**RealNest - Real Estate Listing Platform**

**Project Requirements**

Build a full-stack Real Estate Platform that enables users to list, browse, and manage properties for sale or rent. The system supports role-based access (Admin/Customer), search functionality, listing approval, and detailed property views.

**1. Functional Requirements**

**1.1)Customer Role:**

* Register/Login with role-based access (Customer/Admin)
* Create/Edit/Delete property listings.
* Add details: Title, Description, Price, Type (Sale/Rent), Location.
* Image Upload for properties (**Cloudinary**).
* View Dashboard to manage owned listings.
* Search and Filter listings by:
  + Location
  + Price range
  + Type (Sale/Rent)
  + Keywords
* View property detail pages with full information and contact

Admin Role:

* User Management: View/Edit/Delete users.
* Approve/Delete property listings

**Database**

* Database Type: MySQL / PostgreSQL/MongoDB
* Tables/Entities:
  + User: id, name, email, password, role
  + Property: id, title, description, price, type, location, ownerld, dateListed

**Backend**

* Framework: Spring Boot (Java)
* Authentication & Authorization:
  + **Spring Security + JWT** for secure role-based access.
* REST APIs:
  + Property management, User login/register, Admin controls.
* Validation: Hibernate Validator (@Valid, @NotBlank, etc.)
* Exception Handling: Centralized using @ControllerAdvice.
* API Documentation: **OpenAPI/Swagger** UI via SpringDoc

**Frontend (Choose One)**

* Option 1 - Thymeleaf (Server-Side Rendering)
  + For forms and views: listing, dashboard, search, detail.
* Option 2 - React (Client-Side Rendering)
  + Axios for API calls, React Router, Tailwind/Bootstrap for UI.
  + Formik + Yup for form handling and validation.

**Note:** The above entities are for reference only and can be customized by the learner.

**Testing:**

* Write JUnit 5 and Mockito tests for backend services.
* Use Mockito for testing REST APIs.

How do I submit my work?

**For Code Submission:**

**Code:**

* Push your complete project code to GitHub
* Organize your code into /backend and /frontend folders (if using React).

**Deployment:**

* Option 1: Deploy Backend on Render/Railway/AWS EC2(Mandatory)
* Option 2: Frontend on Vercel/Netlify(Mandatory)
* Option 3: Record a 1-2 min walkthrough video and upload to Google Drive (public access) (Optional)