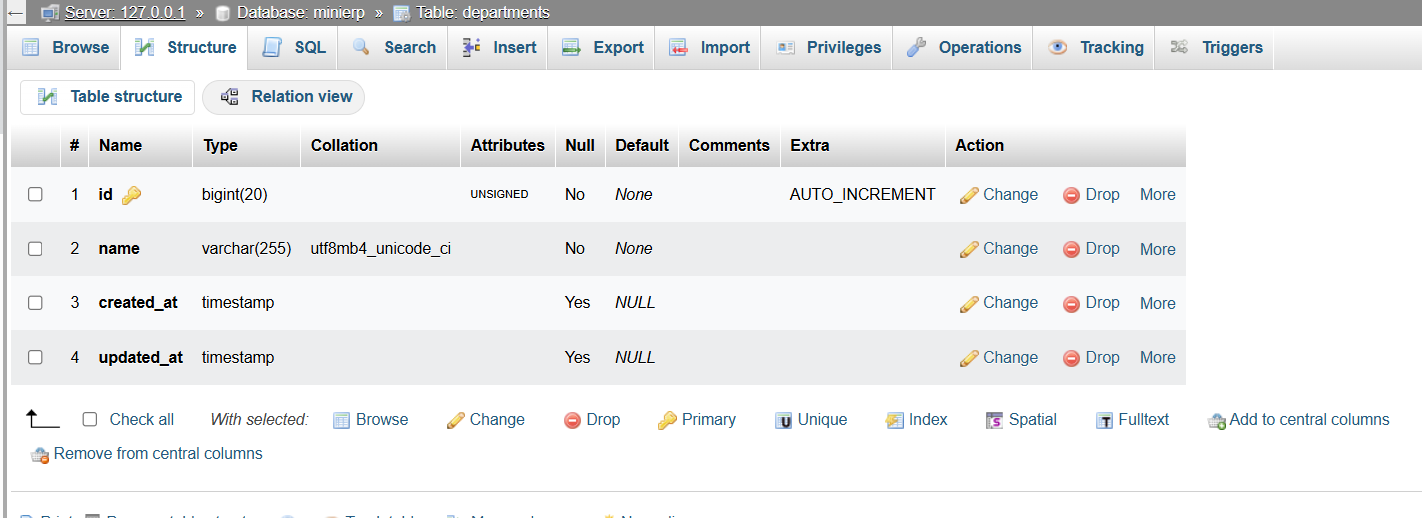


**Viewing Metadata:**

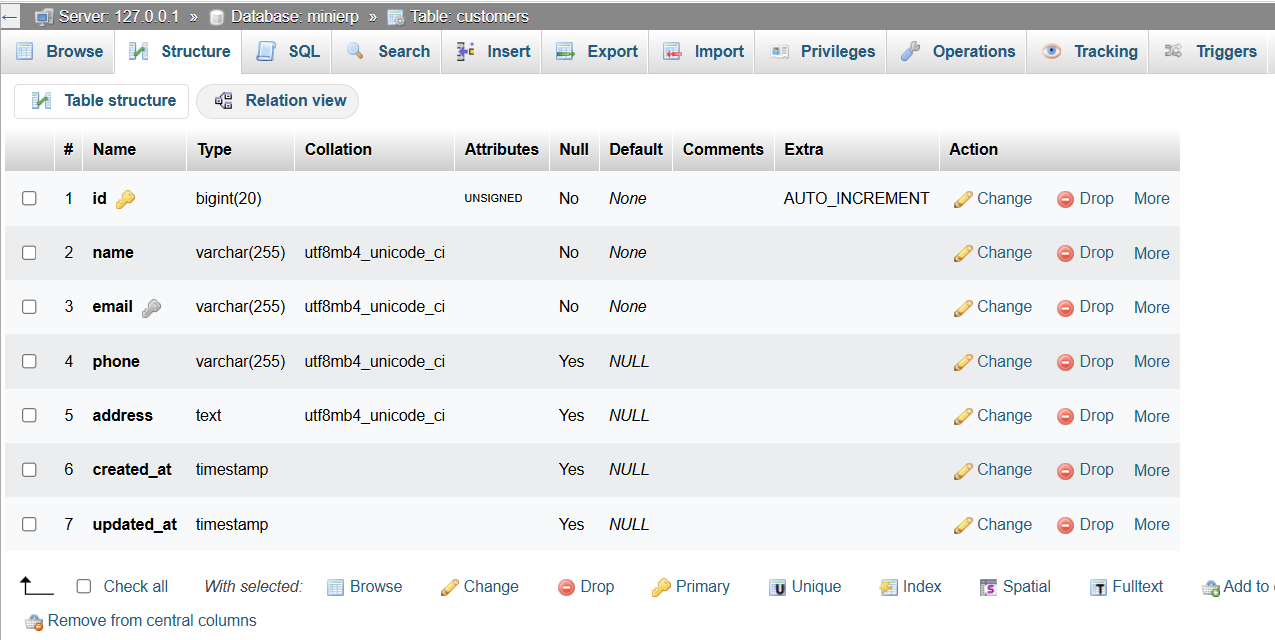
* Structure of attendance table:
* Structure of categories table:



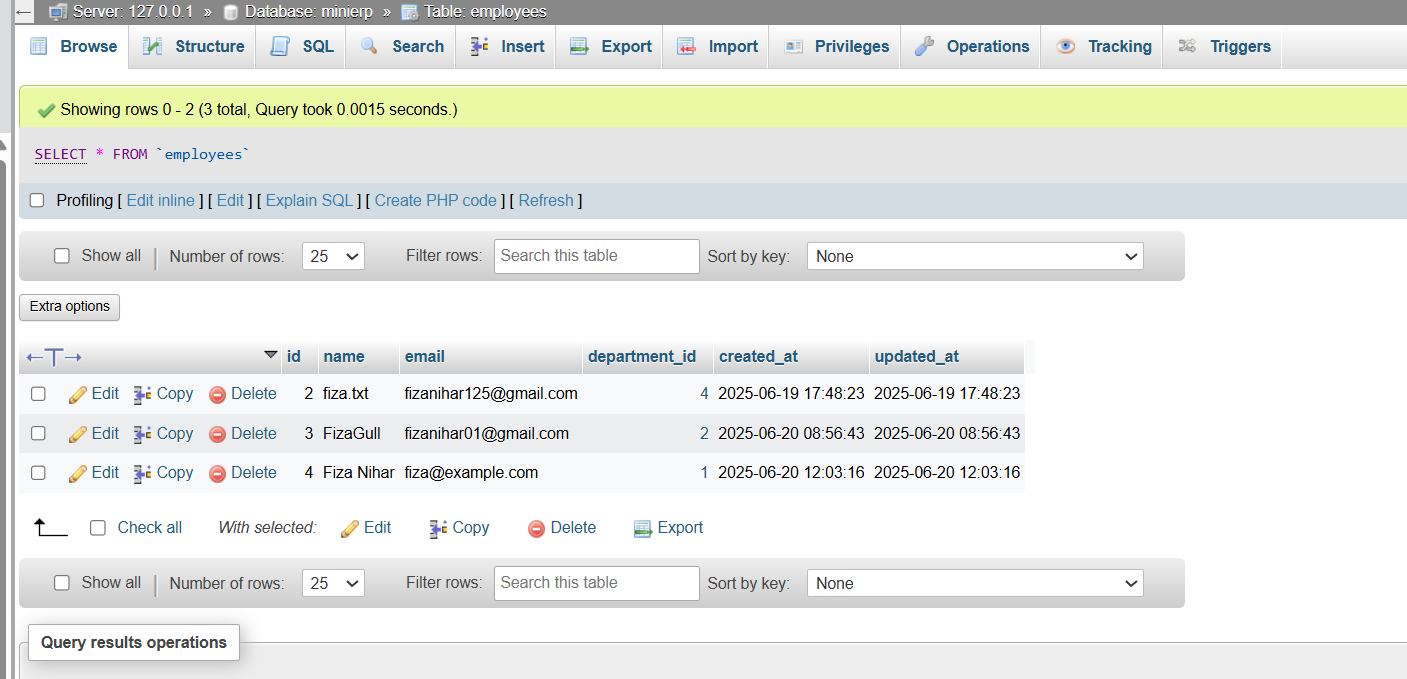
* Structure of categories table:



