Testleaf
Always Ahead

Classroom Assignment: Promise

Learning Objective:

- Learn how Promises can simulate tasks like fetching data, where you can use a simple condition (e.g., `const data = true`) instead of actually fetching data from a server

Expected Completion Time:

- **Best Case:** 10-15 minutes

Assignment Steps:

Create a function called `fetchDataFromDatabase()` that returns a Promise.

- Inside the function, use `setTimeout()` to simulate a delay of **3 seconds**.
- Use a simple condition like `const data = true` to simulate whether data is available.
- If `data` is `true`, **resolve** the Promise with the message `"Data fetched successfully!"`.
- If `data` is `false`, **reject** the Promise with the message `"Data not found!"`.

Call the function `fetchDataFromDatabase()` and handle the Promise:

  - Use `.then()` to print `"Data fetched successfully!"` if the Promise is resolved.
  - Use `.catch()` to print `"Data not found!"` if the Promise is rejected.

What Happens:

- The message `"Fetching data from database..."` is printed immediately.
- After 3 seconds, either:
  - `"Data fetched successfully!"` (if data is found) or
  - `"Data not found!"` (if data is not found).

Key Concepts:

- **Promise**: Simulates asynchronous tasks like fetching data.
- **.then()**: Runs when the task succeeds (data fetched).
- **.catch()**: Runs when the task fails (data not found).