Mid-Term Exam

February 19, 2024, 6:30 PM - 8:20 PM

Be sure to read the following general instructions carefully:

- This lab test must be completed **individually** by all the students.
- Implement the solution for Exercise #1 if you are an SET student. Implement the solution for Exercise #2 if you are a G P student.
- You must submit the solution to **eCentennial** when you finish the test. You VS Code project name should be named "YourFullName_MidTermTest_Winter_2025" and should be zipped in a file "YourFullName_MidTermTest_Winter_2025.zip".

Exercise #1 – Only for Software Engineering Technology students

Create an Express.js app with GraphQL to manage Al-generated prompts and responses

Requirements:

Model Definition:

Model Name: Prompt

Properties:

- chatld: String, unique identifier for each prompt/response pair.
- prompt: String, the user's input text.
- response: String, the response generated by the Al model.
- createdAt: String, the date and time the prompt was created.
- chatTitle: String, the title of the chat.
- upVotes: Number, the number of upvotes the response has received.
- downVotes: Number, the number of downvotes the response has received.

GraphQL Schema:

- Define GraphQL types for the model.
- Implement mutations for adding a prompt and deleting a prompt.
- Define queries for fetching prompts.

Resolvers:

Implement functionality for add, list, and delete operations.

Server Configuration:

- Use MongoDB with the database name YourFullName-prompt-db.
- Set up an Express.js server that connects to MongoDB, creates an Apollo Server, and starts

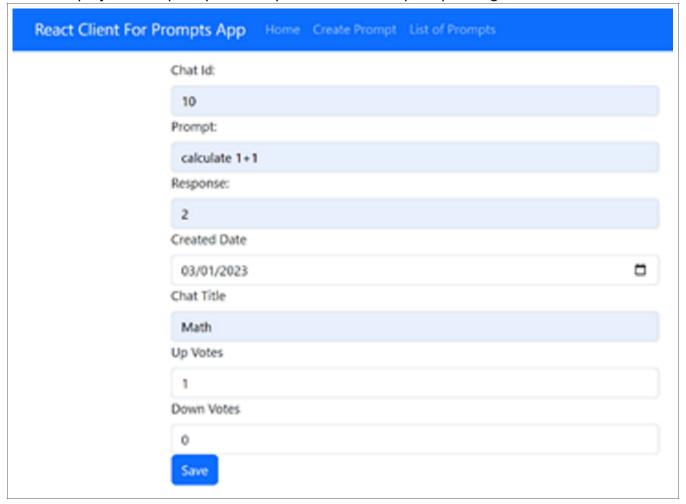
the server on port 4000.

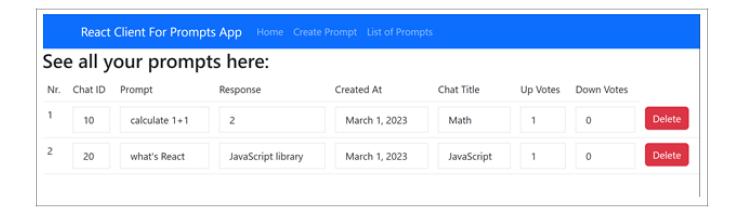
React UI:

Create a form for inserting prompt information.

Lab Test Page 1 of 5

Display a list of prompts with options to delete a prompt using the same form.





You are encouraged to reference the code from Week 5 and 6 examples or the Sample Test provided in Week 6 to guide your implementation. However, it is important to adhere to best practices by using appropriate naming conventions for models, schemas, and resolvers.

Lab Test Page 2 of 5

Additionally, ensure that all React UI components are implemented as functional components.

(100 marks)

Evaluation:

MongoDB database Configuration and Mode	15%	
Express GraphQL API:		
Add	15%	
List	15%	
Delete	15%	
React UI:		
Add prompt: React form as shown above	10%	
List prompt: React form as shown above.	10%	
Delete prompt: the same React form as List.	10%	
Friendliness (using CSS to align the React elements, React-Bootstrap, ES6+ syntax, UI design, etc.)	10%	
Total:	100%	

Exercise #2 – Only for Game – Programming students

Create an Express.js backend with GraphQL to manage movies and TV shows.

Requirements:

Model Definition:

Model Name: Movie

Properties:

- title: String, the title of the movie or TV show.
- year: Number, the release year.
- genre: String, the genre.
- description: String, a brief description.
- rating: Number, the rating.
- watched: Boolean, indicates if the movie has been watched (optional).

GraphQL Schema:

• Define types, queries, and mutations for adding, listing, and updating movies.

Resolvers:

Implement functionality for add, list, and update operations.

Server Configuration:

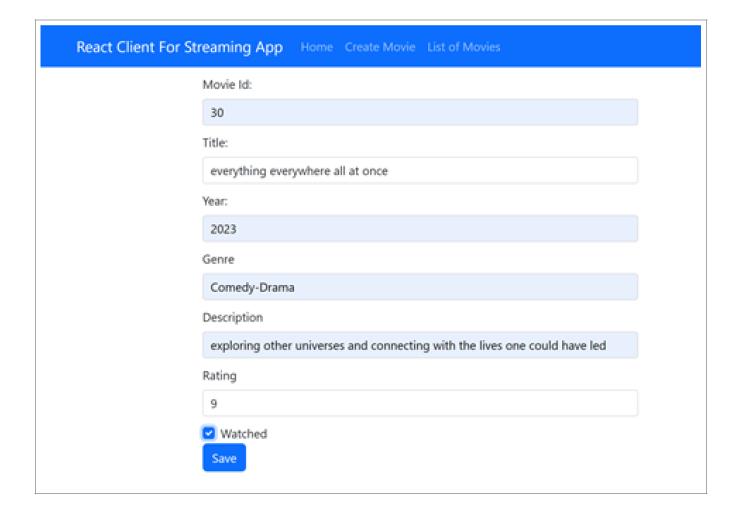
Use MongoDB with the database name YourFullName-streaming-db.

Lab Test Page 3 of 5

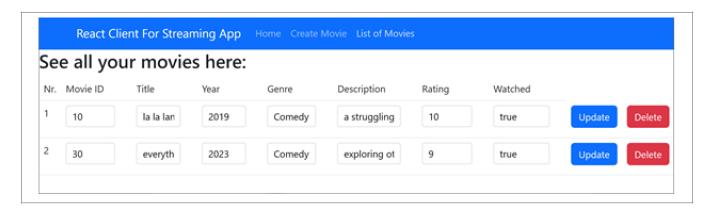
 Set up an Express.js server that connects to MongoDB, creates an Apollo Server, and starts the server on port 4000.

React UI:

- A form to add movie information.
- Display a list of movies with an option to update a movie using the same form.



Lab Test Page 4 of 5



You are encouraged to reference the code from Week 5 and 6 examples or the Sample Test provided in Week 6 to guide your implementation. However, it is important to adhere to best practices by using appropriate naming conventions for models, schemas, and resolvers. Additionally, ensure that all React UI components are implemented as functional components.

(100 marks)

Evaluation:

MongoDB Database Configuration and Model	15%	
Express GraphQL API:		
Add	15%	
List	15%	
Update	15%	
React UI:		
Add movie: React form as shown above	10%	
List movies: React form as shown above.	10%	
Update movie: the same React form as List.	10%	
Friendliness (using CSS to align the React elements, React-Bootstrap, ES6+ syntax, UI design, etc.)	10%	
Total:	100%	

Lab Test Page 5 of 5