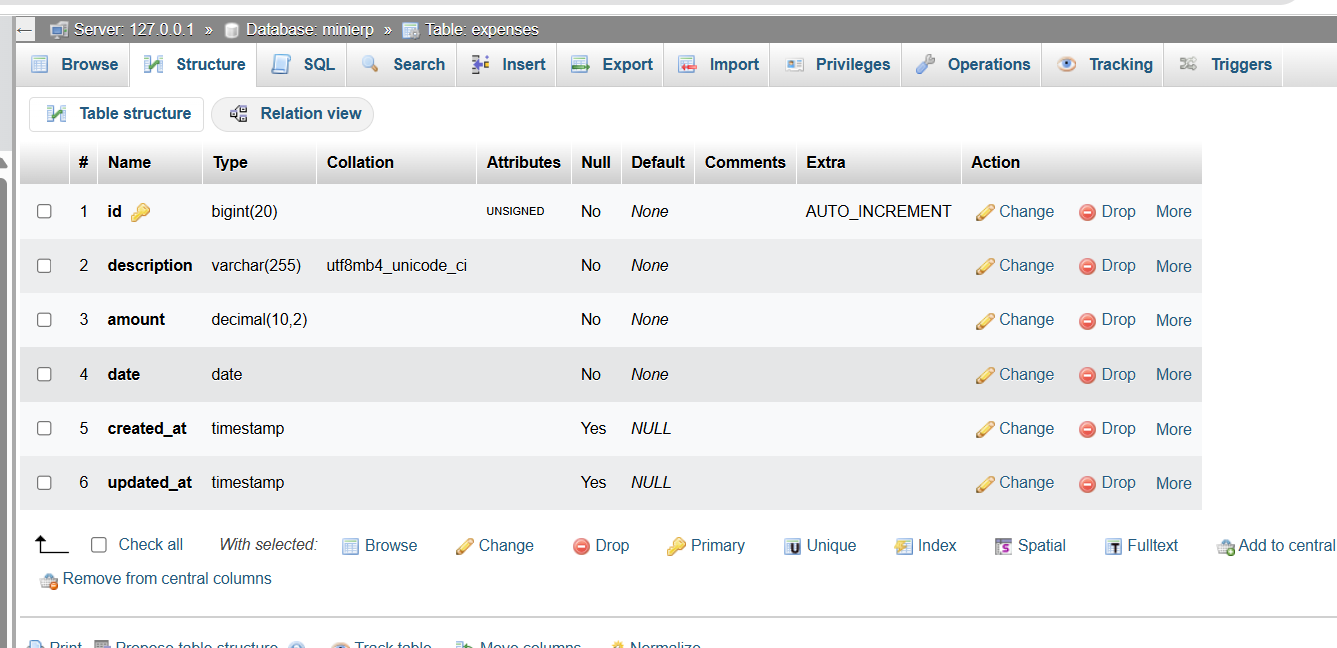
* Structure of customers table:



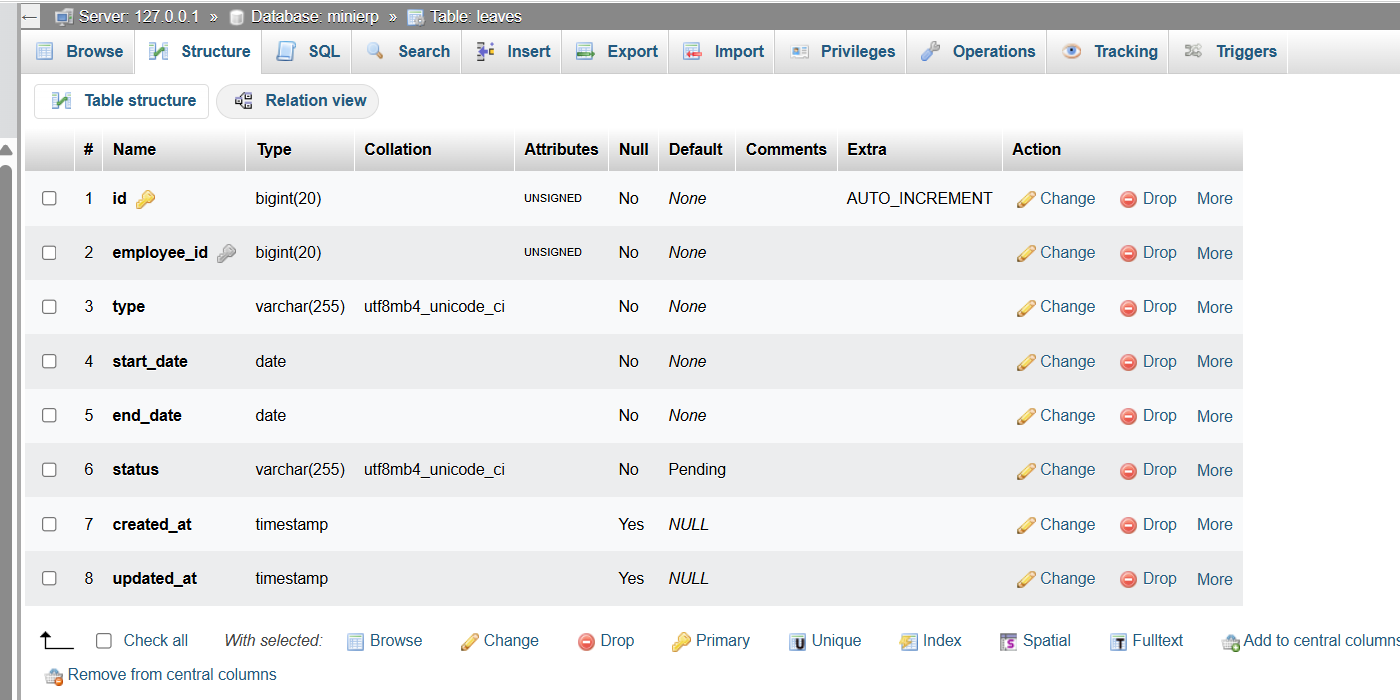
* Structure of employes table:



