軟體人才培訓學院 / **Back-End Class**

Remote Learning Assignment - Week 4

Assignment 1: JavaScript Asynchronous function

Basically, there are three ways to implement the asynchronous function, which are:

1. callback
2. promise
3. async / await

A. Complete the function below to show a delayed result in the console.

```
function delayedResult(n1, n2, delayTime, callback) {
 // your code here
}
delayedResult(4, 5, 3000, function (result) {
 console.log(result);
}); // 9 (4+5) will be shown in the console after 3 seconds

delayedResult(-5, 10, 2000, function (result) {
 console.log(result);
}); // 5 (-5+10) will be shown in the console after 2 seconds
```

Hint: You should use setTimeout() for time scheduling.

B. To implement delayedResult again using **promise** this time. It should look like:

```
function delayedResultPromise(n1, n2, delayTime) {
 // your code here
}
delayedResultPromise(4, 5, 3000).then(console.log);
// 9 (4+5) will be shown in the console after 3 seconds
```

C. To implement it using async/await this time.

```
async function main() {
 // your code here, you should call delayedResultPromise here and
get the result using async/await.
}
main(); // result will be shown in the console after <delayTime>
seconds
```

Assignment 2: Callback Function and Advanced HTML DOM

Complete the functions below to make AJAX call to a URL by GET method, and show the response data on the page. You may render UI with any style.

```
function ajax(src, callback) {
 // your code here
}

function render(data) {
 // your code here
 // document.createElement() and appendChild() methods are
preferred.
}

ajax(
 'https://remote-assignment.s3.ap-northeast-1.amazonaws.com/products',
 function (response) {
   render(response);
 }
); // you should get product information in JSON format and render
data in the page
```

Reminder: You cannot connect to arbitrary URLs because of the Same-Origin Policy which is implemented by the browser. So, feel free to use my URL for testing, otherwise, you can refer to an advanced topic: CORS.

Assignment 3: Build a Website with MySQL

You already have experience in building an HTTP server by Spring Boot. Now, it's time to add a MySQL database server to your website for a simple member sign-up and sign-in feature.

1. Build a simple HTTP server by Spring Boot as usual.
2. Setup MySQL server and database schema:
    a. Install MySQL server on your computer (version 8.x is recommended).
    b. Create a database named assignment.
    c. Create a user table named user.
    d. The user table should contain at least 3 columns: id, email, and password.
    e. Column id should be a **primary key** and increase automatically.
3. Try connecting to the MySQL server from your HTTP server.
4. Provide a home page, including sign-up and sign-in forms, for people to take action.
    a. When people submit the sign-up form, you should add a new row in the user table if the same email wasn't registered before.
    b. When people submit the sign-in form, you should check if the pair of emails and passwords match any existing row in the user table.
5. If people sign-up or sign in successfully, redirect them to a member page including any simple welcome message. **Done.**
6. If people failed to sign-up or sign in, just keep them on the home page with a proper wrong message. **Done.**

Reminder:

- Using Docker or any container technique to run MySQL is not allowed for this assignment.
- Please use JDBC to create connections to the database and query data, using ORM is not allowed for this assignment.
- In this assignment, you need to hand in an additional data file in SQL format (the file name is **backup.sql**) by mysqldump tool in the Assignment-3 folder.

Assignment 4: Design Table Schema (Advanced Optional)

In assignment 3, you created a user table. Now, design an article table with columns for title and content. Also, you have to design a column to store the author of the article. The author must be one of the users of this system.
After designing this table, create it in your database assignment, it was built in assignment 3, and generated fake data, at least 30 articles.

1. Write an SQL statement to select all articles with their author's email.
2. Write another SQL statement to select articles from 7th to 12th sorted by id.

As with assignment 3, please use the mysqldump tool to export these tables to a SQL dump file (backup.sql), write your SQL statements in assignment-4.sql, then add and push them to your GitHub repository.

Reminders:

- Before taking on this challenge, it is highly recommended that you complete the Querying Relational Databases course.