HOUSEHUNT

Finding Your Perfect Rental Home

1. Introduction

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

2. Project Overview

- **Purpose:**

  The purpose of House Hunt 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 callswith **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** (NoSQLdatabase) with **Mongoose** for schema modeling
  - Collections : Users ,Courses ,Enrollments ,Reviews
  - Relationships:
    - One-to-many(Teacher→Courses)
    - Many-to-many(Student↔CoursesthroughEnrollment)

3. Setup Instructions

- **Prerequisites:**
  - Node.jsv18+
  - MongoDB
  - Git
  - npmoryarn
- **Installation:**

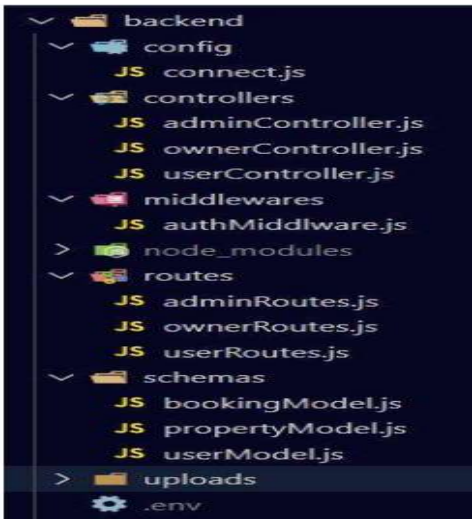
```
S:\>cd House Hund
The system cannot find the path specified.

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)	☑Yes (Renter)
`/api/properties`	`POST`	Add a new property (after owner approval)	☑Yes (Owner)

Sample Request:

POST/api/auth/register

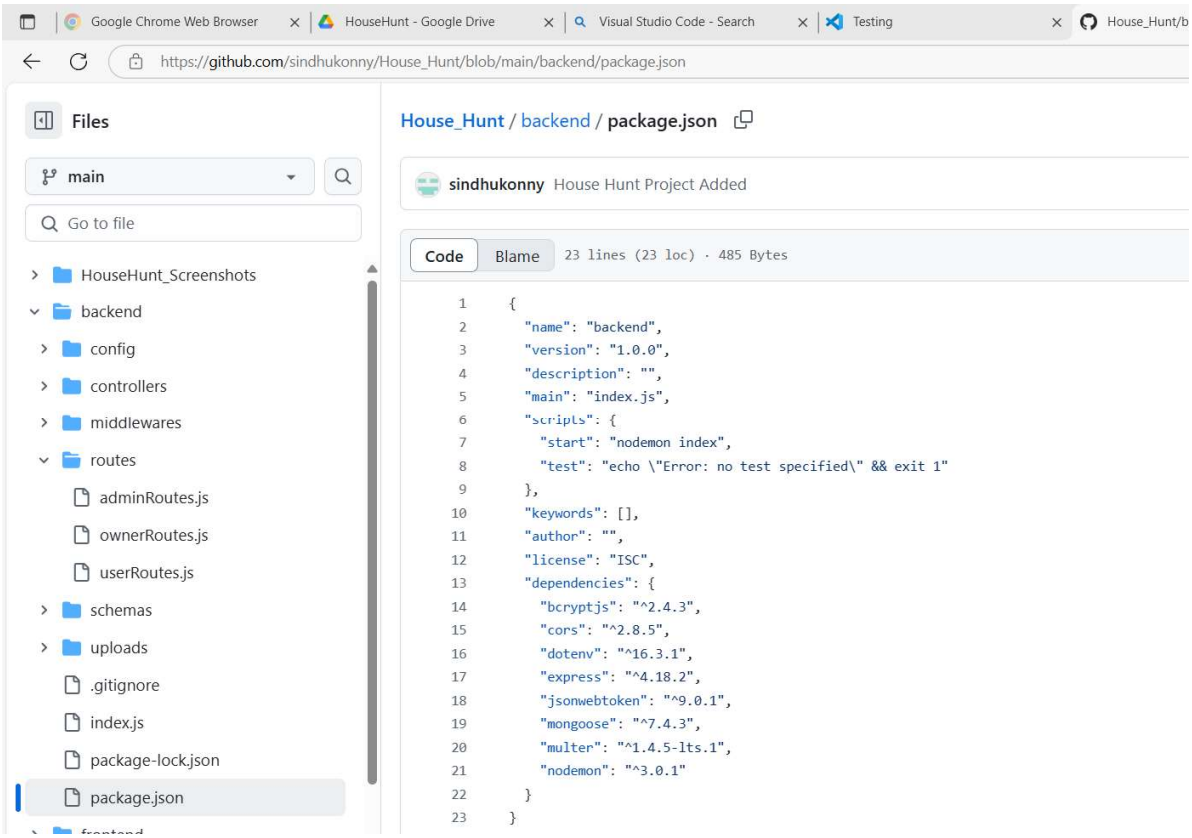
```
JS const User = require("../models/User"); Untitled-1 ●
1   const User = require("../models/User");
2   const bcrypt = require("bcryptjs");
3
4   exports.register = async (req, res) => {
5     const { name, email, password, type } = req.body;
6
7     try {
8       const existingUser = await User.findOne({ 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)



