

RealEstateGo - Real Estate Management Platform

Rutvik B Nakum (202312017) (MSc IT - 2025 Batch)

Introduction

RealEstateGo is an advanced real estate listing platform designed to simplify property transactions. Initially developed with the MERN stack, it has been fully migrated to Go for improved performance and scalability. The platform supports high traffic loads, ensuring a seamless experience for buyers, sellers, and real estate professionals.

This documentation provides a comprehensive guide for installation, setup, deployment, database schema, and API integration.

Key Features

- **Authentication and Security:** Secure user authentication using JWT, bcrypt password hashing, and role-based access control.
- **Property Management:** Users can add, update, delete, and view property listings with images, descriptions, and pricing.
- **Advanced Search and Filtering:** Filter properties based on location, price, and type.
- **Optimized Backend:** Developed using the Go Gin framework and MongoDB.
- **Modern Frontend:** Built with React.js and styled with Tailwind CSS.

Project Structure

```
1 realestatego/  
2     api/           # Backend (Go)  
3     client/        # Frontend (React.js)  
4     .gitignore     # Git ignore file  
5     README.md      # Documentation  
6     package.json   # Node.js package file  
7     package-lock.json # Dependency lock file
```

Installation and Setup

Prerequisites

Ensure the following dependencies are installed:

- Go (latest version)
- MongoDB (Local or MongoDB Atlas)
- npm (for React frontend)

Backend Setup (Go)

```
1 git clone https://github.com/rutviknakum/Real-Estate-Management-Platform.git
2 cd realestatego/backend
3 go mod tidy
4
5 touch .env
6 # Set environment variables
7 uri := "mongodb+srv://202312017:R490J8R9ouE6YQEw@cluster0.wyik0.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"
8 go run main.go
```

Frontend Setup (React.js)

```
1 cd ../frontend
2 npm install
3 npm start
```

Database Schema

Users Collection

```
1 {
2   "_id": "ObjectId",
3   "name": "string",
4   "email": "string",
5   "password": "string",
6   "role": "string" ("user", "admin"),
7   "created_at": "timestamp"
8 }
```

Properties Collection

```
1 {
2   "_id": "ObjectId",
3   "title": "string",
4   "description": "string",
5   "price": "number",
6   "location": "string",
7   "images": ["string"],
8   "owner_id": "ObjectId",
9   "created_at": "timestamp"
10 }
```

API Endpoints

Authentication Routes

- **POST** /api/auth/register - Register a new user
- **POST** /api/auth/login - User login

Property Routes

- **GET** /api/properties - Retrieve all properties
- **POST** /api/properties - Add a new property
- **PUT** /api/properties/:id - Update a property
- **DELETE** /api/properties/:id - Delete a property

Deployment

Backend Deployment

```
1 # Ensure Go and MongoDB are installed
2 cd backend
3 export MONGO_URI=your_mongo_database_url
4 export JWT_SECRET=your_secret_key
5
6 go run main.go
```

Frontend Deployment

```
1 npm run build
2 # Deploy build folder to a web server
```

Technologies Used

- **Frontend:** React.js, Tailwind CSS
- **Backend:** Go (Gin framework)
- **Database:** MongoDB
- **Authentication:** JWT for secure login
- **Cloud Storage:** AWS S3 / Firebase Storage

License

This project is licensed under the MIT License.

Useful Links

- [Gin Framework Documentation](#)
- [MongoDB Official Site](#)
- [React.js Documentation](#)

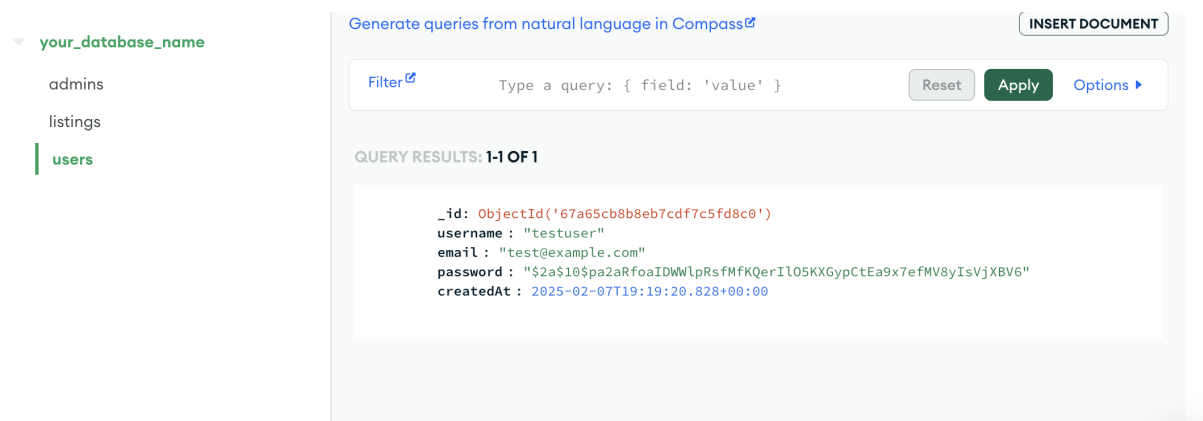


Figure 1: Add User



Figure 2: Add Admin

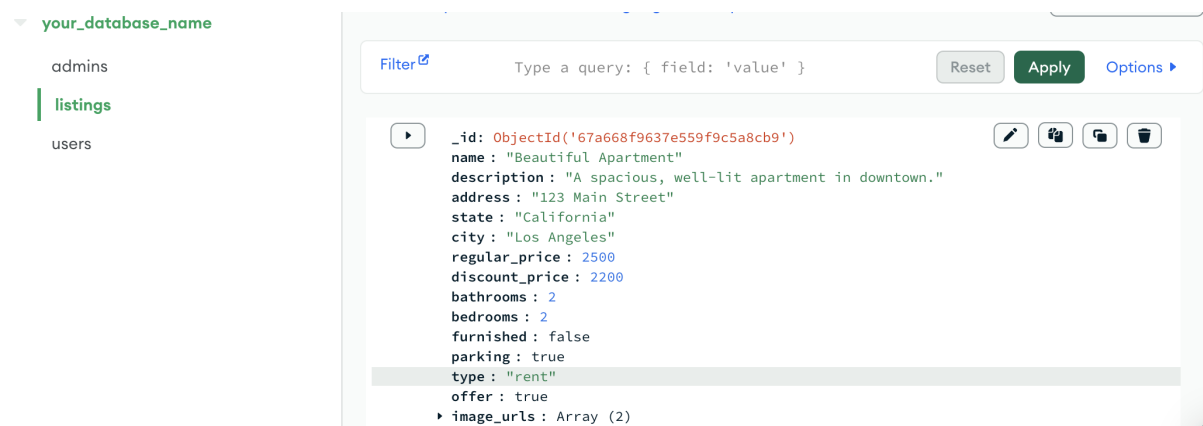


Figure 3: Add Listing