

FullStack Development MERN

Project Documentation

1. Introduction

- Project Title: Home-Rentals Platform
- Team Members:
 - Siddartha(Frontend Developer)
 - Venkatesh(Backend Developer)
 - Venkatesh(Database & DevOps)
 - Eswar(QA & Testing)

2. Project Overview

- Purpose:

To provide a seamless platform for users to search, book, and manage rental properties, and for hosts to list and manage their properties.
- Features:
 - User registration and login
 - Property listing creation and editing
 - Property search and filtering
 - Booking and payment integration
 - User profile management
 - Dashboard for hosts and renters
 - Email notifications

3. Architecture

- Frontend:

Built with React.js, using functional components, React Router for navigation, Redux for state management, and SCSS for styling.
- Backend:

Node.js with Express.js, RESTful API design, JWT-based authentication, Multer for file uploads, and error handling middleware.
- Database:

MongoDB Atlas, with Mongoose schemas for users, listings, bookings, and reviews. Relationships managed via references and population

4. Setup Instructions

- Prerequisites:
 - Node.js v18+
 - MongoDB Atlas account

- npm
- Installation:
 1. Clone the repository
 2. Run npm install in both [client](#) and [server](#) directories
 3. Set up .env files for both client and server (API keys, DB URI, etc.)
 4. Start MongoDB Atlas cluster
- 5. **Folder Structure**
 - Client:
 - src/components/ (UI components)
 - src/pages/ (Page-level components)
 - src/redux/ (State management)
 - src/styles/ (SCSS files)
 - Server:
 - routes/ (API endpoints)
 - models/ (Mongoose schemas)
 - middleware/ (Auth, error handling)
 - database/ (DB connection)
 - public/uploads/ (Image storage)
- 6. **Running the Application**
 - Frontend:

```
cd client && npm start
```
 - Backend:

```
cd server && npm start
```
- 7. **API Documentation**
 - Example Endpoint:

```
POST /api/auth/register
```

 - Request: { email, password }
 - Response: { user, token }
 - See full API docs in [routes](#) for all endpoints (listings, bookings, users, etc.)
- 8. **Authentication**
 - JWT tokens for user sessions
 - Protected routes for dashboard, listing management
 - Passwords hashed with bcrypt
 - Token stored in HTTP-only cookies
- 9. **User Interface**
 - Screenshots:
 - Registration/Login page
 - Property listing page
 - Booking confirmation
 - Dashboard view

10. Testing

- Strategy:
 - Unit tests for backend (Jest, Supertest)
 - Component tests for frontend (React Testing Library)
 - Manual end-to-end testing
- Tools: Jest, React Testing Library

11. Known Issues

- Image upload fails for files >5MB
- Occasional delay in email notifications
- Payment gateway sandbox only

12. Future Enhancements

- Mobile app version
- AI-powered property recommendations
- In-app messaging between renters and hosts
- Multi-language support

