**CMPS 356 - Software Development of Enterprise Applications**

**Final Practical Exam – 29 April 2019 (duration 2.5h)**

|  |  |
| --- | --- |
| Student Name (Student ID) |  |

**Goal:** Build a Hifz Web App SPA Web Application using React, Node.js and MongoDB.

Download the midterm base solution from <https://cmps356s19.github.io/FinalExam-HifzApp.zip> and unzip it to your GitHub repository. Unzipping will create a new subfolder named **FinalExam-HifzApp**. Open this folder in WebStorm and implement the tasks below.

**Implementation Tasks**

1) [40 pts] **Surah Explorer**:

**1.1.** [10 pts] Update HifzRepo.**getSurahs** to use MongoDB instead of files.

Test your implementation using Postman.

**1.2.** [10 pts] Implement **addSurah** and **initSurahDB** to add the Surah data to the database. [Test your implementation using **COMPASS**, **ROBO3T** or the **Terminal** before moving to the next task]

**1.3.** [20 pts] Develop **SurahExplorer** React component to get the Surahs by type and display them. Whenever the user selects a Surah Type from the dropdown, the app should get the Surahs using the available Web API. Test the **SurahExplorer** component by adding it to the App.js component.

2) [60 pts] **Hifz Register**:

**2.1.** [10 pts] Update HifzRepo.**getHifz** and HifzRepo.**addHifz** to use MongoDB instead of files.

**2.2.** [5 pts] Implement **initHifzDB** to add the Hifz data to the database. [Test your implementation using POSTMAN before moving to the next task]

**2.3.** [20 pts] Develop **HifzList** React component to get the hifz entries and display them when the component loads.

**2.4.** [25 pts] Develop **HifzForm** React component to allow adding a Hifz entry. The first time **the component is rendered** it should fill the Surah dropdown with the surahs returned by the get Surahs Web API. When the user submits the hifz entry, the componentshould call the Add Hifz Web API to save the entered hifz object to the database.

**2.5.** [5 pts] (optional bonus question) Implement the App main menu navigation using React Router. It should behave the same way as the provided base solution. The solution should be fully implemented and working to claim the bonus. A partial solution will NOT be considered.

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

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | % | Functionality\* | Quality of the implementation |
| 1.1. Update HifzRepo.getSurahs to use MongoDB | 10 |  |  |
| * 1. Implement **addSurah** and **initSurahDB** | 10 |  |  |
| * 1. SurahExplorer React component | 10 |  |  |
| * 1. Update HifzRepo.**getHifz** and HifzRepo.**addHifz** to use MongoDB | 10 |  |  |
| * 1. Implement **initHifzDB** | 5 |  |  |
| * 1. **HifzList** React component | 20 |  |  |
| * 1. **HifzForm** React component | 25 |  |  |
| * 1. App main menu navigation using React Router | 5 |  |  |
| Provide screenshots in *testing.docx* | - |  | *[-10pts if missing]* |

**\* 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. Must submit screenshots in **Testing-GradingSheet.docx** (otherwise -10pts).

# **Testing evidence**

* 1. **Update HifzRepo.getSurahs to use MongoDB (Test using Postman)**
  2. **SurahExplorer React component (Working UI screenshot)**

**2.1. Update HifzRepo.getHifz and HifzRepo.addHifz to use MongoDB (Test using Postman)**

**2.3. HifzList React component (Working UI screenshot)**

**2.4. HifzForm React component (Working UI screenshot)**

**2.5. App main menu navigation using React Router (Working UI screenshot)**