Full Stack Development with MERN

Project Documentation format

1. Introduction

- **Project Title:** House Rental Management System Using PHP MySQL
- **Team Members:**
  Team Leader : Shaik Jasmine
  Team Member: Bogimi Poojitha
  Team Member: ST.Md Faiz
  Team Member: K Bhanusree

2. Project Overview

- **Purpose:** The House Rental Management System is designed to simplify the rental process for landlords. It enables tracking of house details, tenants, and rent payments, while reducing manual errors and improving efficiency.
- **Features:**

  a) Admin Dashboard
  b) House Types and Houses Management
  c) Tenants Management
  d) Rent Payment Tracking
  e) Report Generation
  f) User Role Management (Admin & Staff)

3. Architecture

- **Frontend:** HTML, CSS, JavaScript, Bootstrap, AJAX
  a) Responsive UI using Bootstrap
  b) Dynamic forms and tables using JavaScript and AJAX
- **Backend:**
  a) PHP (Procedural or OOP)
  b) Handles CRUD operations for tenants, payments, houses
  c) Validates and processes form submissions
- **Database: MySQL**
  a) Tables: users, tenants, houses, payments, house_types
  b) Relationships: One-to-many (House Type → Houses, House → Tenants)

4. Setup Instructions

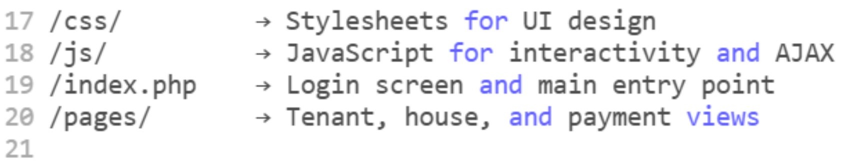
- **Prerequisites:**
  a) XAMPP / WAMP / LAMP
  b) PHP 7+
  c) MySQL

- **Installation:** Step-by-step guide to clone, install dependencies, and set up the environment variables.

  1. Clone or download the project files
  2. Copy to htdocs/ (if using XAMPP)
  3. Import the SQL file into phpMyAdmin
  4. Configure db.php with your database credentials
  5. Run the project via http://localhost/house-rental-system/

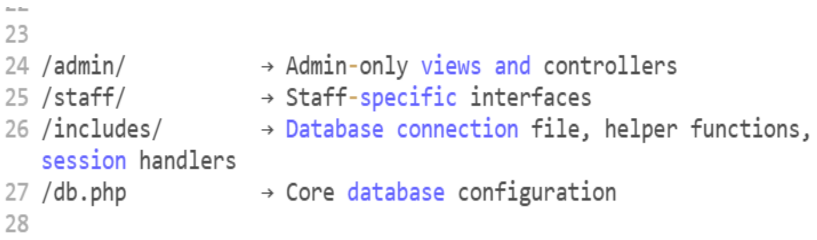
5. Folder Structure

- **Client:**
  The client-side consists of static files using **HTML**, **CSS**, **JavaScript**, and **Bootstrap**. These files handle the user interface (UI) and are located in:



```
/css/           → Stylesheets for UI design
/js/            → JavaScript for interactivity and AJAX
/index.php      → Login screen and main entry point
/pages/         → Tenant, house, and payment views
```

- **Server:** The server-side is powered by **PHP**, which handles all backend logic including database operations, session management, and user role authentication. The structure is as follows:

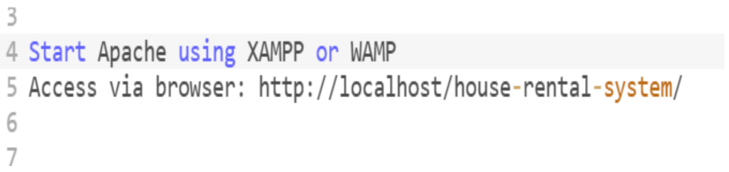


```
/admin/            → Admin-only views and controllers
/staff/            → Staff-specific interfaces
/includes/         → Database connection file, helper functions,
session handlers
/db.php            → Core database configuration
```

6. Running the Application

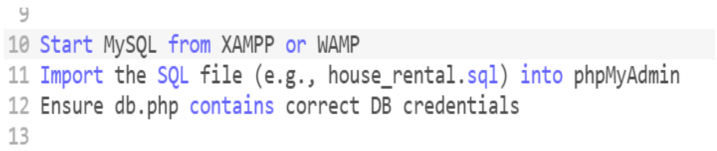
- Provide commands to start the frontend and backend servers locally

Frontend: No npm start is required. The frontend (HTML/CSS/JS/Bootstrap) is rendered by PHP templates and served via Apache. To run it:



```
Start Apache using XAMPP or WAMP
Access via browser: http://localhost/house-rental-system/
```

Backend: The backend logic (PHP scripts and database queries) runs on the server (Apache) with a MySQL database. To initialize:



```
Start MySQL from XAMPP or WAMP
Import the SQL file (e.g., house_rental.sql) into phpMyAdmin
Ensure db.php contains correct DB credentials
```

7. API Documentation

- This project uses form-based PHP submission rather than RESTful APIs, but here's a mapping of key functionalities:
    - **POST /add_tenant.php**
      Add a new tenant
    - **POST /add_payment.php**
      Record payment
    - **GET /report.php**
      Generate report
    - **GET /admin/dashboard.php**
      Admin dashboard view

8. Authentication

- Login system with sessions
- Roles:
    - **Admin:** Full privileges
    - **Staff:** Limited access (no user management)
- Credentials stored in the users table with password encryption (optional)

9. User Interface

- Bootstrap-based clean and responsive interface
- Use of Modals for edit/view actions
- Navigation bar for module switching
- Color-coded status indicators (Paid/Unpaid, Occupied/Vacant)

10. Testing

- Manual testing performed on:
  a) Form validations
  b) Payment entry
  c) Report generation
- Browser compatibility: Chrome, Firefox
- SQL Injection mitigated through input sanitization

11. Screenshots or Demo

- Provide screenshots or a link to a demo to showcase the application.

12. Known Issues

- No email/SMS alert for rent dues
- No tenant login module
- Payments cannot be updated after entry (currently only deletable)

13. Future Enhancements

- Add online rent payment integration (e.g., Razorpay)
- Introduce a tenant login portal
- Enable automated SMS/email reminders
- Mobile-responsive dashboard improvements
- improvements that could be made to the project.