

SSAD Assignment - 2 (*Quiz Game*)

Welcome to your second assignment for SSAD! You will be building a web based quiz game based on two frameworks - Go and React.

Introduction

You have to write a Go server that implements REST API and interacts with the database at the backend. The front end must be a React application that is an interactive quiz. Good programming practices are a must. Since this is a big application, a good structure will take you a long way in completing the assignment on time. A tutorial will be conducted soon to give you a brief idea of how your code should be structured.

Requirements (100 marks)

- Registration and login for users (10 marks)
- Multiple genre of quizzes are a must - at least 2 must be present. (10 Marks)
 - Example of genres: movies, sports, politics
 - Each genre should have at least two quizzes.
 - Each quiz must have at least 5 questions.
- At least two different types of questions (20 marks)
 - MCQ Single Correct
 - MCQ Multiple Correct
- There should be a leaderboard for the users in each genre, and an overall leaderboard across genres. (10 marks)
- Admin privileges to at least one user, along with an admin panel. Admin panel features:-(20 marks)
 - View/Create/Delete quizzes
 - Create/Delete/Edit questions/options in each quiz
 - View all users
 - Delete users
- Users must be able to see all the quizzes they have attempted along with the final score for each quiz. (10 marks)
- Design of the game (20 marks)
 - Minimalistic designs are sufficient, so long as functioning is smooth.
 - All errors must be appropriately handled at both the backend and frontend of the application.

Bonus (30 marks)

- Login through any social media (Facebook, Twitter, Google, etc) handles (10 marks)
- Different types of questions (5 marks per extra type): like identifying images, audio, etc (max 10 marks)
- Power ups or lifelines (5 marks per lifeline) (max 10 marks)

Submission

- Please follow a proper directory structure. This will be covered in the tutorial over what constitutes a good structure.
- Exact specification over submission format will be intimated on the Moodle forum of the course.

Library Usage

- Languages: Golang and React JS
- Ensure that you have a `README.md` file that lists down all the packages used in Go.
- For React, submit your `package.json` file along with the source code. **DO NOT** submit your `node_modules` folder.
- You are free to use any packages for Golang and React. However, do NOT use off-the-shelf templates. Bootstrap/CSS templates are allowed, however.
- In case of any doubts regarding whether a particular package is allowed or not, post on Moodle to get it confirmed by a TA.

Additional Notes

- Plagiarism is a strict NO. If found, you will be given 'F' grade for the course.
- Please test your application thoroughly. Any errors occurring during evaluation will be frowned upon.
- All the details of packages/templates used must be mentioned in your README file. Since Go lacks a central framework (unlike React) from which packages are installed, you **MUST** cite the link to the package as well.
- It is your application, make it as creative as you can.