JULY 2025

Full stack development

Gateshead College

Amarjit

Assignments

Assignment 01 - Create a Record format in JavaScript.

Assignment 02 - Creating a working Database/Record System.

Github repo

https://github.com/hey-amo/SkillsBootcamp-GC/

Assignment 01 - Record format in JavaScript

Using a web-based technology: Express/Node/EJS create information relating to your favourite tv/movie/radio shows (you choose) or your own topic within a record format.

You will need to add some additional information like: Title, seasons, episodes, year, cast, type,rating, poster, and directors etc..

Display the data/record on web browser by using either node, express or EJS, remember you need to show an images & code listing to show it is working.

Assignment 01 - Record

Batman Begins (2005)

- **Type:** Movie
- **Director:** Christopher Nolan
- **IMDb Rating:** 8.2
- **Cast:**
  - Christian Bale
  - Michael Caine
  - Liam Neeson
  - Katie Holmes
  - Gary Oldman

```
// Assignment 01
// Note: Requires Node/Express, EJS
//
const express = require("express");
const app = express();
const port = 3000;

app.set('view engine', 'ejs');
app.use(express.json());
app.use(express.static('public'));

// Parse JSON requests
app.use(express.json());

// Movie Record
const movie = {
    title: "Batman Begins",
    year: 2005,
    type: "Movie",
    seasons: null,
    episodes: null,
    director: "Christopher Nolan",
    imdbRating: "8.2",
    poster: "https://m.media-amazon.com/ima
    cast: [
        "Christian Bale",
        "Michael Caine",
        "Liam Neeson",
        "Katie Holmes",
        "Gary Oldman"
    ]
};

// Routes
app.get('/', (req, res) => {
    res.render('index', { movie });
});

app.use((req, res) => {
  res.status(404).send(`
    <h1>404 Not Found</h1>
    <p>The requested page <strong>${req.ori
  `);
});

app.listen(port, () => {
```

Sample code

- Record in JS
- GET Routes in Express
- Renders HTML via a ejs template

Github link:

https://github.com/hey-amo/SkillsBootcamp-GC/tree/main/assignment/assignment01

Assignment 01 - Route & View

```
app.get('/', (req, res) => {
    res.render('index', { movie });
});
```

