Module 1 - Front-End Web Development

JavaScript – Async Operations and Promises

Overview

In this exercise, **you will work in a pair of 2 as assigned by your lecturer.**

In this exercise, you'll dive into asynchronous JavaScript programming using Promises and async/await syntax. You'll learn to handle asynchronous operations, manage concurrent tasks, and handle errors in asynchronous code. This will prepare you for real-world scenarios like API calls and data fetching.

Requirements:

// TODO: Create a Promise that simulates fetching user data

// - The Promise should resolve after 1.5 seconds

// - If userId is positive, resolve with user data object

// - If userId is negative or zero, reject with an error

// - User data should include: id, name, email, and registrationDate// TODO: Create a function that uses template literals for HTML generation

// TODO: Create a Promise that simulates fetching user posts

// - Should resolve after 1 second

// - Return an array of post objects

// - Each post should have: id, title, content, and userId

// - If userId doesn't exist, reject with error

// TODO: Create a function that chains multiple Promises together

// - First fetch user data

// - Then fetch their posts

// - Combine the data into a single object

// - Handle any errors that occur in the chain

```
// TODO: Convert the above Promise chain to use async/await
// - Use try/catch for error handling
// - Log each step of the process
// - Return combined user and posts data


// TODO: Create a function that fetches multiple users in parallel
// - Take an array of userIds
// - Fetch all users simultaneously using Promise.all
// - Handle errors for individual user fetches
// - Return array of successfully fetched users


// TODO: Create a function that fetches users and their posts in parallel
// - Fetch user data for multiple users
// - Once user data is received, fetch all their posts in parallel
// - Combine user and posts data
// - Handle errors appropriately


// TODO: Test success cases
// - Test single user fetch
// - Test multiple user fetch
// - Test error handling
```

Submission:

Save your code in a file named ***async_promises.js*** and upload it to the LMS. Demonstrate your code and the console output to your instructor.