HOUSE RENT APP USING MERN

Introduction :

- **Project Title: House Rent Application**

- **Team Members:**
  1. **VISWANATHAN S -** Backend
  2. **ARAVINDHAN A R -** Frontend
  3. **PUSHPARAJ R -** DB
  4. **RISHI KESAVAN R -** Testing
  5. **ANBUMANI C -** Integration

Project Overview :

- **Purpose:** To create a platform for listing and renting houses where users can search, book, and manage properties.
- **Features:**
  - User authentication (login/signup).
  - Property listing with search and filter options.
  - Booking and payment integration.
  - Admin dashboard for managing properties and bookings.

Architecture :

- **Frontend**: Built with React.js for an interactive and responsive user interface.
- **Backend**: Node.js with Express.js for managing APIs and server-side logic.

- **Database**: MongoDB for storing user data, property details, and bookings.

Setup Instructions :

- **Prerequisites:**
  1. Node.js
  2. MongoDB
  3. npm or yarn package manager
- **Installation**:

  Navigate to folders:

  - Frontend
  - Backend
- **Install dependencies:**

  Run `npm install` in both .
- **Set environment variables:**

  Backend: Configure `.env` for MongoDB URI, JWT secret, etc.

Folder Structure :

- **Client**: Details for React frontend, including components, routes, and services.
- **Server**: Explanation of Node.js backend with routes, controllers, and middleware.

Running the Application :

- **Frontend:** Run `npm start` in the directory.
- **Backend:** Run `npm start` or use `nodemon index.js` in the directory.

API Documentation :

- Document key endpoints like:
    - **POST /api/auth/login** - User login.
    - **GET /api/properties** - Fetch property listings.
    - **POST /api/booking** - Book a property.

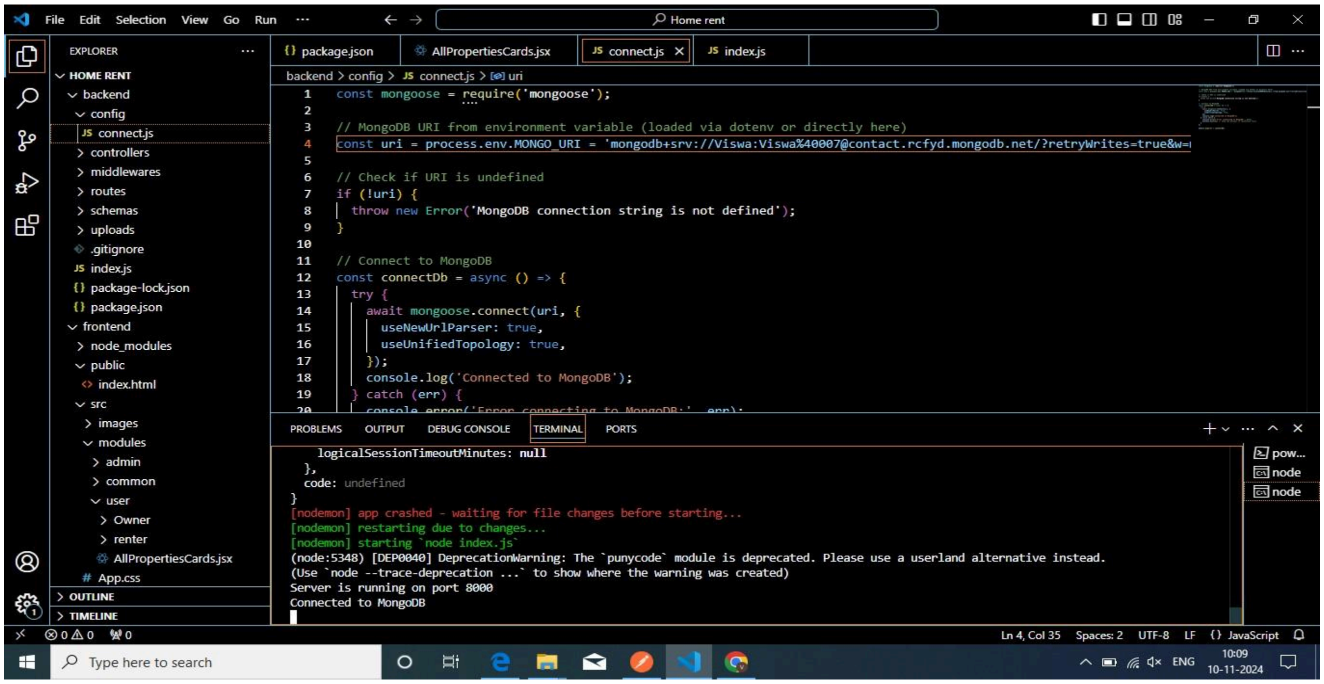
Authentication :