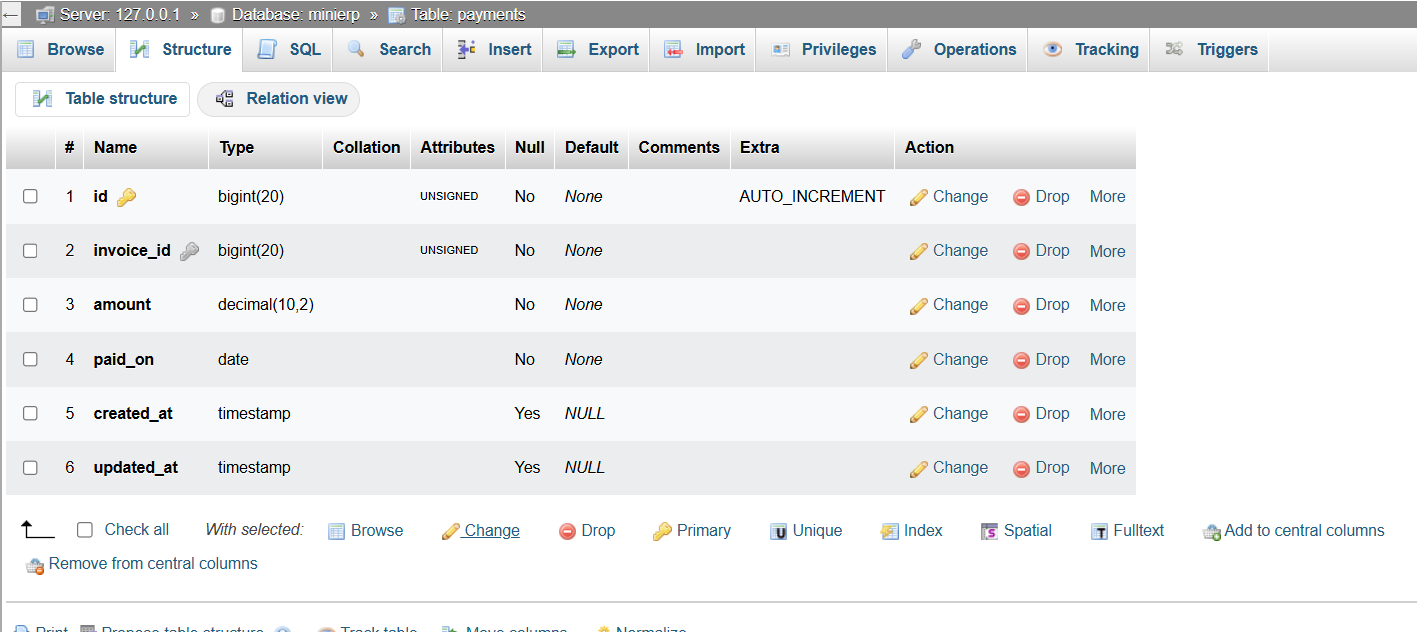
* Structure of expenses table:



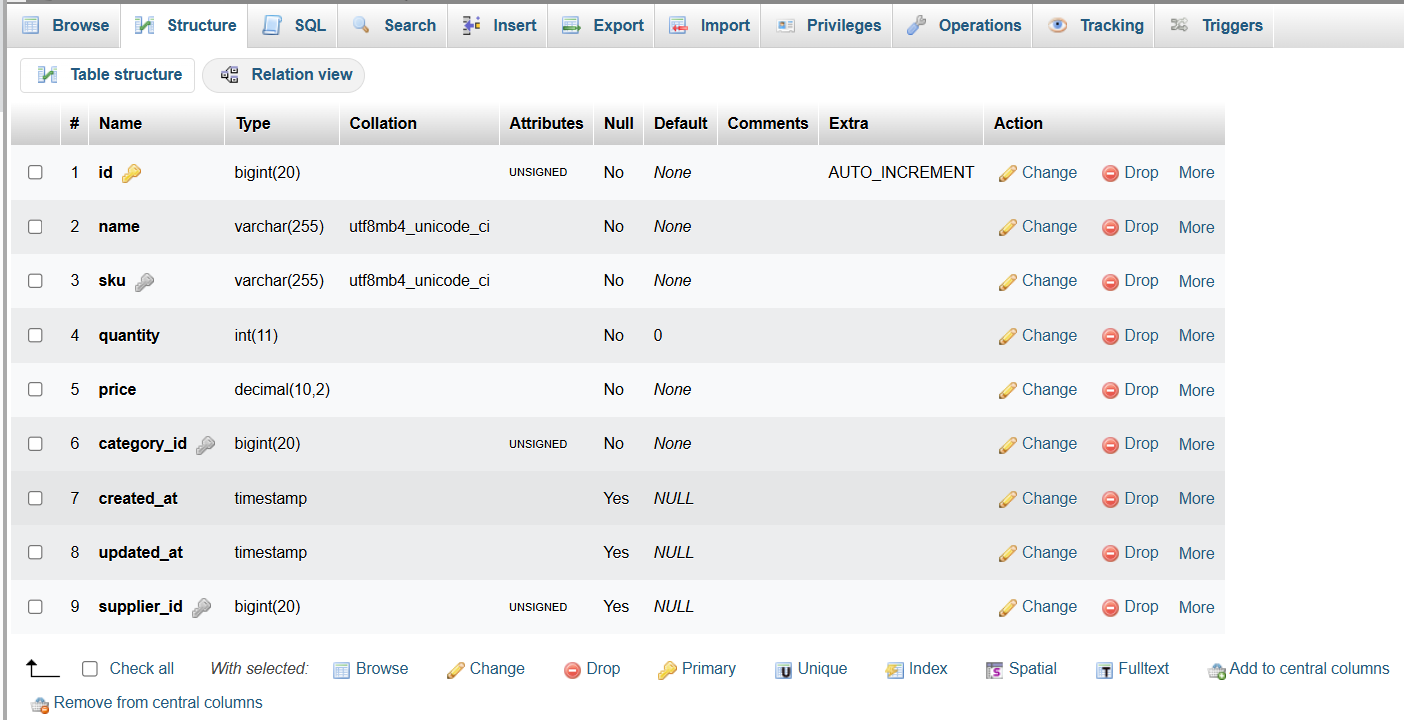
* Structure of leaves table:



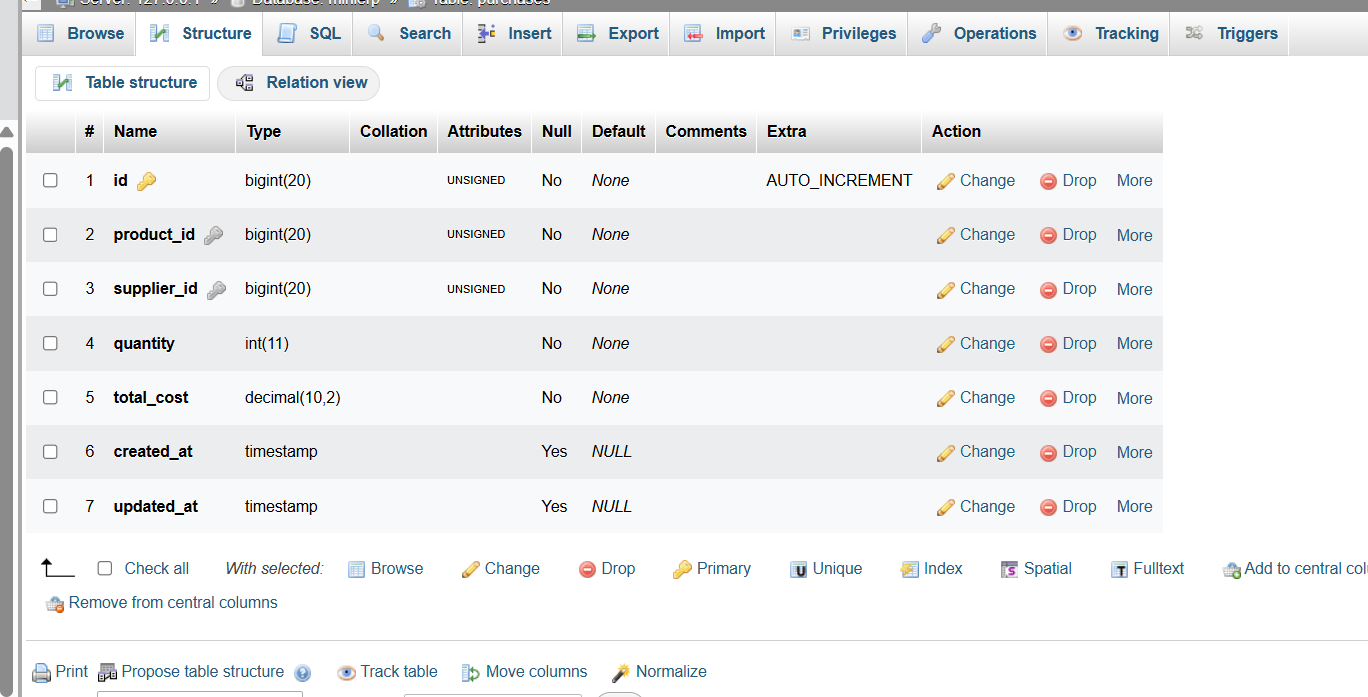
* Structure of payments table:



