

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. Your 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 AI-generated prompts and responses

Requirements:

Model Definition:

Model Name: Prompt

Properties:

- chatId: String, unique identifier for each prompt/response pair.
- prompt: String, the user's input text.
- response: String, the response generated by the AI 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.

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

React Client For Prompts App

HomeCreate PromptList of Prompts

Chat Id:

10

Prompt:

calculate 1+1

Response:

2

Created Date

03/01/2023

Chat Title

Math

Up Votes

1

Down Votes

0

Save

React Client For Prompts App

HomeCreate PromptList of Prompts

See all your prompts here:

Nr.	Chat ID	Prompt	Response	Created At	Chat Title	Up Votes	Down Votes	
1	10	calculate 1+1	2	March 1, 2023	Math	1	0	Delete
2	20	what's React	JavaScript library	March 1, 2023	JavaScript	1	0	Delete

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 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.

- 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.

React Client For Streaming App [Home](#) [Create Movie](#) [List of Movies](#)

Movie Id:

Title:

Year:

Genre

Description

Rating

☒ Watched

React Client For Streaming App
Home
Create Movie
List of Movies

See all your movies here:

Nr.	Movie ID	Title	Year	Genre	Description	Rating	Watched		
1	10	la la lan	2019	Comedy	a struggling	10	true	Update	Delete
2	30	everyth	2023	Comedy	exploring ot	9	true	Update	Delete

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%