**CMPS 350 Project Phase 2 – Report**

**Education Platform**

**(10% of the course grade)**

**The report must be submitted in Word format only**

|  |  |
| --- | --- |
| **Group Members** | Aiman Alhetari (202104619)  Abdelrahman Kotb (202205705)  Ahmed Mohamed (202003797)  **Emails:** aa2104619@qu.edu.qa; ak2205705@qu.edu.qa; am2003797@qu.edu.qa; |
| **GitHub link** | https://github.com/aimanalhetari/web-project.git |

**Grades :**

**The student fills only the “Implementation Percentage”: Please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality**\* | **Quality of the implementation** | **Grade** |
| Design and implement the Data Model. | 10 | Working 100% |  |  |
| Init DB: populate the database with the data from the json files in seed.js | 5 | Working 100% |  |  |
| Server actions, APIs and Repository Implementation to read/write data from the database | 25 | Working 100% |  |  |
| Statistics use-case with NextJS | 40 | Working 100% |  |  |
| **Documentation**  - Data Model diagram.  - UI Design with screenshots and description.  - Database queries.  - Conducted tests and evidence.  - **Contribution** of each team member [-10pts if not done] | 20 | Working 100% |  |  |
| **Total** | 100 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation. | -100 |  |  |  |

**Important remark: In case of copying and/or plagiarism or not being able to explain or answer questions about the implementation, you lose the whole grade.**

**\* Criteria for grading the functionality:**

- The functionality is working: you get 70% of the assigned grade.

- The functionality is not working: you lose 40% of assigned grade.

- The functionality is not implemented: you get 0.

- The remaining grade in all cases from above **is assigned to the quality of the implementation**,

- The grades are distributed on the various use cases, when the design/implementation is partial, you get only the grades of designed/implemented use cases.

Code quality criteria, include:

- Use of meaningful identifiers for variables and functions (e.g. using JavaScript naming conventions)

- Pages are responsive

- Clean code: simple and concise code, no redundancy

- Clean implementation without unnecessary files/code

- Use of comments where necessary

- Proper code formatting and indentation.

**You lose marks** for code duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.

**Important Remark**:

**[Grades: 100-85]:** Will be given only to **fully functional application** with **all the quality criteria cited above met** and the project has excellent **design for the various functionalities**. **The report is professional**.

**[Grades: 85-80]:** Will be given only **to functional application** **with most of all the quality criteria cited above met** and the project has good design for the various functionalities. **The report is professional**.

**[Grades: 80-75]:** 80% of the application functionalities are functional. The project respects partially the quality criteria. **The report is professional** but misses some information.

The grades are not negotiable. We expect that only a small portion (around 15%) of the class will be able to meet the criteria for the grades **[100-85]. You should work hard to and demonstrate the merits of your application to earn those grades.+**

# Description of your proposed platform

Give an overview of how your application works

Our platform is a student management system built for the CSE department at Qatar University. It supports three types of users: students, instructors, and administrators. Each user has access to specific features such as course registration, grade submission, and course validation.

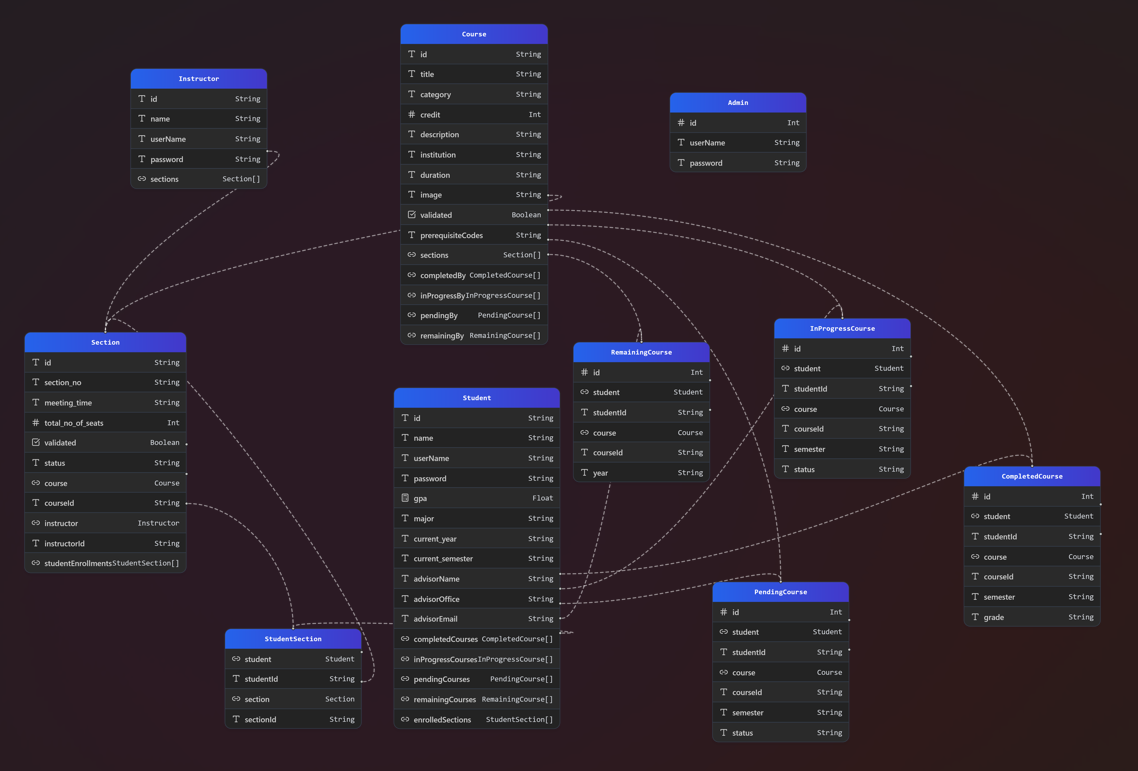
In Phase 1, we created the full interface using HTML, CSS, and JavaScript. Data was stored in JSON files. Students could log in, search for courses, register if they met the conditions, and view their academic progress. Instructors could access their classes and submit grades. Administrators were able to validate or remove courses and create new classes.