- Use JWT for secure user authentication.
- Include login, signup, and role-based access (e.g., admin and user).

User Interface :

- Homepage
- Property listing page
- Booking page
- Admin dashboard

MONGODB / BACKEND :

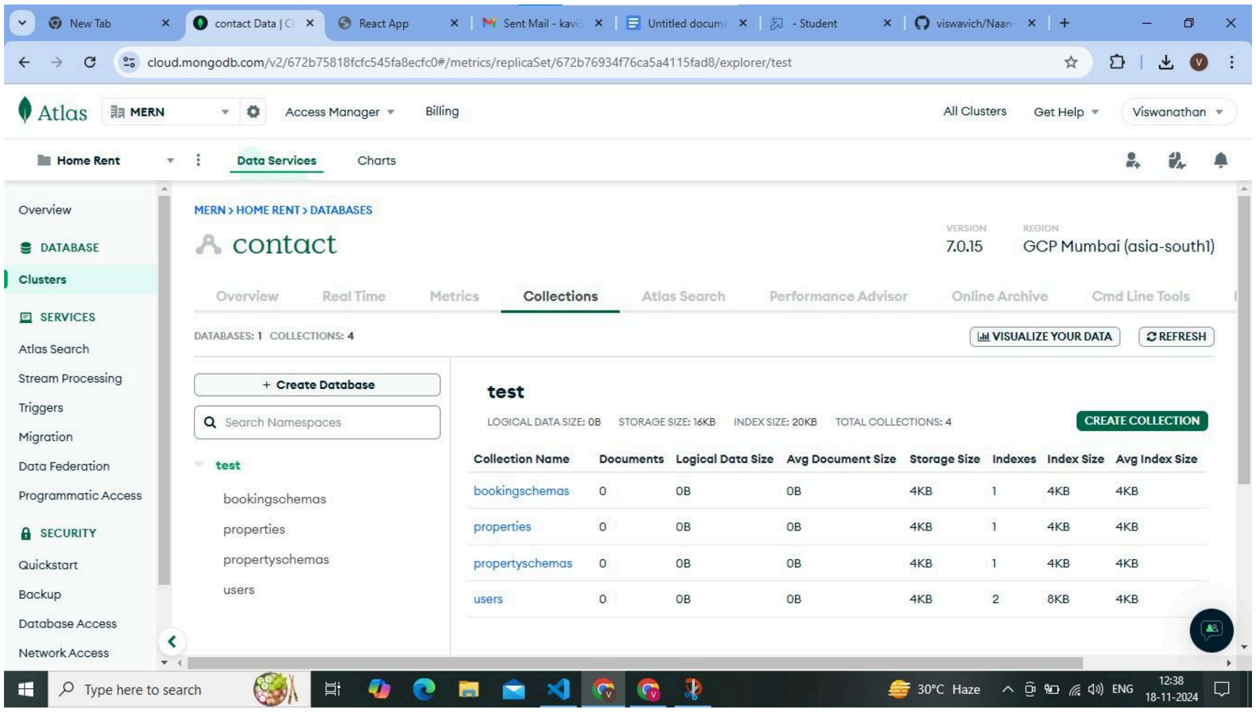


backend > config > connect.js > uri

```javascript
const mongoose = require('mongoose');

// MongoDB URI from environment variable (loaded via dotenv or directly here)
const uri = process.env.MONGO_URI = 'mongodb+srv://Viswa:Viswa%40007@contact.rcfyd.mongodb.net/?retryWrites=true&w=

// Check if URI is undefined
if (!uri) {
  throw new Error('MongoDB connection string is not defined');
}

// Connect to MongoDB
const connectDb = async () => {
  try {
    await mongoose.connect(uri, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log('Connected to MongoDB');
  } catch (err) {
    console.error('Error connecting to MongoDB:', err);
```

```
    logicalSessionTimeoutMinutes: null
  },
  code: undefined
}
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
(node:5348) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
Server is running on port 8000
Connected to MongoDB
```

Ln 4, Col 35 Spaces: 2 UTF-8 LF { } JavaScript



cloud.mongodb.com/v2/672b75818fcfc545fa8ecfc0#/metrics/replicaSet/672b76934f76ca5a4115fad8/explorer/test

MERN > HOME RENT > DATABASES

contact

VERSION 7.0.15 REGION GCP Mumbai (asia-south1)

