

# HOUSEHUNT

Finding Your Perfect Rental Home

## 1. Introduction

- **Project Title:** HOUSEHUNT-Finding your perfect Rental Home
- **Team Members:**
  - *Amrutha Varshini*—Full Stack Developer
  - *Sindhu*— Database Administrator
  - *Mahita*— Deployment/DevOps Engineer
  - *Krupa* – Quality Analyst

## 2. Project Overview

- **Purpose:**

The purpose of HouseHunt is to streamline the real estate journey by providing a comprehensive, user-friendly platform that connects home buyers, sellers, renters, and real estate professionals. It aims to simplify the often complex and time-consuming process of searching for or listing a property, offering tools and resources that enhance the experience for all parties involved.

### Features:

- Advanced Property Search
- Interactive Property Listings
- Market Insights and Trends
- Property Alerts
- Professional Connection

## Architecture:

- **Frontend:**
  - API calls with **Axios** **React**: JavaScript library used for building the user interface.
  - **Bootstrap & react-bootstrap**: Used for responsive layout and basic styling components.
  - **Material UI (MUI)**: Provides modern UI components with built-in responsiveness.
  - **Ant Design (Antd)**: Added for rich and customizable component libraries.
  - **MDB React UI Kit**: For elegant and Material Design-inspired components.
  - **Axios**: Handles API calls to the backend server.
  - **Moment.js**: Used for parsing, validating, and formatting dates in the UI.
- **Backend:**
  - Set up **Express.js** server and created `index.js` in the backend folder.
  - Configured `.env` file with **PORT**, **MongoDB URI**, and **JWT Secret**.
  - Added **CORS** and **body-parser** for API request handling.
  - Implemented authentication using `authMiddleware.js` inside a `middleware` folder.
- **Database:**
  - **MongoDB** (NoSQL database) with **Mongoose** for schema modeling
  - Collections : `Users` ,`Courses` ,`Enrollments` ,`Reviews`
  - Relationships:
    - One-to-many(`Teacher`→`Courses`)
    - Many-to-many(`Student`↔`Courses` through `Enrollment`)

## 3. Setup Instructions

- **Prerequisites:**
  - Node.js v18+
  - MongoDB
  - Git
  - npm or yarn
- **Installation:**

```
S:\>cd House Hunt
The system cannot find the path specified.

S:\>cd House hunt

S:\House hunt> cd frontend

S:\House hunt\frontend> npm install

up to date, audited 1628 packages in 10s

252 packages are looking for funding
  run 'npm fund' for details

34 vulnerabilities (4 low, 13 moderate, 16 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.

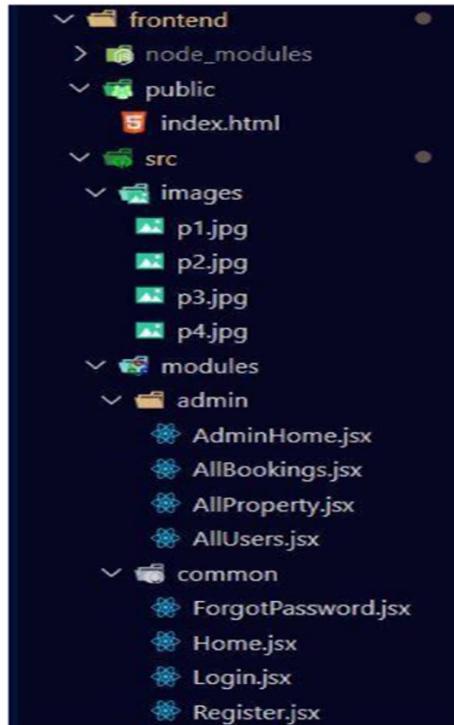
S:\House hunt\frontend>npm start

> frontend@0.1.0 start
> react-scripts start
```

S:

#### 4. Folder Structure

- Client(React):



- Server(Node.js):



## 5. Running the Application

- Frontend:

```
cd frontend
npm start
```

- Backend:

```
cd backend
npm run dev
```

## 6. API Documentation

Endpoint	Method	Description	Auth Required
/api/auth/register	POST	Register a new user (renter or owner)	✗ No
/api/auth/login	POST	Login and receive JWT token	✗ No
/api/properties	GET	Get all available rental properties	✗ No
/api/bookings	POST	Book a property (send inquiry to owner)	<input checked="" type="checkbox"/> Yes (Renter)
/api/properties	POST	Add a new property (after owner approval)	<input checked="" type="checkbox"/> Yes (Owner)

