**Project Proposal – OSD Angular Project**

## **Project Description**

**What’s the idea behind the application?**

The idea is to build an MEAN stack web application that allows property owners, managers to list their properties, manage and lease them to tenants who can browse and manage their contracts and request maintenance on the site.

So on a role-based access basis, different users can do different things like manage tenants if you’re a property owner or apply to rent properties if you’re looking for housing etc.

**Who are the potential users?**

Property owners who want to list their properties for rent and manage their properties, tenants in them and maintenance requests for their properties.   
Tenants who are looking to rent – find and apply for properties and if they are already on a lease, manage their contract and make maintenance requests to the property owner/manager.

Maybe in the future agencies can use the application at an enterprise level or offering, investors can possibly do a share-based distribution of rental income, housing groups can be set up for houses in an estate…

**What are the main features of the web app?**

* Authentication & role-based access to different functions.
* A central dashboard for everyone –properties and users if they’re property owners.
* CRUD operations implemented thoroughly for users depending on their scope.
* Serverless backend with AWS Lambda functions
* Frontend deployment on Firebase hosting
* Search with filters

**How is this version different from the other version(s)?**

|  |  |  |
| --- | --- | --- |
| **Area** | **Previous Version** | **This Version** |
| **Features** | Basic CRUD, Auth | Admin tools, search, analytics, optimisations |
| **Deployment** | Localhost + simple Express server | Firebase frontend + serverless backend |
| **State Management** | Minimal | Signals and RxJS |
| **Testing** | Manual testing only | Unit tests |
| **CI/CD** | None | GitHub Actions |
| **UI** | Basic Material | Cleaner layout, accessibility enhancements |

## **Design**

### MoSCoW Hierarchy of Needs

|  |  |
| --- | --- |
| **Priority** | **Requirement** |
| **Must** | Authentication, CRUD for properties/leases/users/tenants |
| **Must** | Admin dashboard with suspend/delete functionality |
| **Must** | Firebase deployment of frontend |
| **Must** | Use Angular Material for responsive UI |
| **Should** | Integrate property valuation API |
| **Should** | Add serverless backend using AWS Lambda |
| **Should** | Provide screenshots and unit test coverage |
| **Could** | Add real-time updates (e.g. WebSockets) |
| **Could** | Add user activity analytics or usage trends |
| **Won’t** | Build a mobile app or native experience in this version |

### User Stories

When I log in as a tenant, I want to see the lease I am currently on so I am aware of my rent payment.

I should be able to manage my tenants so I can keep track of who is renting what.

I want to assign leases and manage properties as part of tracking the tenants assigned to each property.

I’d like to look for properties by both the area they are in and their prices to find what I need quickly.

Existing Work Overview  
The last version of the property management app included:

* Angular frontend (with Material)
* MongoDB database storing users and properties.
* Basic CRUD operations for users and properties.
* Express/NodeJS backend RESTful API
* An attempt at JWT tokens/authentication.

I have no screenshots of the old application because I rewrote pretty much all of it.

## Schema

* Users: firstName, lastName, phoneNumber, address, hashed auth credentials
* Properties: title, location, rental price or asking price, description, numberOfBeds, numberofBathrooms, description

## Existing Issues

* Admin capabilities were limited (no ability to suspend or delete users)
* Search and filtering of properties was only minimally implemented.
* No state management pattern was properly used
* No serverless or containerisation deployment
* Minimal testing coverage
* The design was functional but lacked polish