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.

Refactor the phase 1 Mizān Web App:

- Change the repositories to use a database and Prisma.
- ~~Convert the UI to a component-based UI using Next.js.~~

Maintain all original phase 1 use cases. Use either your phase 1 solution or the provided base code as the starting point.

Deliverables:

1) **Implement Prisma Schema**: Define and generate the database schema using Prisma.
2) **Seed Database**: Populate the database with initial data from JSON files.
3) **Develop DB Repositories**: Create database-driven repositories mirroring Phase 1 functionality. Optimize data retrieval using database-level filtering and aggregation.
4) ~~**Refactor UI with Next.js**: Rebuild the UI using Next.js (Server/Client Components). Handle data operations via Server Actions (no separate API). Enhance UX with React Hooks.~~
5) **Documentation**: Deliver the Testing Document (Word, with screenshots as the provided per template), and complete the Grading Sheet Functionality column. Include an Entity Relations Diagram (ERD) of your database.
6) **Project Demo** 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.
- ‘Sync Assessments to Google Calendar’ and ‘Deploy app to Vercel’ are not required.