### Sample Request:

POST/api/auth/register

```
JS const User = require("../models/User"); Untitled-1 ●

1  const User = require("../models/User");
2  const bcrypt = require("bcryptjs");
3
4  exports.register = async (req, res) => {
5    const { name, email, password, type } = req.body;
6
7    try {
8      const existingUser = await User.findOne({ email });
9      if (existingUser)
10        return res.status(400).json({ message: "User already exists" });
11
12      const hashedPassword = await bcrypt.hash(password, 10);
13
14      const newUser = new User({
15        name,
16        email,
17        password: hashedPassword,
18        type
19      });
20
21      await newUser.save();
22      res.status(201).json({ message: "User registered successfully" });
23
24    } catch (err) {
25      res.status(500).json({ message: "Registration failed", error: err.message });
26    }
27  };
28
29
```

### Sample Response:

```
{
  "message": "User registered successfully",
  "token": "JWT_TOKEN_HERE"
}
```

## 7. Authentication

- **Method Used :**(JSON Web )

The screenshot shows a GitHub repository interface for the 'House\_Hunt' project. The repository name is 'House\_Hunt/b'. The current view is on the 'package.json' file. The left sidebar shows a tree view of the project's directory structure, including 'main', 'HouseHunt\_Screenshots', 'backend', 'config', 'controllers', 'middlewares', 'routes' (which contains 'adminRoutes.js', 'ownerRoutes.js', and 'userRoutes.js'), 'schemas', 'uploads', '.gitignore', 'index.js', 'package-lock.json', and 'package.json'. The right panel displays the contents of the 'package.json' file:

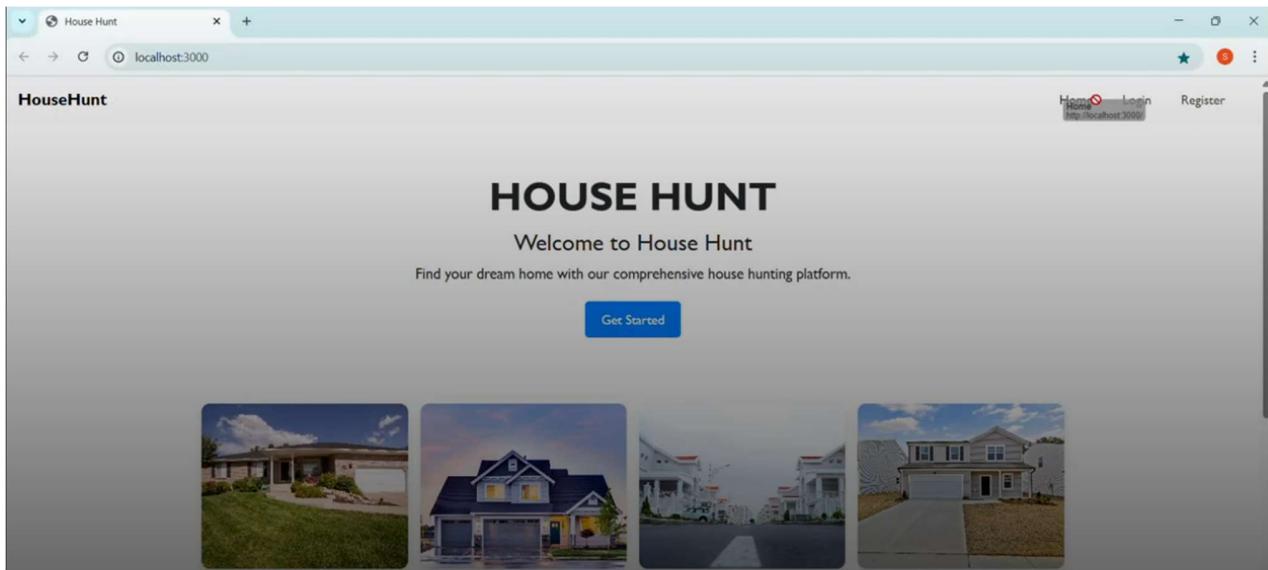
```
1  {
2    "name": "backend",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start": "nodemon index",
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "bcryptjs": "^2.4.3",
15     "cors": "^2.8.5",
16     "dotenv": "16.3.1",
17     "express": "4.18.2",
18     "jsonwebtoken": "^9.0.1",
19     "mongoose": "^7.4.3",
20     "multer": "^1.4.5-lts.1",
21     "nodemon": "^3.0.1"
22   }
23 }
```

