Champlain College - Lennoxville
Project: Analysis

PROGRAM:	420.B0 Computer Science Technology	
COURSE:	Transactional Web Applications 2	
COURSE CODE:	420-530-LE	
WEIGHT:	5% of the final score (5% out of 15% of the project, part 1)	
SEMESTER:	Fall 2023	
INSTRUCTOR:	Francis Gauthier fgauthier@crcmail.net	Office C-239

Multiplayer Quiz App
The goal of this project is to create a multiplayer quiz app that users can play simultaneously. The project is a web-based application, that must be written with JavaScript frameworks and use WebSocket to provide real-time features.

Analysis part
First, before coding the application, a thorough analysis of the work to be done must be handed-in. The analysis must include:

1. Some mock-ups of each view of the application
2. A breakdown of each React component that will compose the application
3. A breakdown of each WebSocket event that will be used.
4. A list of the HTTP routes used in the backend if any HTTP routes are used

1. Mock-ups
For each view of the application, the team must provide a mock-up of the view of the application.
The mock-up can be made in various ways:

- Using a mock-up software
- Written on a whiteboard (you will have to take a photo)
- Written on paper

2. React components
Each view in a React application is composed of components. You must list all the components that make up the application.
For each component, you must specify:

- Its name
- If it receives *props* from another component
- The state variables needed

- The action required on component load
- The interaction possible on this component (like a button click)
- The WebSocket events it is emitting or listening on

3. WebSocket events

Next, you must list all the WebSocket events that will be used in your application. For each event, specify:

- Its name
- Which part of the application emits it
- Which part of the application is listening for that event
- The actions that must be performed when this event occurs (datastore updates, new event fired, etc.)

4. HTTP requests

If any HTTP requests are made, then a list of the HTTP requests must be provided. For each request, specify:

- The route (path and parameters)
- The HTTP method
- The payload sent
- The actions that must be performed when this request occurs

Design Requirements

Teacher point-of-view

Creation of a new room:

- Must provide the ability to create new quizzes of <u>different categories</u>
- Must be able to specify the <u>timer setting</u> for each quiz (reflection time allowed for each question). **Each question will have the same time.**
- Must be able to select the <u>difficulty level</u>
- Must be able to select the <u>number of questions</u>
- A unique alphanumeric code must be provided to share with users on quiz creation

Monitoring the quiz:

- Must be able to see the quiz progress, but cannot interfere once the quiz is started

After the quiz:

- Must be able to see the scores of all participants and the winner

Player point-of-view

Joining a new room:

- Players must join by entering the quiz code

- Players must enter their name

While the quiz is running:

- Multiple users must be able to simultaneous participate of in a single quiz session
- Real-time synchronization of questions and answers across all players' screens
- Interactive countdown timer for each question
- Intuitive user interface for answering questions
- Results are shown after each question:
    - Correct answer
    - Current player score

After the quiz:

- Must be able to see the scores of all participants and the winner

Scope

Here are features that are **not required:**

- Persistent leaderboard. The quiz results are shown once and are lost after each quiz is finished.
- Ability to interrupt a launched quiz.

Datastore vs Database

The backend server will keep the data stored about the current quiz rooms.
Reflect on what will be the data format and what data needs to be kept in the server. The data does not have to be persistent, so you can simply use JavaScript variables to hold the data. That means that the data will be wiped on server restart, but that is acceptable.

The need for a persistent database is not required here. A team can decide to use a database to keep data persistence, but it is not required to.

If a database is used, then the documents schema should be provided in the analysis.

Submission

You are free to use the software and documents of your choice to produce the analysis.

When your work is done, submit your documents in LEA. If you have any paper mock-ups, they must be handed in in class or in my office.

Submission must be made before **Wednesday September 7th, End of Day**. Late submissions are accepted with 10% penalty for any late day.