|  |  |  |
| --- | --- | --- |
|  | **Mizān – A Workload Management Web App**  **Phase 2 – Data Access Layer using a Database (weight 10%)** | |
| **Group Id:** | |  |
| **Group Members:** | | **Emails:** |

# 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 |
| 1. Entity-Relation (ER) Diagram | | 8% |  |  |
| 1. Seed Database | | 10% |  |  |
| **2) DB Repositories implementation** | | **76%** |  |  |
|  | **Login** | 5 |  |  |
|  | **Image result for add task icon Add Assessment** | 10 |  |  |
| **https://cdn1.iconfinder.com/data/icons/MetroStation-PNG/128/MB__tasks.png Update Assessment** | 6 |  |  |
| **Image result for delete task icon Delete Assessment** | 6 |  |  |
| **Related image Get Assessments** | 15 |  |  |
| **Related image Add Course Comment (including reply to a comment)** | 14 |  |  |
| **Related imageGet Course Comments** | 8 |  |  |
| Image result for Report icon**Courses Workload Summary Report** | 12 |  |  |
| **3) Testing documentation** using screen shots illustrating the app testing.  - Discussion of the project contribution of each team member [-10pts if not done] | | 6% |  |  |
| **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 Web Dev 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.

# Entity-Relation (ER) Diagram

# Testing

For each use case, include as evidence of a working solution both a screenshot of the app UI and a **screenshot from Prisma studio of the content of database tables used**.

# Login

# Add Assessment

# Update Assessment

# Delete Assessment

# Get Assessments

# Add Course Comment (including reply to a comment)

# Get Course Comments

# Courses Workload Summary Report

# Summary of team member contributions

|  |  |
| --- | --- |
| **Member** | **Completed use cases / tasks** |
|  |  |
|  |  |
|  |  |
|  |  |