

# JavaScript Capstone Task – Course Management System (All 3 Days Concepts)

## Scenario

You are building a Course Management System for an academy. The system should manage courses, instructors, protect data, handle user events, and simulate async server operations using modern JavaScript features.

## Day 1 Requirements (OOP + Set + Map + Proxy + Mixins)

- Create a Class called Course with properties: name and duration.
- Create a Mixin called startCourseMixin that adds a startCourse() method.
- Use a Set to store all created courses (no duplicates).
- Use a Map to map each course to its instructor.
- Use a Proxy to prevent changing the course name after creation and validate that duration is positive.

## Day 2 Requirements (Call Stack + Event Queue + Events + Callback Hell)

- Create a button 'Add Course' (Event).
- When clicked: validate inputs synchronously (Call Stack).
- Simulate saving the course using setTimeout (Async – Event Queue).
- After saving, log 'Course saved!'.
- Then simulate assigning an instructor using another async task inside the first one (Callback Hell).

## Day 3 Requirements (Promises + Async/Await + Fetch + Modules + Generator + Observable Concept)

- Refactor Callback Hell using Promises, then using Async/Await.
- Use Fetch to get instructors from: <https://jsonplaceholder.typicode.com/users> and randomly assign one.
- Split the project into modules: course.js, storage.js, api.js, main.js.
- Create a Generator to generate unique course IDs.
- Simulate observing button clicks and log each click (Observable concept).

## What Students Must Deliver

- A working multi-file JavaScript project using modules.
- Proper usage of Class, Mixin, Set, Map, Proxy, Events, Promises, Async/Await, Fetch, Generator.
- Console output showing correct execution order.
- A diagram explaining Call Stack and Event Queue during execution.

## Expected Console Flow Example

Button clicked

Validating inputs...

Course saved!

Fetching instructors...

Instructor assigned: [Name]

Course started!