



# Mizān - A Workload Management Web App

## CMPS 350 Project Phase 2 – Data Access Layer using a Database and Components-based UI

**This phase is worth 10%. The project submission is due by 11:59pm Thursday 15 May 2025.**

You are required to refactor Mizān Web App that you have developed in phase 1 by (1) changing the Data Access Layer to use a Database and (2) Changing the UI to use Components-based UI. You will deliver the same use cases as phase 1 by either changing your phase 1 solution or the provided base solution.

### Deliverables:

- 1) Design and implement the Data Model using Prisma.
- 2) Populate the database with the data from the json files.
- 3) Implement the Repositories to offer the same functionality as phase 1 but it should read/write to a Database. The implementation should make use of the Database capabilities (e.g., all data filtering should be done by the Database and only the needed data should be retrieved, use aggregate query to get the data for the summary report).
- 4) Refactor the app UI to use Next.js pages composed and server, client components, and **server actions**. No **Do NOT Web API**. Make use of Next.js features and React hooks to enhance the user experience.
- 5) Using phase 2 template, submit your testing document (in Word format) including screenshots of conducted tests illustrating a working implementation. Fill-up the Functionality column of the grading sheet. Include an Entity Relations Diagram of your database.
- 6) Demo your implementation and answer questions about the implementation upon request.

### Important notes:

- When creating an entity instance (such as an assessment or a comment):
  - **id** should be auto assigned by the database
  - **createdBy** should be set to the current logged-in user
  - **createdDate** should be set to current date and time
- The app menu should appear in every page except the login page. It should be shared between pages and placed in a layout file.
- The same form should be used for add/update assessment.
- Assessment type should be a dropdown and it should be filled from taskType table.
- Implement the needed validation when adding an assessment such as ensuring that the number of assessments per type does not exceed the maximum number of allowed assessment per type (e.g., maximum 10 quizzes per course).
- To keep it simple you may organise your implementation in 2 repositories. The minimum methods of each repo could be as shown in the provided repository diagram.
- Push your implementation and documentation to your group GitHub repository as you make progress.
- 'Sync Assessments to Google Calendar' and 'Deploy app to Vercel' are not required.