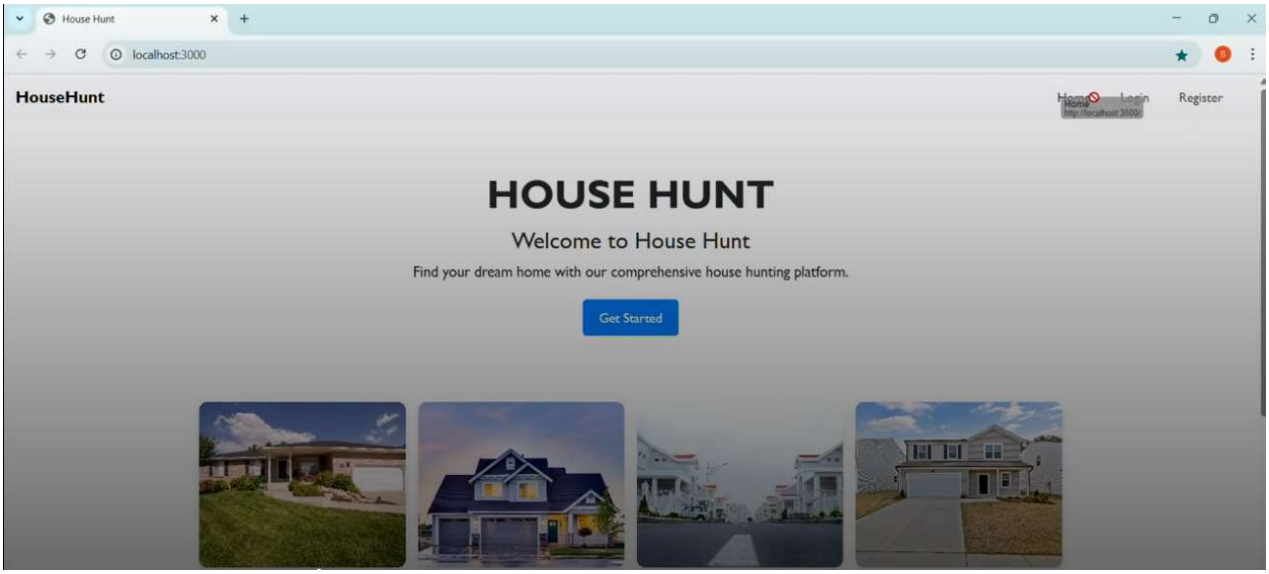
- **Workflow:**
- **User logs in** → receives a **JWT token**, which is stored in the browser (local Storage).
- **Token is verified** on each request using middleware to protect renter, owner, and admin routes
- **Role-based actions**

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)