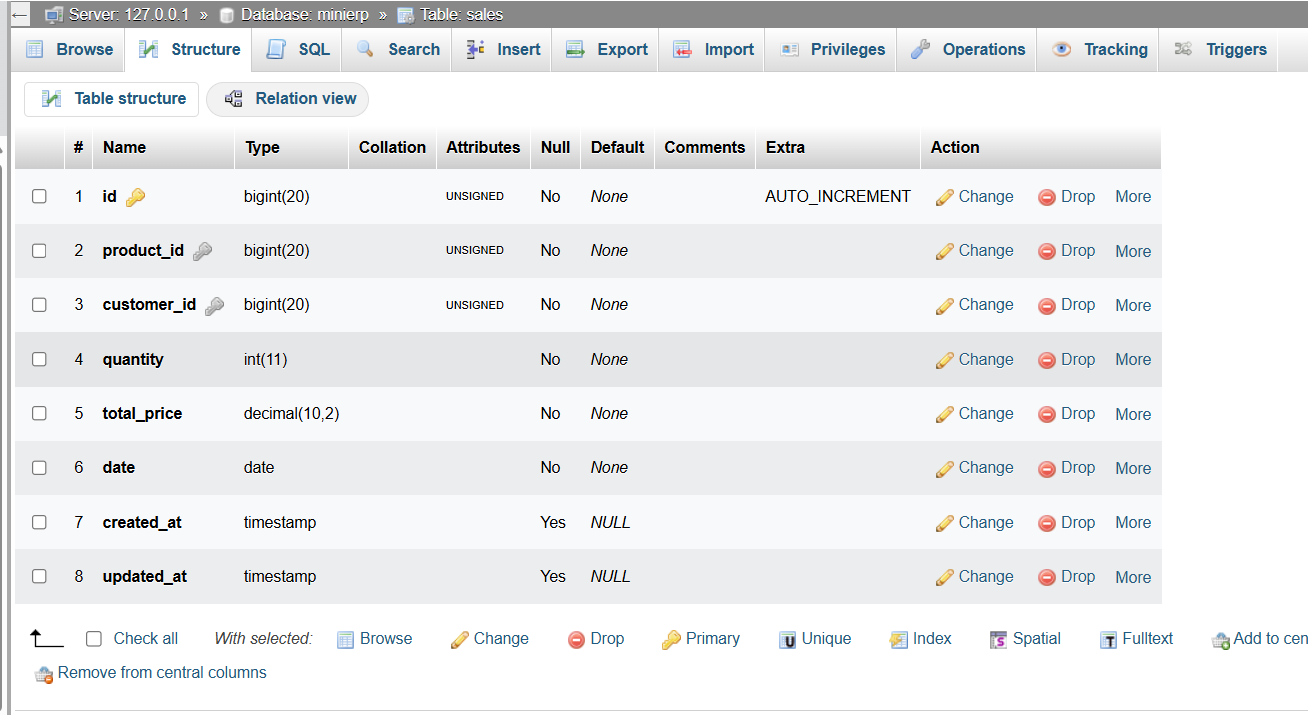
* Structure of products table:



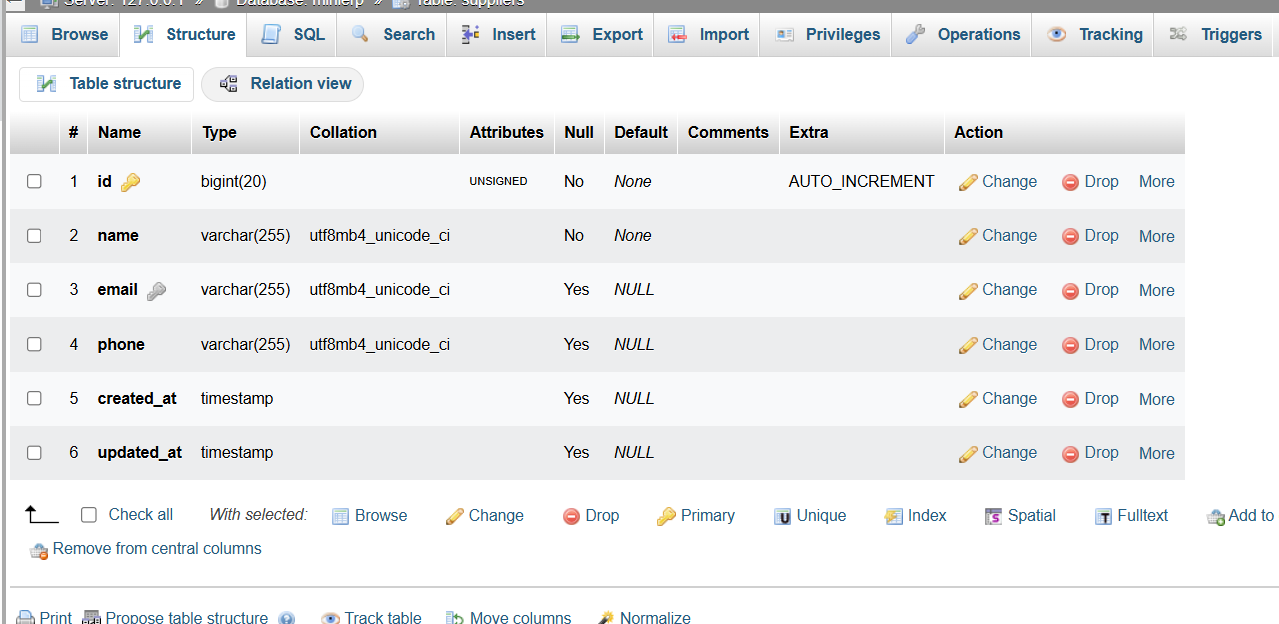
* Structure of purchases table:



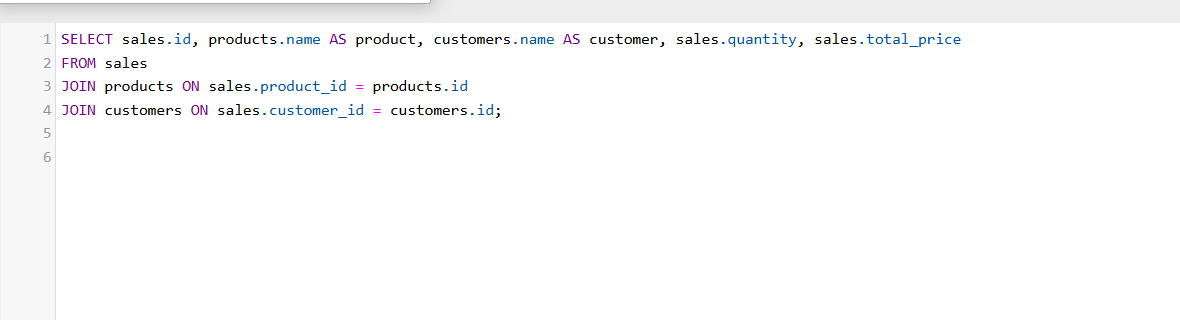
* Structure of sales table:



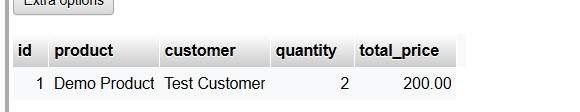
* Structure of suppliers table:



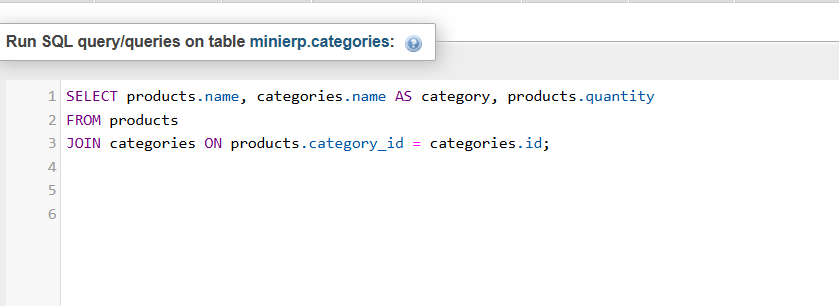
**QUERIES:**

**Query 1: Show All Sales with Product and Customer Name:**

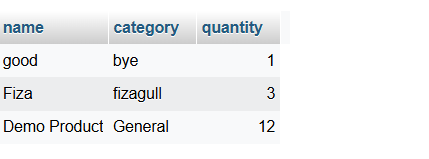
**OUTPUT:**

****

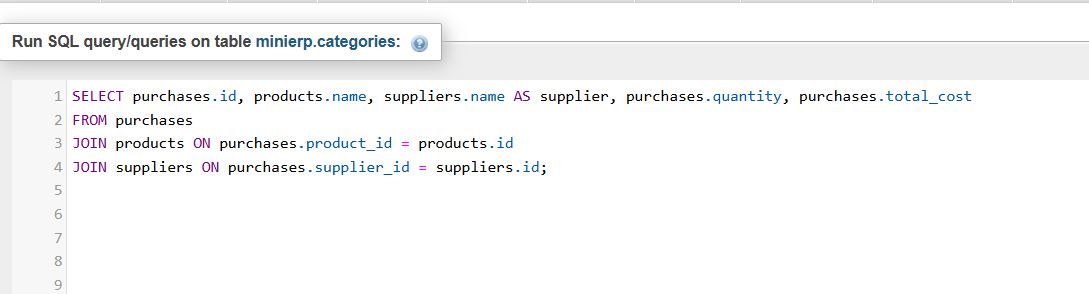
**Query 2: Products with Category Info:**

****

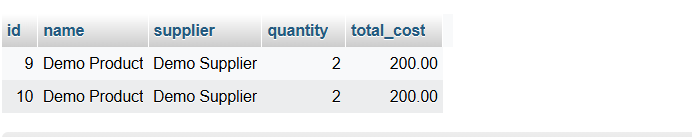
**OUTPUT:**



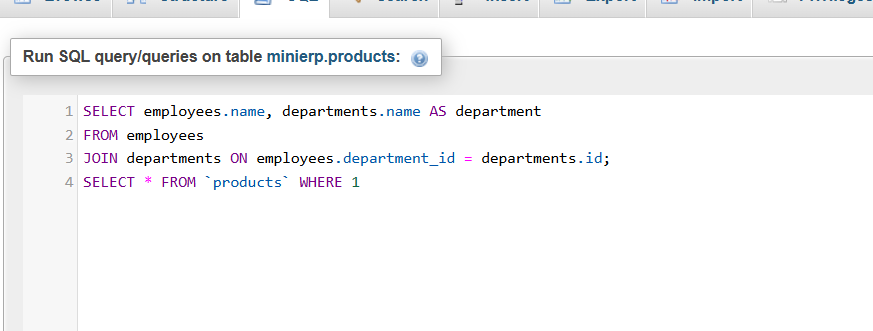
**Query 3: Purchases with Supplier Info:**

****

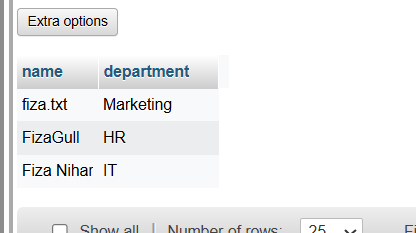
**OUTPUT:**



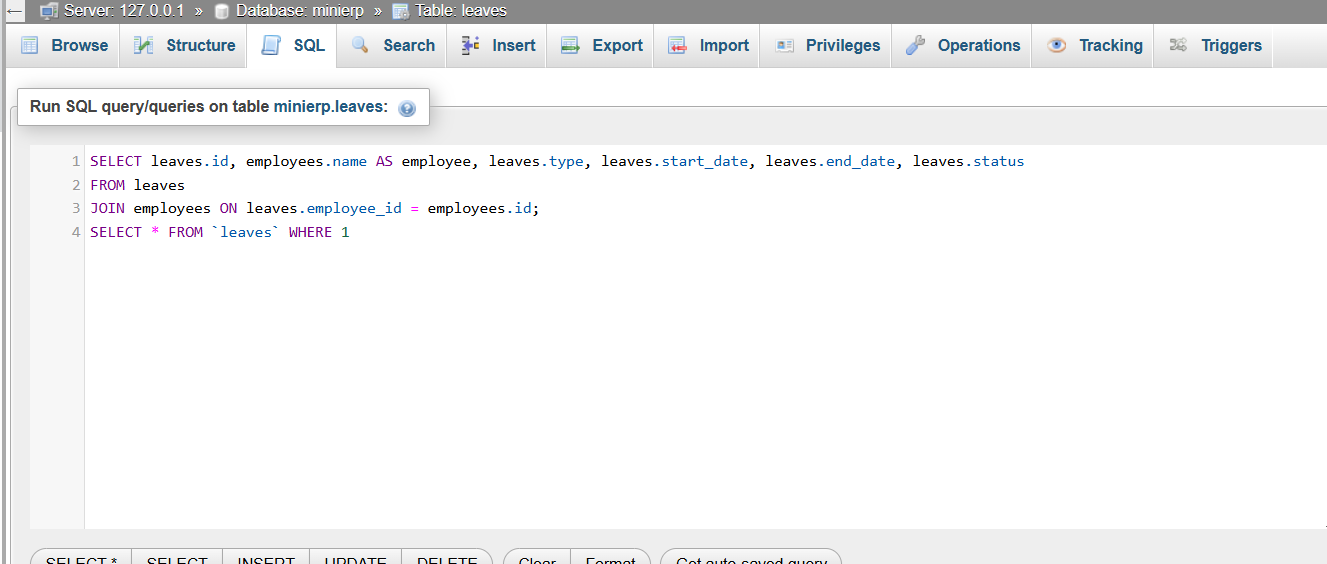
#### **Query 4: Employees with Department Names:**



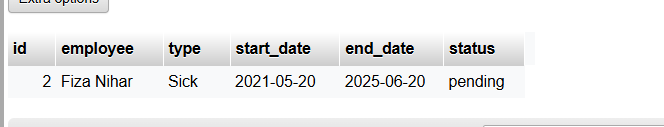
**OUTPUT:**



**Query 5: Show All Leave Requests:**

****

**OUTPUT:**



## Comprehensive Laravel Implementation Details

This section outlines how the **ERP System** was developed using the **Laravel framework**, providing details on models, controllers, routes, and frontend interaction for each module.

### Laravel Project Structure

The project follows a RESTful API architecture and is divided into modular components, each representing a core entity: Departments, Employees, Categories, Products, Suppliers, Purchases, Attendances, Leaves, Customers, and Sales.