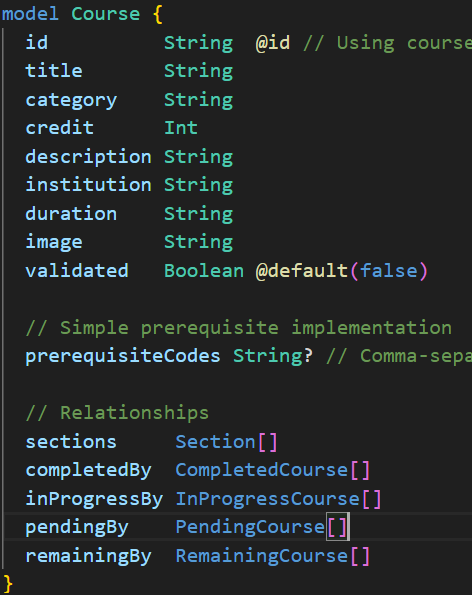
In Phase 2, we moved the data to a real relational database using Prisma ORM and SQLite. We built a repository using Prisma Client to handle data operations, with filtering and aggregation done directly in the database through queries.

We also added a new statistics page for administrator, built with Next.js and React. This page shows various academic statistics such as failure rates, GPA trends, course popularity, and instructor activity. The data is served through an API using Prisma queries.

Authentication is handled using NextAuth, with support for both GitHub and username/password login.

# Data Model





A computer screen with text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

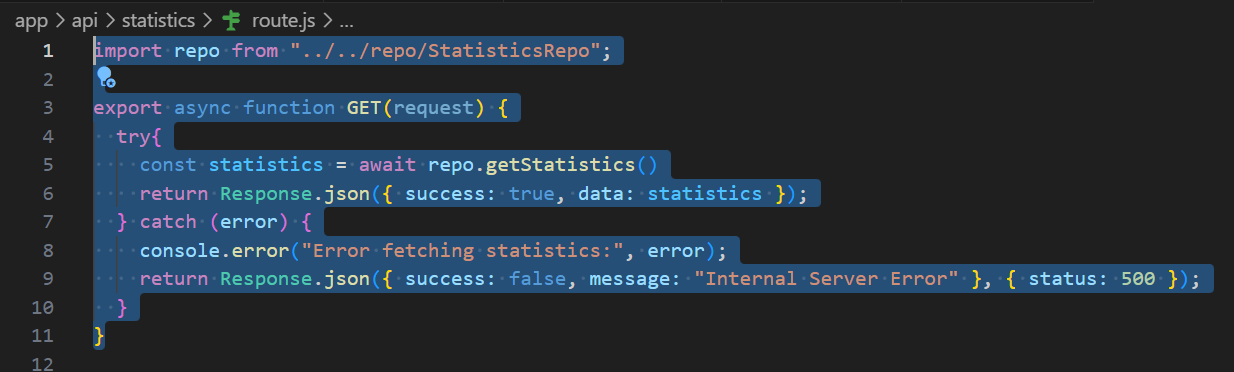
AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

# Web API, Server Actions and repository

GET /api/statistics: Calls getStatistics() from StatisticsRepo and returns the statistics in JSON.



A screenshot of a computer program

AI-generated content may be incorrect.

