

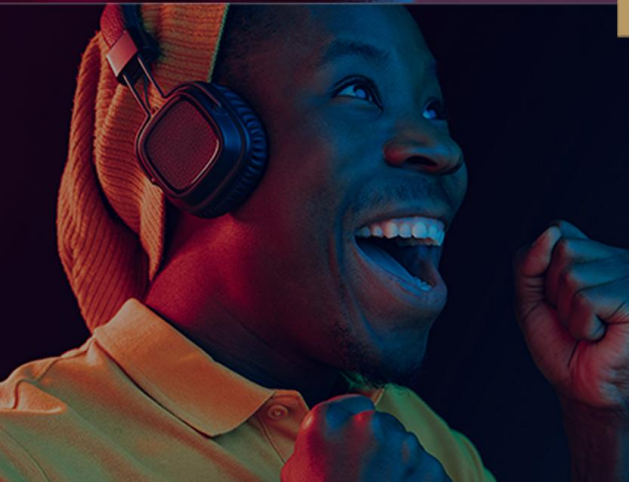


# ACADEMY OF DIGITAL ARTS EGYPT





START  
YOUR TECH JOURNEY  
WITH ADA





# JavaScript Session 6 - Student Task

## Academy of Digital Arts Egypt - JS Session 6 Task

### Objective:

By completing this task, you will:


- Master async/await syntax for handling asynchronous operations cleanly
- Create and handle Promises for custom asynchronous functions
- Make HTTP requests using Fetch API with proper error handling
- Implement GET and POST requests to communicate with external APIs
- Build a complete weather application demonstrating real-world async patterns



# JavaScript Session 6 - Student Task

## Academy of Digital Arts Egypt - JS Session 6 Task

### Step 1 – Create Promise-Based Functions


- Build a function that returns a Promise to simulate data loading
  - Create async functions using async/await syntax
  - Implement proper error handling with try/catch blocks
  - Practice sequential vs parallel operations with Promise.all()
- 



# JavaScript Session 6 - Student Task

## Academy of Digital Arts Egypt - JS Session 6 Task

### Step 2 – Implement GET Requests with Fetch API


- Use fetch() to get weather data from a public API (OpenWeatherMap or similar)
  - Handle response status codes and parse JSON data
  - Add error handling for network failures and invalid responses
  - Create reusable functions for different API endpoints
- 



# JavaScript Session 6 - Student Task

## Academy of Digital Arts Egypt - JS Session 6 Task

### Step 3 – Add POST Requests and Data Submission


- Implement POST requests to save weather preferences
  - Send JSON data in request body with proper headers
  - Handle API responses and error cases
  - Create a simple API client class with GET and POST methods
- 



# JavaScript Session 6 - Student Task

## Academy of Digital Arts Egypt - JS Session 6 Task

### Step 4 – Build Complete Weather Application

- Combine all async concepts in a weather dashboard
  - Fetch current weather and forecast data
  - Save user preferences using POST requests
  - Display loading states and handle errors gracefully
  - Use async/await throughout the application
- 

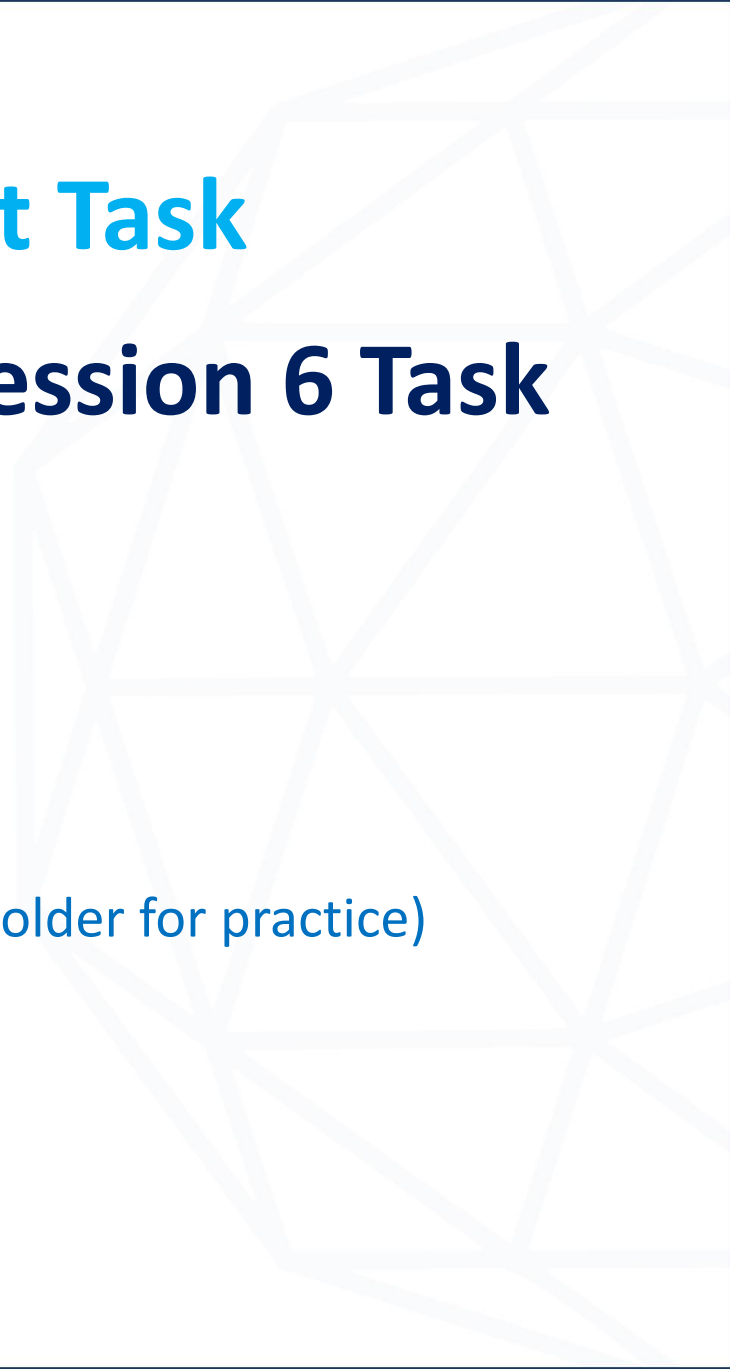


# JavaScript Session 6 - Student Task

## Academy of Digital Arts Egypt - JS Session 6 Task

### Requirements

Tools:

- Text editor (VS Code recommended)
  - Node.js installed for running JavaScript
  - Internet connection for API testing
  - Free API key from a weather service (optional, can use JSONPlaceholder for practice)
- 



# JavaScript Session 6 - Student Task

## Academy of Digital Arts Egypt - JS Session 6 Task

### Reminder

- Always use `async/await` instead of `.then()` for cleaner code
- Wrap `fetch` requests in `try/catch` blocks for error handling
- Check `response.ok` before parsing JSON to handle HTTP errors
- Use `Promise.all()` for parallel operations to improve performance
- Include proper error messages for user-friendly feedback
- Test with both valid and invalid API calls to ensure robust error handling

THANK YOU

ADAEGY