### Laravel Models Used

Each entity in the ERP has a dedicated Eloquent Model:

* Department.php
* Employee.php
* Category.php
* Product.php
* Supplier.php
* Purchase.php
* Attendance.php
* Leave.php
* Customer.php
* Sale.php

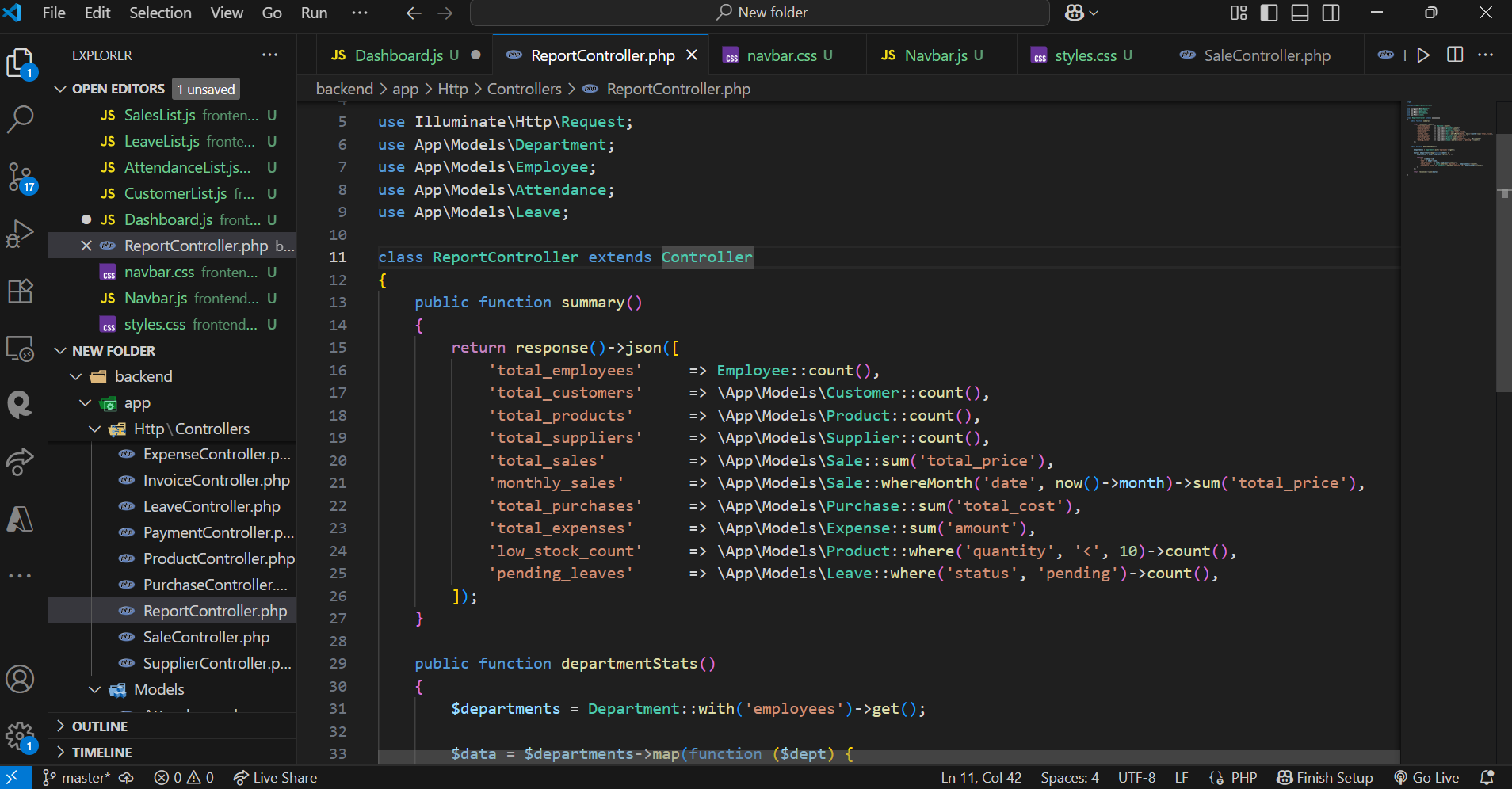
These models define table relationships and encapsulate business logic.

### Controllers

Each resource has its own controller:

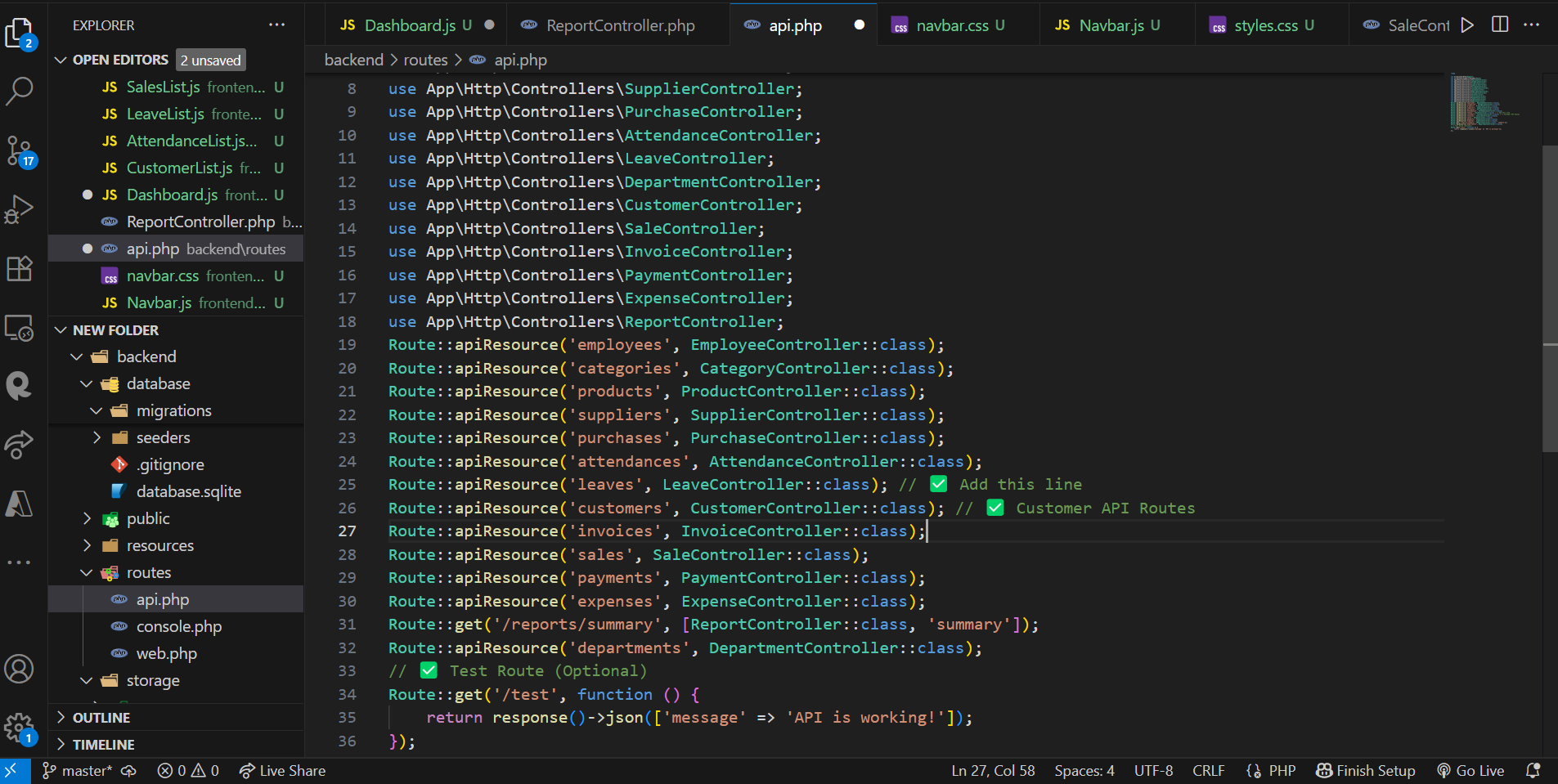
* DepartmentController: Handles department CRUD.
* EmployeeController: Manages employees and their department relationship.
* CategoryController, ProductController, SupplierController, etc.: Control respective inventories.
* PurchaseController: On storing a purchase, auto-increments product quantity.
* SaleController: Auto-decrements product quantity upon sale.
* AttendanceController and LeaveController: Manage HR activities.
* CustomerController: Manages CRM functions.
* ReportController: Generates summary metrics used in the dashboard.

