HOUSEHUNT

A MERN Stack Property & Authentication System

1️⃣ Title Page

Project Title:
HouseHunt – Full Stack Web Application

Technology Used:
MERN Stack (MongoDB, Express.js, React.js, Node.js)

Submitted By:
Abhishek Danda

Course:
Full Stack Web Development

Year:
2026

2️⃣ Abstract

HouseHunt is a full-stack web application developed using the MERN stack. The system provides secure user authentication and backend API integration using Node.js and Express. The application connects to MongoDB Atlas cloud database for storing user data securely. Passwords are encrypted using bcrypt, and user authentication is managed through JWT tokens.

This project demonstrates practical implementation of REST APIs, database connectivity, authentication systems, and frontend-backend integration.

3️⃣ Introduction

In modern web development, full-stack applications are widely used to build scalable and secure systems. The MERN stack is one of the most popular JavaScript-based stacks.

HouseHunt is designed to:

- Implement user authentication
- Connect frontend with backend APIs
- Store data securely in a cloud database
- Demonstrate full-stack project structure

4️⃣ Objectives

- To develop a full-stack web application
- To implement secure authentication using JWT
- To connect React frontend with Express backend
- To store user data securely in MongoDB Atlas
- To understand REST API development

5️⃣ Technologies Used

Frontend

- React.js
- JavaScript
- HTML
- CSS

Backend

- Node.js
- Express.js

Database

- MongoDB Atlas

Security

- bcryptjs (Password hashing)
- jsonwebtoken (JWT authentication)

Tools

- VS Code
- Postman
- Git

6️⃣ System Architecture

User (Browser)
↓
React Frontend (Port 3000/3001)
↓
Express Backend (Port 5000)
↓
MongoDB Atlas (Cloud Database)

7️⃣ Project Structure

```
househunt/
│
├── backend/
│   ├── models/
│   │   └── User.js
│   ├── routes/
│   │   └── auth.js
│   ├── .env
│   └── server.js
│
└── frontend/
└── React Application Files
```

8️⃣ Database Design

User Schema

Field	Type	Description
name	String	User name
email	String	Unique email
password	String	Hashed password
createdAt	Date	Auto generated

9️⃣ API Endpoints

1. Register User

Method: POST
URL:

/api/auth/register

Request Body:

```
{
 "name": "Abhi",
 "email": "abhi@test.com",
 "password": "123456"
}
```

Response:

```
{
 "message": "User registered successfully"
}
```

---

2. Login User

Method: POST
 URL:

```
/api/auth/login
```

Response:

```
{
 "message": "Login successful",
 "token": "JWT_TOKEN"
}
```

🔟 Execution Steps

Backend Setup

```
cd backend
npm install
node server.js
```

Backend runs at:

```
http://localhost:5000
```

Frontend Setup

```
cd frontend
npm install
npm start
```

Frontend runs at:

http://localhost:3000

1️⃣1️⃣ Security Implementation

- Passwords are hashed using bcrypt
- JWT tokens are generated on login
- Sensitive credentials stored in `.env` file
- MongoDB Atlas provides secure cloud storage

1️⃣2️⃣ Output Screens

(You should attach screenshots here)

- Backend running console
- MongoDB Atlas connection
- Postman register success
- Login token response
- Frontend home page

1️⃣3️⃣ Advantages

- Secure authentication
- Cloud database storage
- Scalable architecture
- Clean folder structure
- Real-world project implementation

1️⃣4️⃣ Limitations

- No password reset system
- No role-based authentication
- UI is basic

1️⃣5️⃣ Future Enhancements

- Add property listing system
- Add admin dashboard
- Add image upload feature

- Deploy to cloud (Render / Vercel)
- Add payment integration

1️⃣6️⃣ Conclusion

The HouseHunt project successfully demonstrates full-stack web development using the MERN stack. It includes secure authentication, REST API development, and cloud database integration. The project provides hands-on experience with modern web technologies and backend architecture.