|  |  |  |
| --- | --- | --- |
|  | | |
|  | **CMPS 350 Project Phase 2 – WebApp UI using Next.js and Data Management using Prisma**  **Conference Management System (ConfPlus)**  **(10% of the course grade)** | |
| **Group Id:** | | G T26 |
| **Group Members:** | | Ghada Al Naimi  Student2 full name (StudentId)  Student3 full name (StudentId)  **Emails:** ga1513753@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 or Not Applicable*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality\*** | **Quality of the implementation** | **Your Grade** |
| Design and implement the Data Model | 10 |  |  |  |
| Init DB: populate the database with the data from the json files in seed.js | 5 |  |  |  |
| **Complete and correct implementation of the use cases including:**  - Repository Implementation to read/write data from the database using Prisma  - Web UI using Next.js and server actions | **70%** |  |  |  |
| * Login using Custom Login and 2 Authentication Providers | 10 |  |  |  |
| * Submit paper | 15 |  |  |  |
| * Review paper | 15 |  |  |  |
| * Get conference schedule | 10 |  |  |  |
| * Edit conference schedule | 35 |  |  |  |
| * Conference Statistics Report | 15 |  |  |  |
| **Design documentation:**   * 3 technical lessons learned from your submitted solution vs. the model solution (3%) * Data Model diagram (2% * UI Design table (3%) * Caching strategy table (2%) | 10 |  |  |  |
| **Testing documentation** using screen shots illustrating the testing results.  - Discussion of the project contribution of each team member. Members should collaborate and contribute equally to the project. | 5 |  |  |  |
| **Total** | 100 |  |  |  |
| 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.

Solution quality also includes meaningful naming of identifiers (according to Android naming conventions), no redundant code, simple and efficient design, clean implementation without unnecessary files/code, use of comments where necessary, proper code formatting and indentation.

**Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.

# Application Design

# 3 technical lessons learned from your submitted solution vs. the model solution

I learned how to create fewer css files rather a css file for everything. This helped understand how to save space. Another thing I learned is html formatting, and how to make sure everything is the same by recycling the same section. Lastly I learned how to use js files and html in an efficient way.

# Data Model diagram

A screenshot of a computer

Description automatically generated with low confidence

# UI Design table

|  |  |  |
| --- | --- | --- |
| **Page** | **Components** | **Brief justification** |
| Home | Header, Session, Presentation | The header for layout and session, presentation because repeating of them |
| Login | Header | For all pages (layout) |
| Submit Paper | Header, Author | The header for layout and author because repeating |
| Schedule Editor | Header, Session, Presentation | The header for layout and session, presentation because repeating of them |
| Review Paper | Page not found | \_ |

# Caching strategy

|  |  |  |
| --- | --- | --- |
| **Page** | **Caching strategy** | **Brief justification** |
| Home | Cache-aside | Because if the cache has the data, the cache hit is faster. |
| Login | No cache here | Because login don’t need fetch data and fill some dropdowns, it's POST method |
| Submit Paper | Cache-aside | Because if the cache has the data, the cache hit is faster. |
| Schedule Editor | Cache-aside | Because if the cache has the data, the cache hit is faster. |
| Review Paper | Page not found | \_ |

# Testing

# Custom Login

# Login using 2 Authentication Providers

# Submit paper

# Review paper

# Get conference schedule

# Edit conference schedule

# Conference Statistics Report

# Discussion of the project contribution of each team member