Example from ReportController:



### API Routes

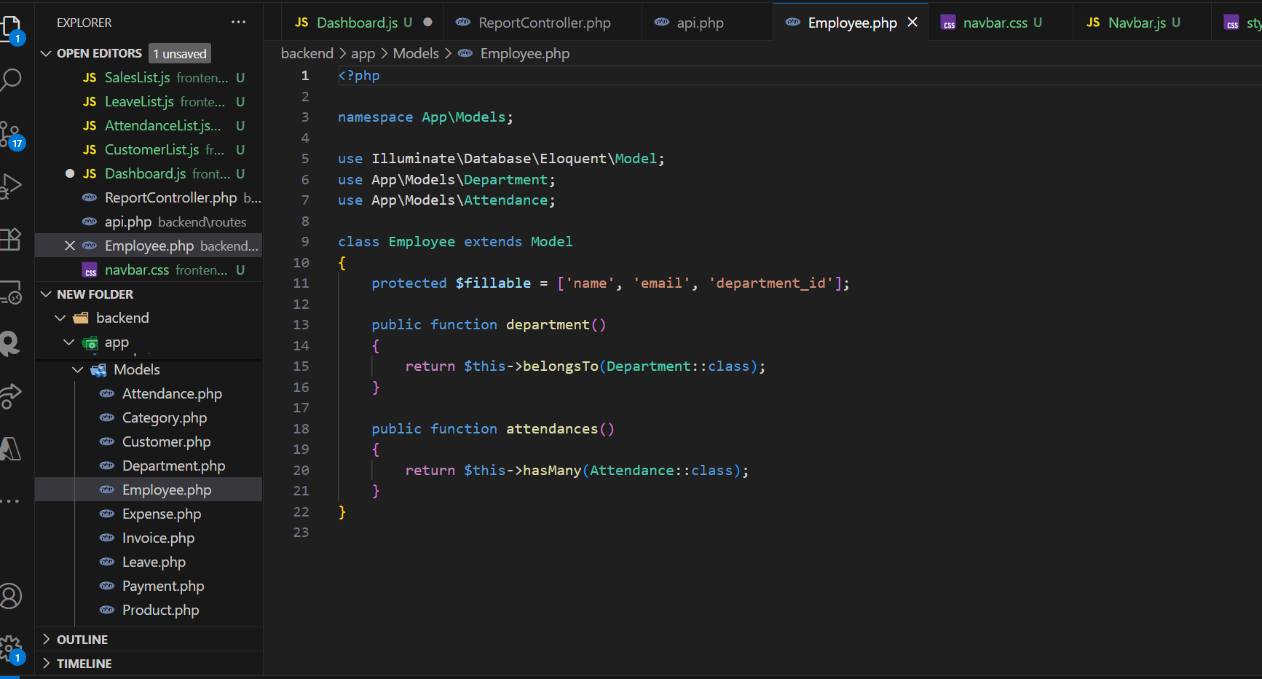
Defined in routes/api.php:

