# House Rent Application Using MERN Stack

## Abstract

The House Rent Application is a full-stack web application designed to streamline   
the process of renting properties. It allows landlords to list their properties and tenants to search   
and book rental properties. The project uses MongoDB for the database, Express.js for server-side logic,   
React.js for the frontend interface, and Node.js for the backend.

## Features

1. User Authentication: Secure login and registration using JWT.  
2. Role-based Access:  
 - Landlords can add, edit, and delete property listings.  
 - Tenants can search, view, and book properties.  
3. Property Listings:  
 - View property details (images, price, location, description).  
 - Search and filter properties based on criteria.  
4. Booking System: Tenants can book properties and view their bookings.  
5. Responsive Design: Fully responsive using CSS for mobile and desktop.

## Technologies Used

- Frontend: React.js, JSX, HTML, CSS.  
- Backend: Node.js, Express.js.  
- Database: MongoDB.  
- Authentication: JSON Web Token (JWT).  
- Styling: CSS (with optional libraries like Material-UI or Bootstrap).

## Architecture

1. Frontend: React.js with JSX for dynamic UI.  
2. Backend: Node.js and Express.js for API routes.  
3. Database: MongoDB with Mongoose for schema definitions.  
4. API: RESTful APIs for communication between frontend and backend.

## Modules

1. Authentication Module:  
 - Register, Login, and Logout functionalities.  
 - Secure token-based authentication.  
  
2. Property Management Module:  
 - Add, update, and delete properties (for landlords).  
 - Search, view, and filter properties (for tenants).  
  
3. Booking Module:  
 - Book a property.  
 - View user-specific bookings.  
  
4. Admin Module (optional):  
 - Manage users and properties.

## Implementation

The implementation of the application involves dividing it into frontend, backend, and database sections.  
The frontend is developed using React.js for dynamic user interfaces. The backend handles API endpoints  
and business logic using Node.js and Express.js. MongoDB is used to manage the database schema for users,  
properties, and bookings.

## Database (MongoDB)

The application uses MongoDB for data storage. The key collections are:  
1. Users: Stores user credentials and role information.  
 - Schema: name, email, password, role (tenant/landlord).  
2. Properties: Stores property listings.  
 - Schema: title, location, price, image, createdAt.

## Deployment

1. Frontend: Deploy on Vercel or Netlify.  
2. Backend: Deploy on Heroku or Render.  
3. Database: MongoDB Atlas for cloud database hosting.

## Testing

- Frontend: Test UI interactions with React Testing Library.  
- Backend: Test APIs using Postman or Jest.

## Future Enhancements

1. Add a payment gateway for online rent payments.  
2. Implement advanced search filters (e.g., by amenities).  
3. Add real-time chat between tenants and landlords.