**Project Directory Structure**

expense-tracker/

├── config/

│ └── db.js # Database connection setup

├── controllers/

│ ├── authController.js # User registration and login logic

│ └── expenseController.js # Expense management logic

├── models/

│ ├── User.js # User model

│ └── Expense.js # Expense model

├── routes/

│ ├── authRoutes.js # Routes for authentication

│ └── expenseRoutes.js # Routes for expense management

├── middleware/

│ └── authMiddleware.js # Authentication middleware

├── .env # Environment variables

├── .gitignore # Git ignore file

├── app.js # Main application file

├── package.json # Project dependencies and scripts

└── README.md # Project documentation

**Database Schema:**  Expenses Tables

-- Users table

CREATE TABLE Users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL

);

-- Expenses table

CREATE TABLE Expenses (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

amount DECIMAL(10, 2) NOT NULL,

date DATE NOT NULL,

category VARCHAR(100),

FOREIGN KEY (user\_id) REFERENCES Users(id)

);

**Bonus question:**  Database server comparison table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database** | **Type** | **Target Audience** | **Key Features** | **Ease of Use** |
| MySQL | Open-source | Small to medium businesses, startups | ACID compliance, good for web apps, high performance | Easy |
| PostgreSQL | Open-source | Complex apps, data-driven organizations | Advanced SQL features, extensible, highly reliable | Moderate |
| Microsoft SQL Server | Commercial | Enterprises, corporate users | Strong BI tools, analytics, integrates with Microsoft | Moderate, easy with MS tools |
| Oracle Database | Commercial | Enterprises, complex data needs | Advanced security, high availability, analytics | Complex |
| Node.js | Open-source | Developers needing non-blocking, I/O apps | Asynchronous, fast, scalable | Easy |