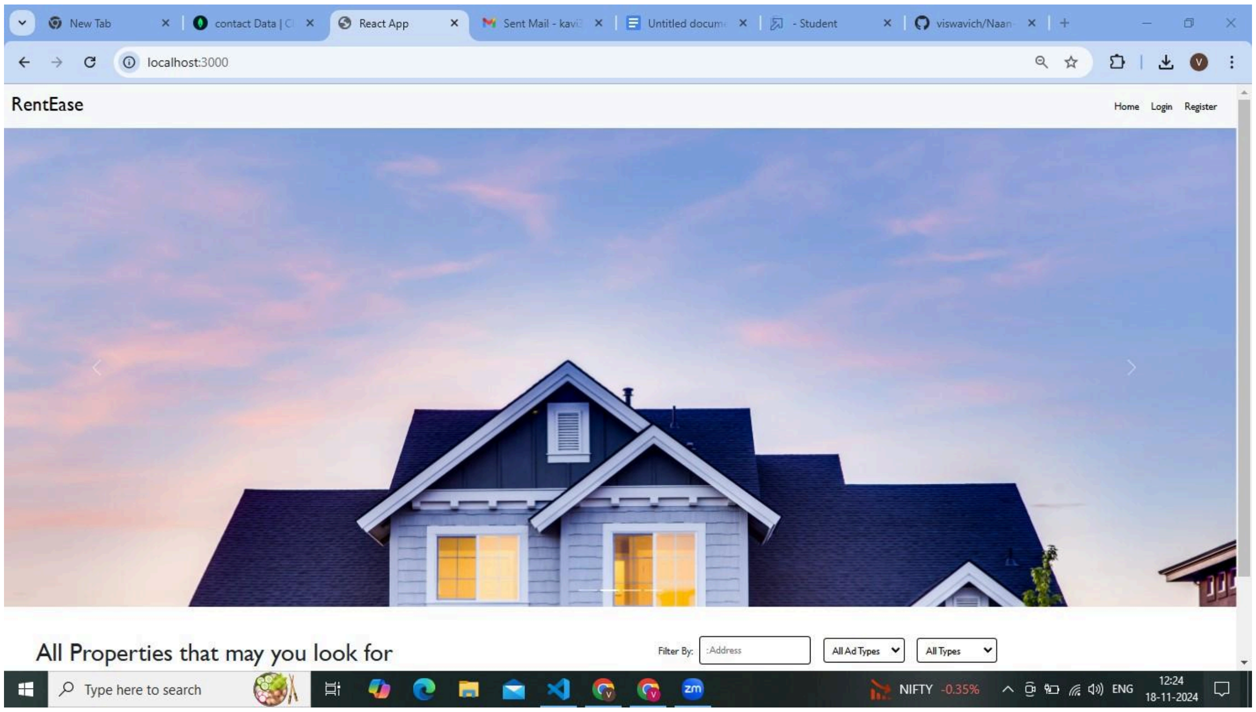
DATABASES: 1 COLLECTIONS: 4

test

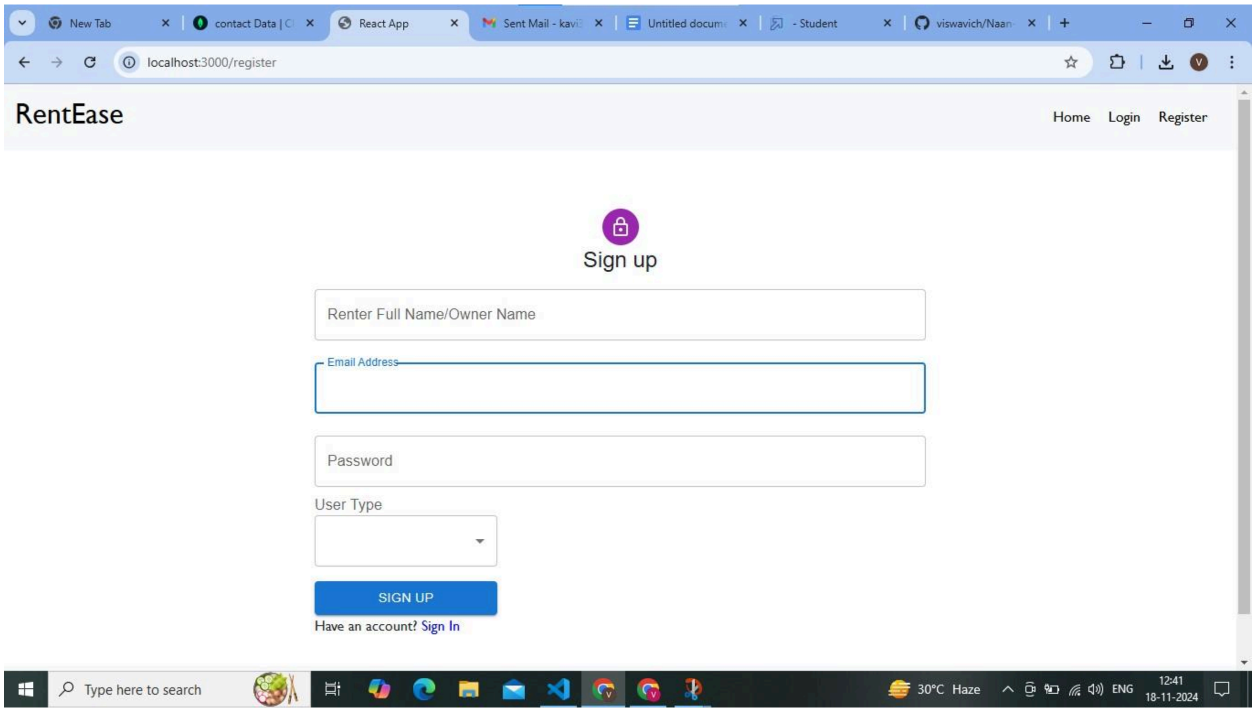
LOGICAL DATA SIZE: 0B STORAGE SIZE: 16KB INDEX SIZE: 20KB TOTAL COLLECTIONS: 4

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
bookingschemas	0	0B	0B	4KB	1	4KB	4KB
properties	0	0B	0B	4KB	1	4KB	4KB
propertyschemas	0	0B	0B	4KB	1	4KB	4KB
users	0	0B	0B	4KB	2	8KB	4KB

