**OlivApp - Olive Oil Mills Management App**





***CMPS356 Project Phase 2 – Web UI Implementation using React, Data Management using MongoDB and Securing the application (15% of the course grade).***

|  |  |
| --- | --- |
| **Group Id:** | G? |
| **Group Members:** | Student1 full name (StudentId)  Student2 full name (StudentId)  Student3 full name (StudentId)  **Emails:** student1@student.qu.edu.qa; student2@student.qu.edu.qa; student3@student.qu.edu.qa; |

**Grading Rubric - In the *Functionality* column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality \*** | | **Quality of the implementation** | **Score** |
| **Database Implementation and Initialization** | | | | | |
| Design and implementation of the database schema to manage the data in a MongoDB database.  (job [4 pts], product [2 pts], user [2 pts] entities) | 8 |  | |  |  |
| Populate the database with the data from the json files. | 3 |  | |  |  |
| Repository implementation to read/write data from/to MongoDB:   * **getJobs [12 pts]** * getJob [3 pts] * addJob [3 pts] * updateJob [3 pts] * cancelJob [4 pts] * completeJob [8 pts] * updateInvoice [5 pts] * payInvoice [6 pts] * **rescheduleJobs [12 pts]** * **getJobsSummary [12 pts]** * getNotes [3 pts] * addNote [3 pts] * getProducts [3 pts] * getProduct [3 pts] * addProduct [3 pts] * updateProduct [3 pts] * login [5 pts] * addUser [3 pts] * updateUser [3 pts] * getUser [3 pts] | 14% |  | |  |  |
| **Implement the Web UI using React**.  Plus customize the application UI and behavior based on the user’s role. | **45%** |  | |  |  |
| Shell page having main menu and routing | 8 |  |  | |  |
| Add Customer | 5 |  |  | |  |
| Jobs List (including links based on job status) | 16 |  |  | |  |
| Add Job | 8 |  |  | |  |
| Update Job | 6 |  |  | |  |
| Cancel Job | 6 |  |  | |  |
| Complete Job (including add Invoice) | 10 |  |  | |  |
| Update Invoice | 7 |  |  | |  |
| View Invoice | 8 |  |  | |  |
| Pay Invoice | 6 |  |  | |  |
| Add Job Note | 5 |  |  | |  |
| Get Job Notes | 5 |  |  | |  |
| Jobs Summary Report | 10 |  |  | |  |
| **Authentication and Authorization** | | | | | |
| Implement authentication by offering the users the option to login using Google, Facebook or local authentication.  (Google [3.5 pts], Facebook [3.5 pts], Local login [3 pts]) | 10 |  | |  |  |
| Secure your API using JSON Web Token (JWT) and React protected routes. | 10 |  | |  |  |
| **Documentation** | | | | | |
| **\* Design documentation:**  - 5 lessons learned from Phase 1  - SPA architecture diagram  - Database schema diagram  **\* Testing documentation:** with evidence of correct implementation using snapshots illustrating the results of testing (you must use the provided template). | 10 |  | |  |  |
| **Mill Calendar (Bonus task)**  (Must be complete and working - no partial bonus mark will be given) | 10 |  | |  |  |
| **Total** | 100 |  | |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation. | 0 |  | |  |  |

**\* Possible grading for functionality**: ***Complete and*** ***Working*** (get 70% of the assigned grade), ***Complete and*** ***Not*** ***working*** (lose 40% of assigned grade) and ***Not done*** get 0. The remaining grade is assigned to the quality of the implementation. In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation. Quality **correct application of MVC**, includes meaningful naming of identifiers, no redundant code, simple and efficient design, clean code without unnecessary files/code, use of comments where necessary, proper white space and indentation. **Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers and unnecessary complex/poor user interface design.

# Design Documentation

# 5 lessons learned from Phase 1

# SPA architecture diagram

# Database schema diagram

# Web API Testing

# Get Jobs

# Get Job

# Add Job

# Update Job

# Cancel Job

# Complete Job

# Update Invoice

# Pay Invoice

# Reschedule Jobs

# Get Jobs Summary

# Get Notes

# Add Note

# Get Products

# Get Product

# Add Product

# Update Product

# Login

# Add User

# Update User

# Get User

# User Interface Testing

# Shell page having main menu and routing

# Add Customer

# Jobs List (including links based on job status)

# Add Job

# Update Job

# Cancel Job

# Complete Job (including add Invoice)

# Update Invoice

# View Invoice

# Pay Invoice

# Add Job Note

# Get Job Notes

# Jobs Summary Report

# Local Login

# Google Login

# Facebook login

# Evidence that Authorization is working

* Customer / Manager should only see the UI options that they have access to. Attempting to access an authorized client URI should return an error (e.g., add note should only accessible to the manager). After authentication, the customer should only have access to getting their jobs, cancel job, view invoice and get notes as well as register as a customer if they do not have an account.
* Attempting to access an authorized Web API URI should return an error (e.g., add note should only accessible to the manager).

# Mill Calendar (bonus task)