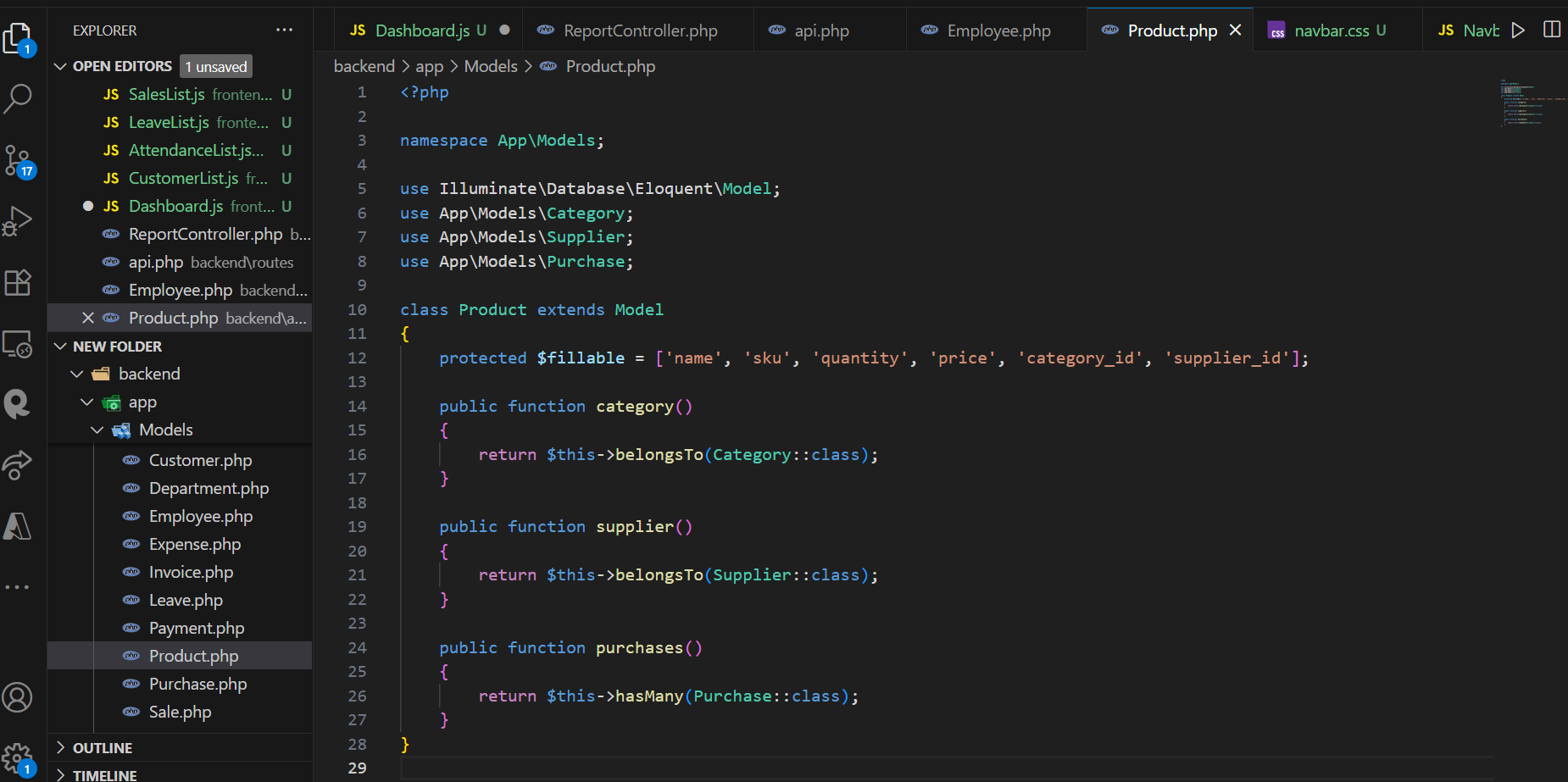


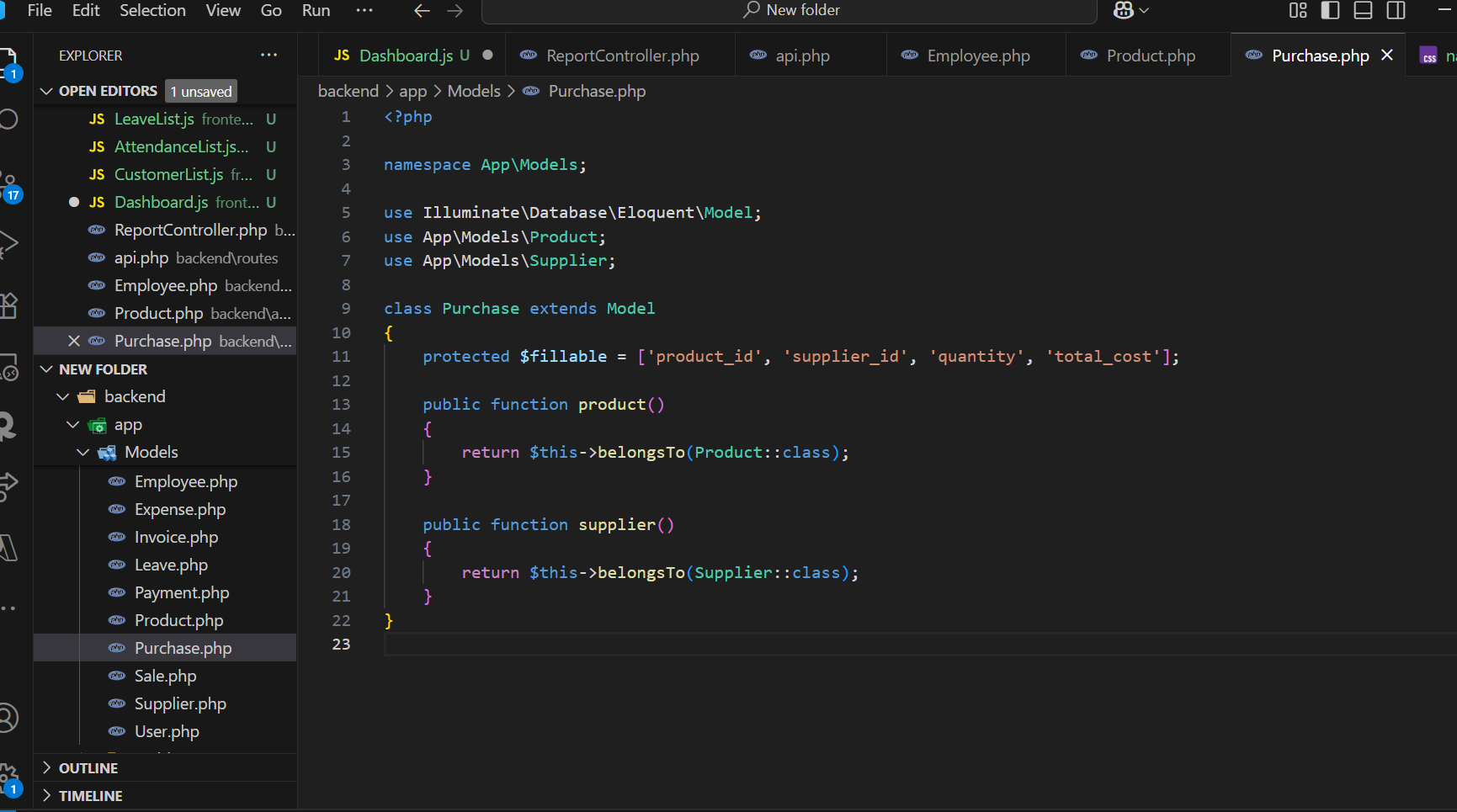
These routes are consumed by the React frontend using fetch() for GET and POST operations.

### Relationships

Defined inside models:



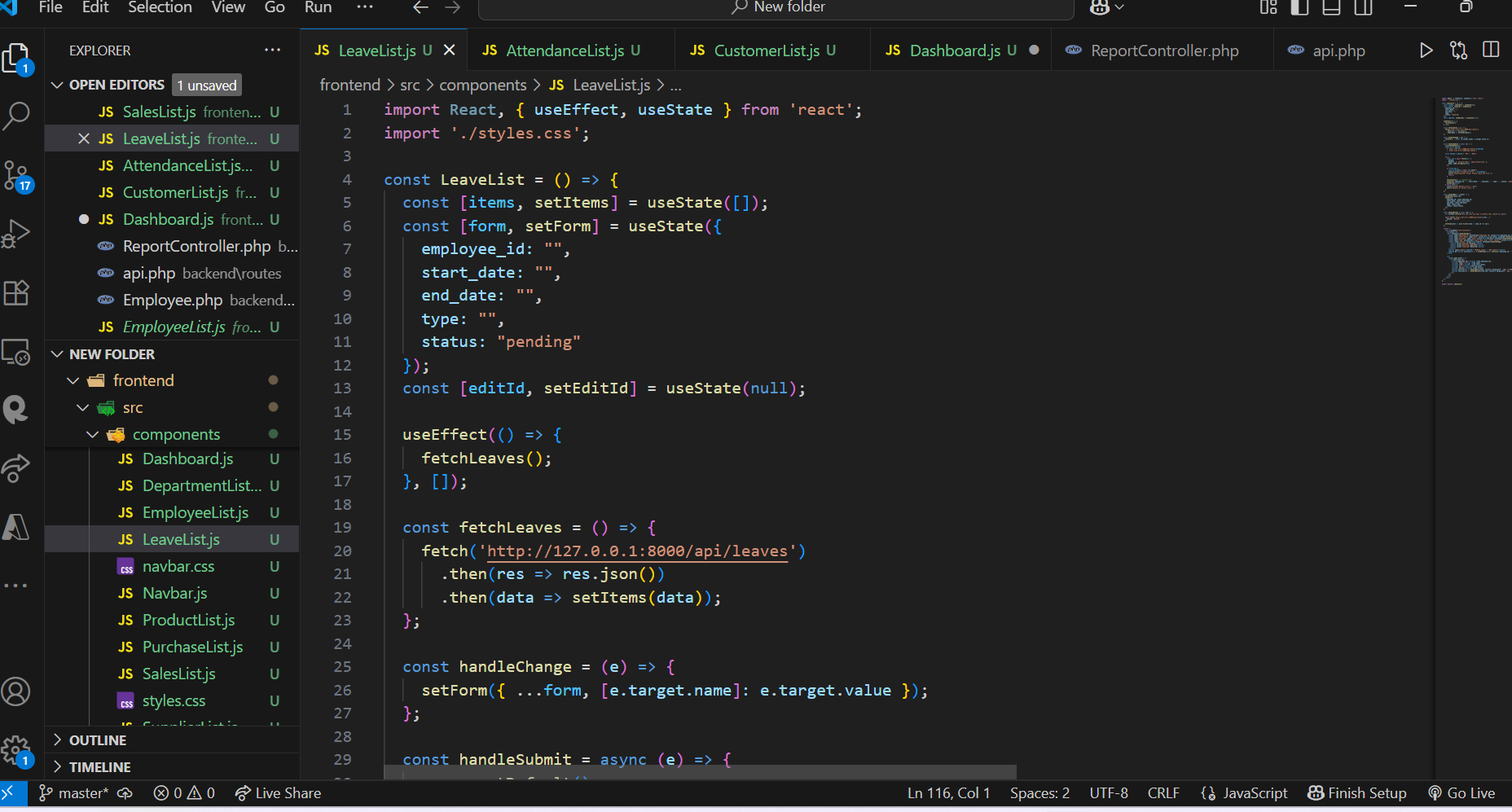




### Frontend Interaction (React)

Each module has a React component (e.g., EmployeeList.js, DepartmentList.js) that:

* Fetches data using Laravel API.
* Submits forms to create new entries.
* Shows lists with edit/delete functionality.



### Special Logic Implemented

* **Inventory Auto-Update**:
  + In PurchaseController, Product::increment('quantity', $request->quantity);
  + In SaleController, Product::decrement('quantity', $request->quantity);
* **Leave Approval Logic**:
  + Status defaults to pending and can be updated later.
* **Dashboard Summary**:
  + Data fetched from ReportController@summary and visualized on a dashboard React component using cards.

### Laravel Features Used

* Eloquent ORM for database interaction
* API Resource Routing
* Form validation via Request class or inline
* Controller-based logic separation
* Middleware (optional for future role-based access)