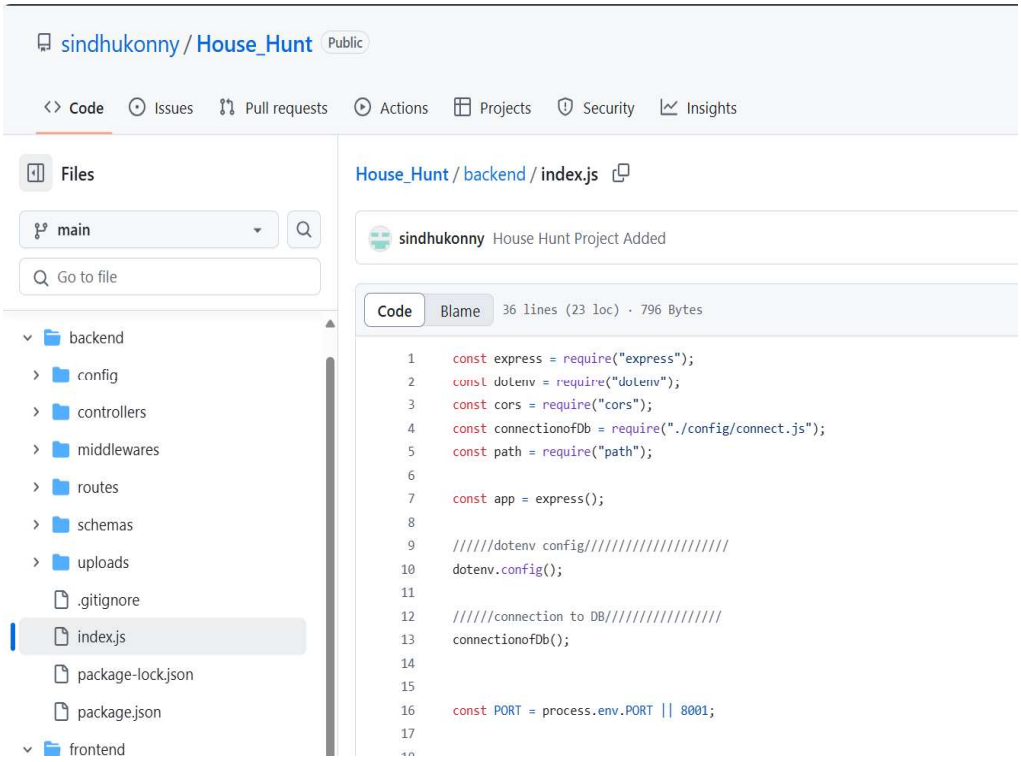
9. Testing

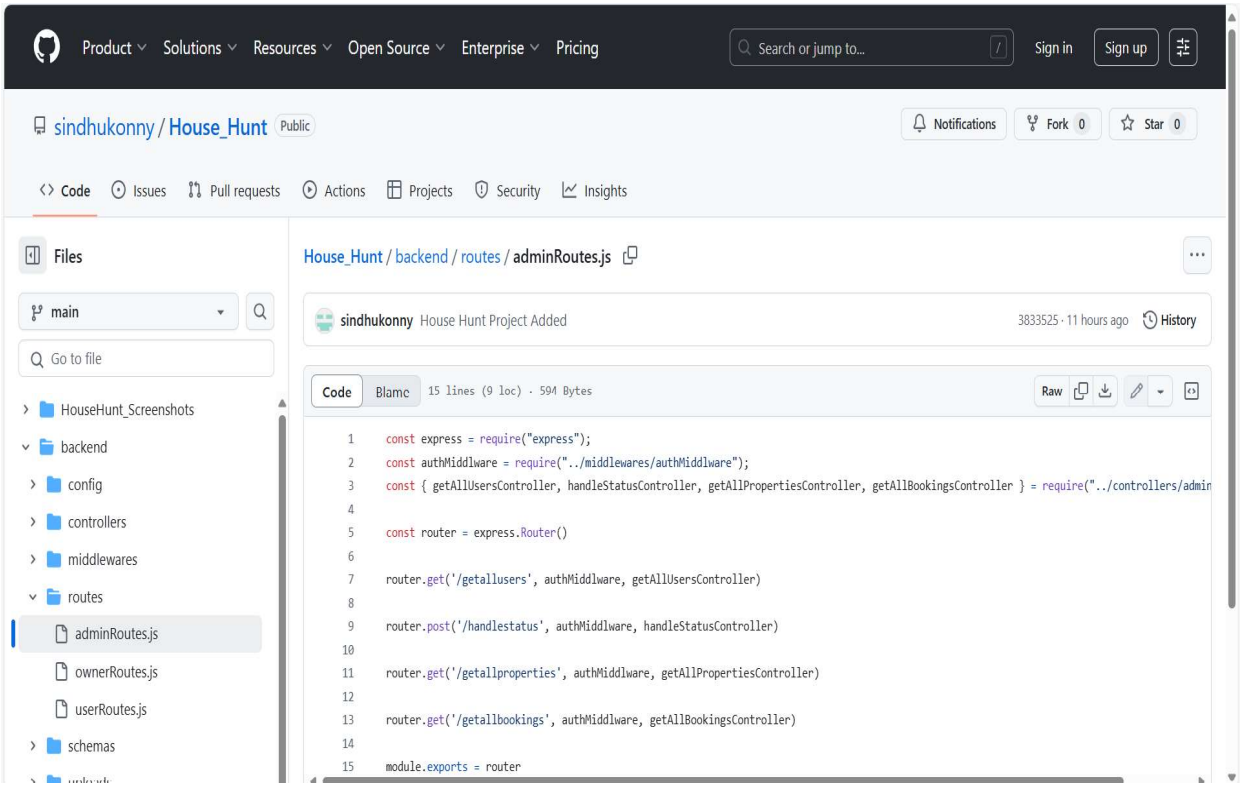
Tools Used: Postman , MongoDB Compass ,Console Logs & DevTools , JWT Debugger