Above is 1 example of a GET request in Express

On the right is the EJS template used to output HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1"
    >
    <title><%= movie.title %> - Info</title>
    <link rel="stylesheet" href="/style.css">
</head>
<body>
    <h1><%= movie.title %> (<%= movie.year %>)</h1>
    <img src="<%= movie.poster %>" alt="Poster for <%= movie.title %>">
    <ul>
        <li><strong>Type:</strong> <%= movie.type %></li>
        <li><strong>Director:</strong> <%= movie.director %></li>
        <li><strong>IMDb Rating:</strong> <%= movie.imdbRating %></li>
        <% if (movie.seasons) { %><li><strong>Seasons:</strong> <%=
        movie.seasons %></li><% } %>
        <% if (movie.episodes) { %><li><strong>Episodes:</strong> <%=
        movie.episodes %></li><% } %>
        <li><strong>Cast:</strong>
            <ul>
                <% movie.cast.forEach(person => { %>
                    <li><%= person %></li>
                <% }); %>
            </ul>
        </li>
    </ul>
</body>
</html>
```

Assignment 02 - Database/Record System

On this Task, you must create a working database record system, using any web-technology you feel comfortable with.

Main features, will include *adding*, *removing*, *deleting*, and *searching* (CRUD). The main point of this task is to demonstrate your capabilities of being a Full Stack Developer

Assignment 02 - My notes/goals

To create a simple CRUD app that adds, edits, deletes, lists **movie** records

To install mongodb locally via homebrew

Used mongosh mongodb://localhost:27017 to run mongo locally

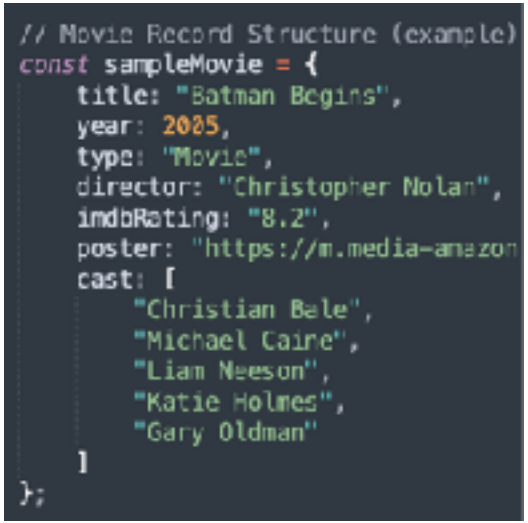
To use EJS for templating HTML

Try to use a simple callbacks/promise to simplify syntax

Use a sample record structure for quick insertion testing

Github link: https://github.com/hey-amo/SkillsBootcamp-GC/tree/main/assignment/assignment02

Assignment 02 - Sample record structure



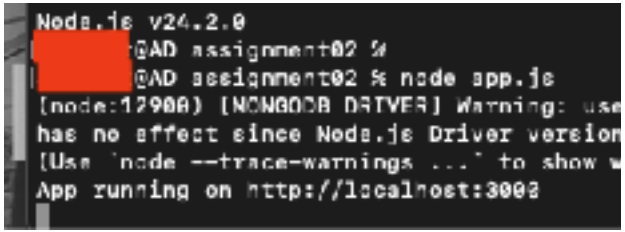
```
// Movie Record Structure (example)
const sampleMovie = {
    title: "Batman Begins",
    year: 2005,
    type: "Movie",
    director: "Christopher Nolan",
    imdbRating: "8.2",
    poster: "https://m.media-amazon
    cast: [
        "Christian Bale",
        "Michael Caine",
        "Liam Neeson",
        "Katie Holmes",
        "Gary Oldman"
    ]
};
```

This is a JSON object which holds a sample movie.

The app will allow the user to add, edit, delete, list movies using this structure

Assignment 02 - Screenshots /1

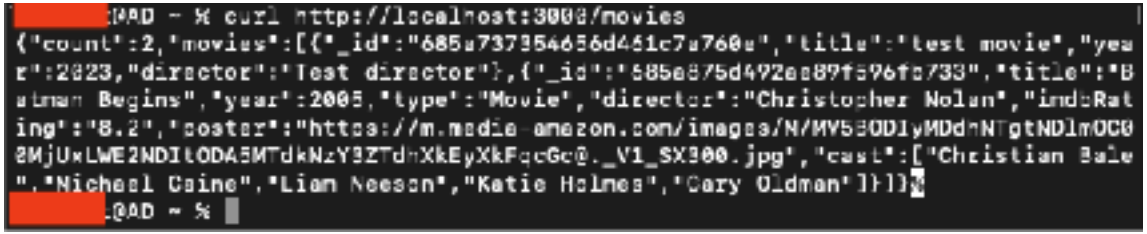
Node running



Mongodb running

Assignment 02 - Screenshots /2

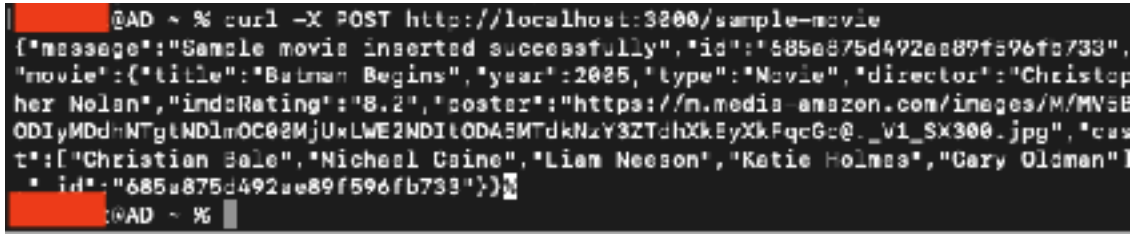
CURL - List all



CURL - Insert

Assignment 02 - Screenshots /3

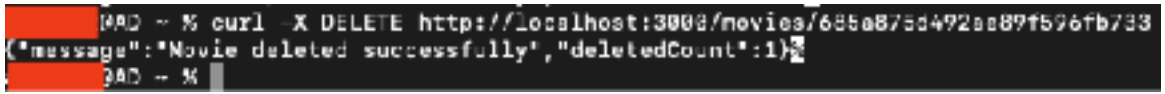
CURL - Insert sample movie



CURL - Update

Assignment 02 - Screenshots /4

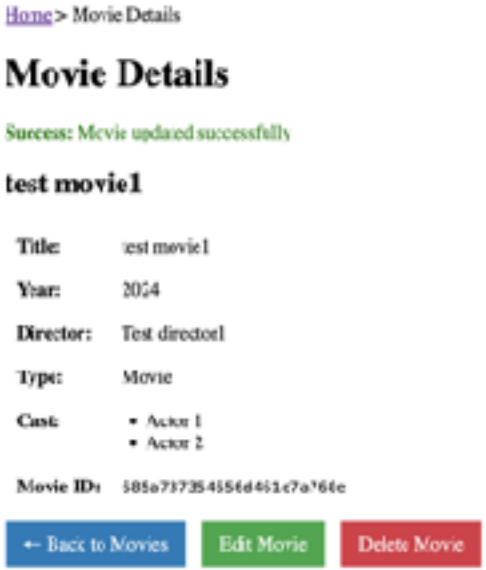
CURL - Insert sample movie



Assignment 02 - Screenshots /5

Listings page

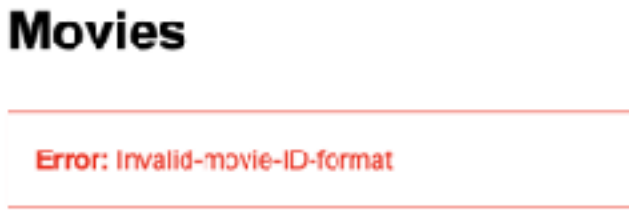
Movie details page



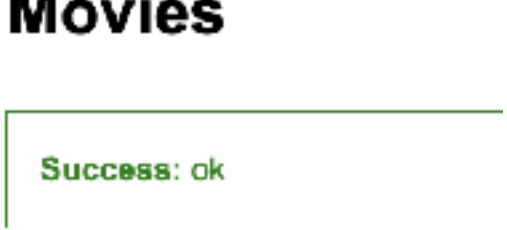
Assignment 02 - Screenshots /6

Edit page

Notification messages - Error



Notification messages - Success



Assignment 02 - Things that could be improved

UI: Visuals, Animations, notifications

Validation: On forms, movie poster URLs, etc

Security: Try to reduce possible XSS, injection attacks

Architecture: Cleaner code