- **Workflow:**

- **User logs in** → receives a **JWT token**, which is stored in the browser (local Storage).
- **Token is verified** on each request using middleware to protect renter, owner, and admin routes
- **Role-based actions**

## 8. User Interface

- **Property Listings Page** - View all available rental card/grid layout and images.
- **Filter/Search Properties** - Location, type, price range, and availability filters.
- **Get Info / Contact Form**



- My Bookings Dashboard - View current booking status: pending, approved, rejected.
- Profile Page - View personal info and booking history (optional)

## 9. Testing

**Tools Used:** Postman , MongoDB Compass ,Console Logs & DevTools , JWT Debugger

- **Strategy:**
  - Unit tests for backend logic
  - Integration tests for routes and middleware
  - Manual testing for frontend components
  - API testing with Postman

## 10. Screenshots or Demo

- **Demo:** Live Site
- **Screenshots:**

sindhukonny / House\_Hunt Public

Code Issues Pull requests Actions Projects Security Insights

Files

main Go to file

backend config controllers middlewares routes schemas uploads .gitignore index.js package-lock.json package.json frontend

House\_Hunt / backend / index.js

sindhukonny House Hunt Project Added

Code Blame 36 lines (23 loc) · 796 Bytes

```
1 const express = require("express");
2 const dotenv = require("dotenv");
3 const cors = require("cors");
4 const connectionofDb = require("./config/connect.js");
5 const path = require("path");
6
7 const app = express();
8
9 ///////////////dotenv config///////////////////
10 dotenv.config();
11
12 //////////connection to DB///////////////////
13 connectionofDb();
14
15
16 const PORT = process.env.PORT || 8001;
17
```

Product Solutions Resources Open Source Enterprise Pricing

Search or jump to... Sign in Sign up

sindhukonny / House\_Hunt Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

Files

main Go to file

HouseHunt\_Screenshots backend config controllers middlewares routes adminRoutes.js ownerRoutes.js userRoutes.js schemas frontend

House\_Hunt / backend / routes / adminRoutes.js

sindhukonny House Hunt Project Added 3833525 · 11 hours ago History

Code Blame 15 lines (9 loc) · 594 Bytes

Raw

```
1 const express = require("express");
2 const authMiddleware = require("../middlewares/authMiddleware");
3 const { getAllUsersController, handleStatusController, getAllPropertiesController, getAllBookingsController } = require("../controllers/admin")
4
5 const router = express.Router()
6
7 router.get('/getallusers', authMiddleware, getAllUsersController)
8
9 router.post('/handlestatus', authMiddleware, handleStatusController)
10
11 router.get('/getallproperties', authMiddleware, getAllPropertiesController)
12
13 router.get('/getallbookings', authMiddleware, getAllBookingsController)
14
15 module.exports = router
```

```

1 const express = require("express");
2 const authMiddleware = require("../middlewares/authMiddleware");
3
4 const {
5   registerController,
6   loginController,
7   forgotPasswordController,
8   authController,
9   getAllPropertiesController,
10  bookingHandleController,
11  getAllBookingsController,
12 } = require("../controllers/userController");
13
14 const router = express.Router();
15
16 router.post("/register", registerController);
17
18 router.post("/login", loginController);
19
20 router.post("/forgotpassword", forgotPasswordController);
21
22 router.get('/getallproperties', getAllPropertiesController)
23
24
25 router.post("/getuserdata", authMiddleware, authController);
26

```

## 11. Known Issues

- No Form Validation      React forms don't yet validate empty fields or email format
- Admin Role Hardcoded      Admin access is set manually; no dynamic role elevation
- Passwords in Plain Text (frontend)      Password fields are not obscured during request logs
- No Pagination      All properties load at once, not optimized for large data sets

## 12. Future Enhancements

- Two-Factor Authentication  
Improve login security with OTP or email verification
- Property Tags & Filters  
Allow search by amenities like "AC", "Parking", "Furnished"
- Mobile App (React Native)  
Launch a cross-platform mobile version

- E-Signed Lease Documents
  - Integrate digital contracts for renters and owner
- Admin Analytics
  - Graphical dashboard showing property stats, users, etc
- Cloud Storage
  - Upload property images to AWS S3 or Cloud
- Map Integration
  - Use Google Maps API to view properties on map