CS 545 - Web Application Architecture

Project Requirements - (React + Spring)

This is an engineering proof of concept. The goal is to get some hands-on experience building a full-stack project with Spring and React technologies.

**Project Topic – Mini Property Management System**

|  |  |
| --- | --- |
| **Feature** | **Done** |
| **Admin** | |
| 1. The admin should have a dashboard page | **done** |
| 1. If the owner registers to the web site, he/she need to get approval from Admin in order to post properties. | **done** |
| **Owner** | |
| 1. Register as Owner | **done** |
| 1. Property (CRUD). If a property is under ‘**pending’** it cannot be deleted. | **done** |
| 1. Owner cannot submit offers on property from the website | **Done** |
| 1. Maintain offers:    1. Reject offer if the owner does not accept. The property status should remain ‘**available**’    2. If ‘**pending**’ phase get accepted from both sides. The property status becomes ‘**contingent**’.    3. Receive messages from a customer (General inquires NOT an offer)    4. Cancel contingency    5. Maintain offers placed (list of placed offers) | **Done**  **Done**  **Done**  **Done**  **Done** |
| **Customer** | |
| 1. Register as Customer | **done** |
| 1. Cannot offer properties on this website | **done** |
| 1. Customer actions:    1. Check offer History    2. Maintain current offers placed    3. Cannot cancel offer after ‘contingency’    4. Place offer, the property status will be changed to ‘**pending**’ if the offer gets accepted    5. Send message to the property owner    6. Maintain Saved List | **Done**  **Done**  **Done**  **Done**  **Done backend**  **Done** |
| **General** |  |
| Login/Logout | **done** |
| Security with JWT (Users should not be able to access other pages links) | **done** |
| Process verifications etc. (user get email of purchase, gets a message)  Adding refresh tokens (Optional) | **Done** |
| Validation is optional. | **Done** |
| **Technical aspects** |  |
| Neat code and organization | **done** |
| Managed packages, folders, and files | **done** |
| **Project Submission** |  |
| 1. List all Team Members with Student ID 2. List all features implemented in your project. Use this table, if you implemented, check on that feature. 3. Document how to configure /Install Application – (***if your project is not using H2 database, configure it with docker****)* 4. Application should have **pre-populated data.** 5. Source code or Github links include documents or links for check-in history 6. Create a video to show functionalities about your project 7. One group submit only one 8. **Due Date: Feb 08, 2023 (Thursday) 11:59 PM.** | |