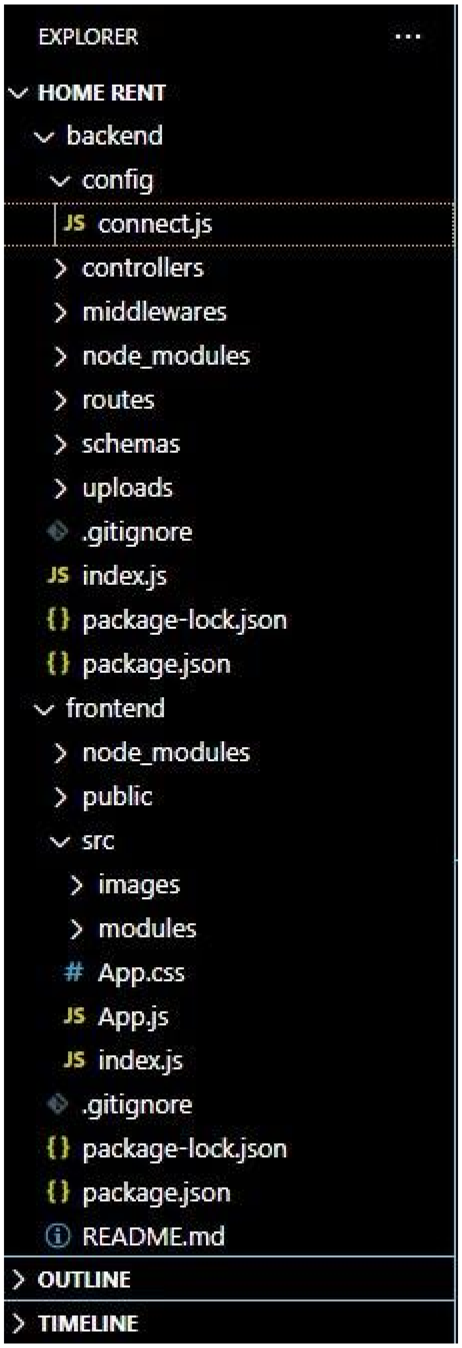
APP HOME PAGE



REGISTER



SOURCE



```
EXPLORER                ...
∨ HOME RENT
  ∨ backend
    ∨ config
      connect.js
    > controllers
    > middlewares
    > node_modules
    > routes
    > schemas
    > uploads
    .gitignore
    index.js
    package-lock.json
    package.json
  ∨ frontend
    > node_modules
    > public
    ∨ src
      > images
      > modules
      App.css
      App.js
      index.js
    .gitignore
    package-lock.json
    package.json
    README.md
> OUTLINE
> TIMELINE
```

Testing :

1. **Tools Used:**
   - **Jest: For unit testing React components and backend APIs.**
   - **React Testing Library: Simulates user interactions for frontend testing.**
   - **Supertest: Integration testing for backend routes.**
2. **Tests Performed:**
   - **Unit Tests:**
     - **Frontend: Components like `SearchBar` and `BookingForm`.**
     - **Backend: Functions like authentication and validation.**
   - **Integration Tests:**
     - **Verify frontend API calls and backend response flow.**
3. **Execution:**
   - **Run Jest: `npm test` for both frontend and backend.**

PROOF :

DEMO LINK :

https://drive.google.com/file/d/1s2krWsD7oZ3GkJqLH9wgxhTHaK10Jej_/view?usp=sharing

Known Issues :

1. **Filtering Functionality**:
   - Takes longer to load with large datasets.
   - Optimization in progress.
2. **Payment Gateway**:
   - Under development; currently uses mock data for testing.
3. **Responsive Design**:
   - Minor layout misalignment on smaller screens.
4. **API Error Handling**:
   - Limited error messages for certain failed requests.
   - Enhancements planned for better user feedback.

Future Enhancements :

- Add Google Maps integration for property locations.
- Enable multi-language support.