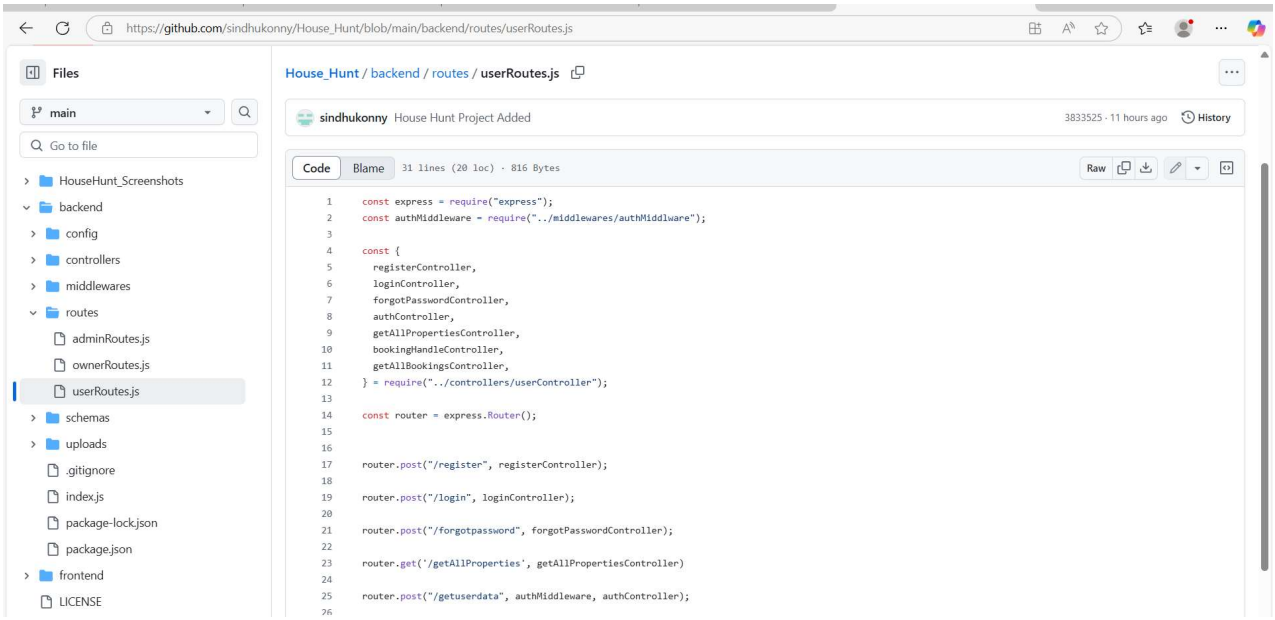
- **Strategy:**
    - Unit tests for backend logic
    - Integration tests for routes and middleware
    - Manual testing for frontend components
    - API testing with Postman

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
const express = require("express");
const dotenv = require("dotenv");
const cors = require("cors");
const connectionofDb = require("./config/connect.js");
const path = require("path");
const app = express();
//////dotenv config//////////////////////
dotenv.config();
//////connection to DB//////////////////
connectionofDb();
const PORT = process.env.PORT || 8001;


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
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
const express = require("express");
const authMiddlware = require("../middlewares/authMiddlware");
const { getAllUsersController, handleStatusController, getAllPropertiesController, getAllBookingsController } = require("../controllers/admin
const router = express.Router()
router.get('/getallusers', authMiddlware, getAllUsersController)
router.post('/handlestatus', authMiddlware, handleStatusController)
router.get('/getallproperties', authMiddlware, getAllPropertiesController)
router.get('/getallbookings', authMiddlware, getAllBookingsController)
module.exports = router



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