|  |  |
| --- | --- |
|  | **HalaqaMetrash Web App** |

Image result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran iconImage result for quran icon

***CMPS 356 Project Phase 2 – Web UI Implementation using client-side JavaScript and Data Management using MongoDB (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** |
| **1) Database Design and Implementation** | **10%** |  |  |  |
| Design and implementation of the database schema to manage HalaqaMetrash data in a MongoDB database. | 7 |  |  |  |
| Populate the database with the data from the json files. | 3 |  |  |  |
| **2) [45%] Implement the Web UI using Client-side JavaScript.**  Plus customize the application UI and behavior based on the user’s role. | **45%** |  |  |  |
| * Add parent | 9 |  |  |  |
| * Add student | 10 |  |  |  |
| * Get tasks (parent view) | 10 |  |  |  |
| * Get tasks (teacher view) | 15 |  |  |  |
| * Add Task | 10 |  |  |  |
| * Update task | 12 |  |  |  |
| * Complete task | 8 |  |  |  |
| * Delete task | 6 |  |  |  |
| * Get messages | 8 |  |  |  |
| * Halaqa Summary Report | 12 |  |  |  |
| **3) [30%] Repository implementation to read/write data from/to MongoDB** | **30%** |  |  |  |
| * Login | 10 |  |  |  |
| * Add parent | 10 |  |  |  |
| * Add student | 10 |  |  |  |
| * Get tasks | 12 |  |  |  |
| * Add Task | 10 |  |  |  |
| * Update task | 8 |  |  |  |
| * Complete task | 8 |  |  |  |
| * Delete task | 5 |  |  |  |
| * Add message | 5 |  |  |  |
| * Get messages | 8 |  |  |  |
| * Halaqa Summary Report | 14 |  |  |  |
| **4) [15%] Design and Testing Documentation**.  **\* [5 %] Design documentation:**  - 5 lessons learned from Phase 1  - Database schema diagram  **\* [10 %] Testing documentation:** using screen shots illustrating the testing of Web UI and Web API (you must use the provided template). | **15%** |  |  |  |
| **Total** | 100% |  |  |  |
| - Discussion of the project contribution of each team member [-10pts if not done] |  |  |  |  |
| - Demo and discussion [-20pts if not done] |  |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation [-100%] |  |  |  |  |

**\* Possible grading for functionality**: ***Working*** (get 70% of the assigned grade), ***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. Design quality includes **correct usage of Single Page Architecture (SPA)**, 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

## Five lessons learned from Phase 1

## Database schema diagram

# User Interface Testing

## Add parent

## Add student

## Get tasks (parent view)

## Get tasks (teacher view)

## Add Task

## Update task

## Complete task

## Delete task

## Add message

## Get messages

## Halaqa Summary Report

# Web API Testing

## Login

## Add parent

## Add student

## Get tasks

## Add Task

## Update task

## Complete task

## Delete task

## Add message

## Get messages

## Halaqa Summary Report