Web Development Phase 2

Student Management Application

1. Description of the work:

You are required to complete the implementation of your Student Management application (described in phase 1) by applying the following:

A. Store and manage the data of your application in a real database (not in JSON files or local storage) by applying the following steps:

1. Design and model your data: create a conceptual model for your app data.
2. Implement your data model using Prisma and store the data using a relational database such as Oracle, SQLite, Postgres, etc.
3. Database Initialization: Implement **seed.js** to populate the database with the data from the JSON files. The database should contain a good number of students, courses, instructors, classes, etc. This is very important because the data will be needed to make the statistics described in the second part (B).
4. Create a **Data Repository** containing all the needed functions to read/write data from the database using Prisma Client queries. To optimize the performance and minimize unnecessary data transfer, only the required data should be retrieved from the database. All data filtering, sorting, and aggregation should be performed by the database server rather than the application code. For example, if you need to select the students taking a course 'XYZ', do not retrieve all students to your application and then make the filtering manually using the code. The filtering should be expressed by the query itself and executed within the database.
5. Create **both Server Actions** and **Web APIs** using NextJS to be called from your application (of phase 1). The methods in the created Server Actions / APIs can use simply the functions of your Data repository – you need to use both Server Actions and Web APIs to demonstrate the mastery of these concepts.

B. **Create new a use-case (exclusively using NextJS and React)** that give all possible statistics about your student/course management application – It is up to you to select the statistics that are relevant to you application, but some examples might include:

- The total of students per year / course category / course, etc.
- The top 3 courses taken by the students,
- The failure rate per course / course category,
- ….

The statistics page should include **at least ten (10)** meaningful **stats** implemented using queries (you should show you queries in the report to be submitted). You must also add sufficient data to your database to give meaningful stats (e.g. at least 500 students, 50 courses, etc.)

C. There is a new functionality extension: you are required to authenticate the user before displaying the content of the statistics web page. The authentication must be done in two ways (**they must be implemented both**): using username/password, and through third-party authentication providers (e.g. GitHub, Google, Facebook, etc.)