# Implemented statistics use case

# User Interface

We put the statistics page under the Admin role only not the student nor the instructor

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

CredentialsProvider:

username: admin1@example.com

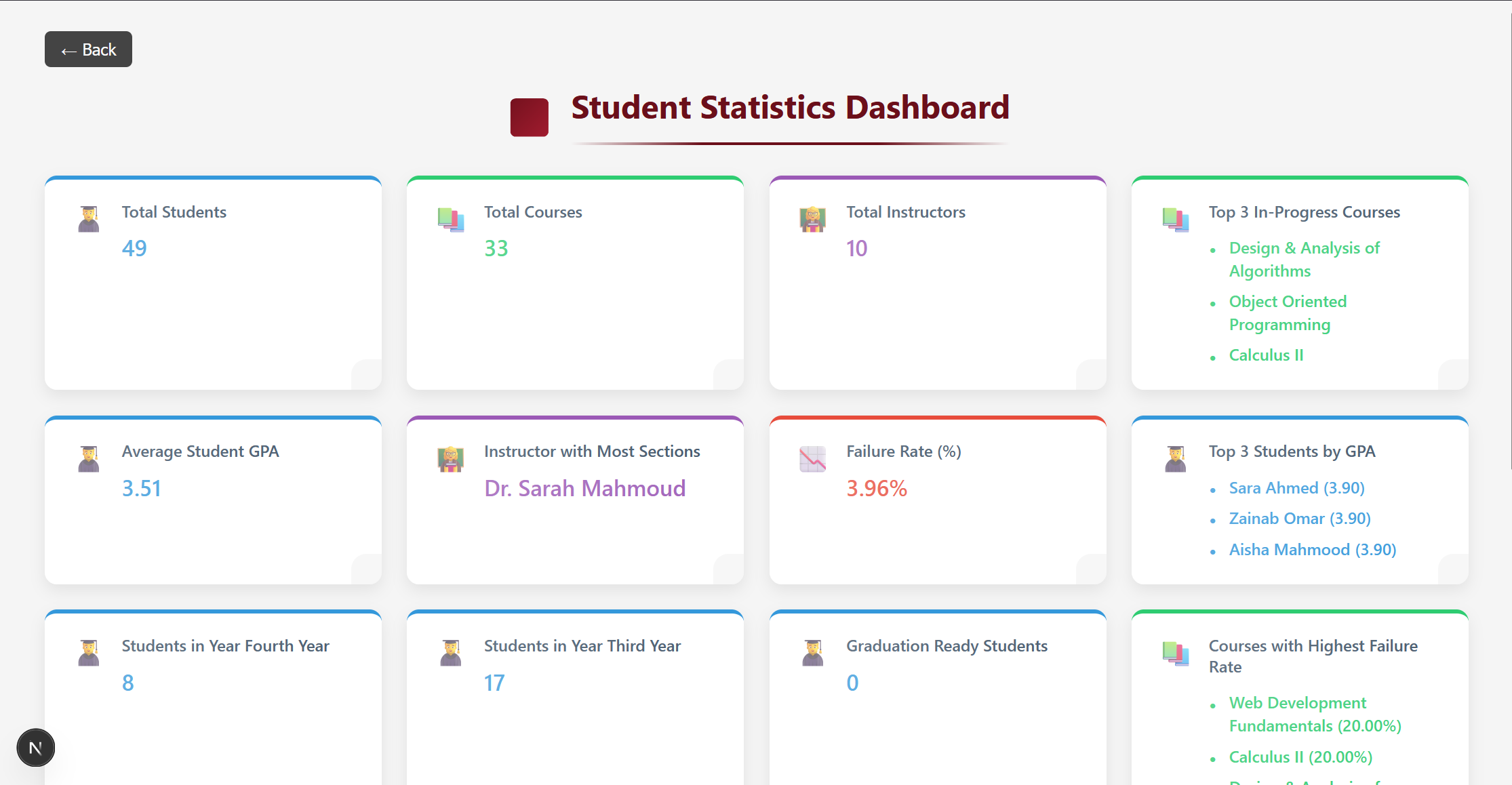
Password: admin123

A screenshot of a computer

AI-generated content may be incorrect.

GitHubProvider

The frontend dashboard displays a list of statistics in a card/grid format. Each card shows the title and value returned by the getStatistics() method.



A screenshot of a computer

AI-generated content may be incorrect.

# Implemented queries

**Implemented Queries in Repository:**

1. Total Students
2. Total Courses
3. Total Instructors
4. Top 3 In-Progress Courses
5. Average Student GPA
6. Instructor with Most Sections
7. Failure Rate
8. Top 3 Students by GPA
9. Students per year (excluding Second Year)
10. Graduation Ready Students
11. Top 3 Courses by Failure Rate
12. Validated vs Non-Validated Courses
13. Most Popular In-Progress Course
14. Average Sections per Instructor
15. Students with No Completed Courses

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with colorful text

AI-generated content may be incorrect.

# Data used in the statics

**Student Table**: for GPA, current year, remaining/completed courses

**Course Table**: for titles and validation

**Instructor Table**: for names and section count

**InProgressCourse Table**: for current enrollments

**CompletedCourse Table**: for grades and completions

**Section Table**: for instructor-course relationships

# Conducted tests

Test for Statistics API

A screenshot of a computer

AI-generated content may be incorrect.

# Discussion of the project contribution of each team member

|  |  |
| --- | --- |
| **Student name** | **Student contributions** |
| Aiman Alhetari | Authentication,statistics repo,report,api,stat page |
| Abdelrahman Kotb | Database Schema,statistics repo,stat page |
| Ahmed Mohamed | Database Schema,statistics repo,stat page |
|  |  |
|  |  |