

W6 PRACTICE

My SQL

Name: Vy Vicheka

IDTB: 100017


Group: 02SE

 *At the end of his practice, you should be able to...*


- ✓ Establish a **MySQL connection** on the back-end app
- ✓ Implement a **repository** layer using **MySQL queries**
- ✓ **Test the endpoints** (REST API client + front-end app)
- ✓ **Extends the project** to handle **4 tables** in the database

 *How to start?*

- ✓ Download **start code** from related MS Team assignment
- ✓ Run `npm install` on both front and back projects
- ✓ Run `npm run dev` on both front and back projects to run the client and the server

 *How to submit?*

- ✓ Submit your **code** on MS Team assignment

 *Are you lost?*

To review MySQL queries syntax

https://www.w3schools.com/mysql/mysql_sql.asp

To install My SQL Server (if needed)

<https://dev.mysql.com/doc/refman/8.4/en/windows-installation.html>

<https://dev.mysql.com/downloads/>

To connect Node back end to MySQL

https://www.w3schools.com/nodejs/nodejs_mysql.asp

<https://sidorares.github.io/node-mysql2/docs/documentation>

EXERCISE 1 – MySQL Manipulation

Before starting !

You should have a MySQL Server running. Check it out with bellow command:

```
mysql -u root -p
```

You should see MySQL monitor run properly:

```
C:\Users\PC>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 9.3.0 MySQL Community Server - GPL
```

If not, you need to install and configure MySQL server properly.

<https://dev.mysql.com/doc/refman/8.4/en/windows-installation.html>

Q1 - Create the database and the table of articles

- Open the terminal and launch MySQL monitor:

```
mysql -u root -p
```

- Create a new database (e.g. **week6Db**) using the command line
- Create a new table (articles) with the columns below:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
title	varchar(255)	YES		NULL	
content	text	YES		NULL	
journalist	varchar(100)	YES		NULL	
category	varchar(50)	YES		NULL	

Q2 - Review My SQL queries

- Complete the bellow table with the appropriate MySQL query

Use case	My SQL Query
Get all articles	SELECT * FROM articles
Get articles written by the journalist 'RONAN'	SELECT * FROM article WHERE journalist = 'RONAN';
Add an article	INSERT INTO article (title, content, journalist, category) VALUES ('Education', 'Exploring how AI transforms classrooms.', 'RONAN', 'Technology');

Delete all articles whose title starts with “R”

DELETE FROM article
WHERE title LIKE 'R%';

EXERCISE 2 – MySQL on Backend

For this exercise, you start with a start frontend and a backend code.

The goal for this exercise is to replace the provided mock repository with a MySQL repository.

Q1 - Run Frontend & Backend

Open a dedicated terminal to run the server:

```
cd back
npm i
npm run dev
```

Open a dedicated terminal to run the client:

```
cd front
npm i
npm run dev
```

Open the browser and check the front end is correctly **connected with the back end** :



The project already works as we provide fake data (mock repository).

Let's understand in detail the back and front ends.

FRONT-END

Q2 - Look at ArticleForm

How does the component know whether to create a new article or update an existing one?

The component know base on it edit property

Why is the **useParams** hook used in this component? What value does it provide when isEdit is true?

To get article ID from URL when in edit mode.

Explain what happens inside the **useEffect** hook. When does it run, and what is its purpose?

useEffect hook runs when the component mount or when ID is changes, fetching the article data if in edit mode.

Q3 - Look at the ArticleList

How are the three promise states (loading, success, and error) handled in the fetchArticles function?

- Loading: When fetching is start 'setIsLoading(true)' is called, This tell the UI to show a loading message
- Success: if articles are fetched successfully, 'setArticles(data)' saves them to state so they can be display
- Error: is something goes wrong, 'setError("Failed to load articles. Please try again.")' sets an error message, which the UI shows.






What is the role of the ArticleCard component, how does it communicate with the parent ArticleList?

ArticleCard displays one article and notifies ArticleList when the user wants to views, edit, or delete the article.
ArticleCard communicates with its parent articleList through props

BACK-END

Q4 - Why 3 layers ?

The backend is composed of the below 3 layers : routes, controllers and repository :

- ▼  src
 - ▼  controllers
 - | JS articleController.js
 - ▼  repository
 - JS mockArticleRepository.js
 - JS sqlArticleRepository.js
 - ▼  routes
 - JS articleRoutes.js
 - ▼  utils
 - JS database.js
 - JS server.js

Describe the **responsibility** of each **layer** by completing the table below:

LAYER	RESPONSABILITIES
Routes	Define API endpoint and map them to controller function
Controller	Handles HTTP request/response, call repository methods, manage business logic
Repository	Handles direct database operations and SQL queries