Full Stack Web Development Project: (APIs in Action Bootcamp)

Project Title: Build a Video Upload API using Node.js, Express & MongoDB

Project Description:

Create a backend REST API that allows users to add and manage video records. This assignment will test your understanding of Express routing, MongoDB (Mongoose) integration, validation, and API response handling.

🧠 **Concepts Covered**

- Express.js route handling
- MongoDB connection using Mongoose
- Schema design and validation
- POST and GET API endpoints
- Handling JSON request bodies
- Proper response codes (201, 400, 500)

📂 **Tasks:**

Setup the Project

Initialize a new Node.js project:

npm init -y

Install dependencies:

npm install express mongoose body-parser nodemon

Create a new file named **server.js**.

Connect to MongoDB

- Connect to a local or cloud MongoDB database named podcast.
- Log a message once connected successfully.

Create a Video Schema

Design a schema with the following fields:

Field Name	Type	Required	Description
Title	String	✅ Yes	Title of the video
Description	String	❌ No	Short summary
CreatedAt	Date	Default: now	Timestamp

Implement a POST API

Endpoint: POST /api/videos

Purpose: Add a new video record.

Validation Rules:

- title and videoUrl are required
- If any required field is missing → return 400 Bad Request.

Expected Success Response (201):

{
"success": true,
"message": "Video added successfully!",
"data": {
"_id": "6508f912c8e4c4a1e17b49b2",
"title": "JavaScript Basics",
“Content”:”description”,
"createdAt": "2025-10-28T14:00:00Z"

Implement a GET API

Endpoint: GET /api/videos

Purpose: Retrieve all uploaded videos.

Expected Response (200):

{
"success": true,
"data": [
{
"_id": "6508f912c8e4c4a1e17b49b2",
"title": "JavaScript Basics",
“Content”:”some description”,
"createdAt": "2025-10-28T14:00:00Z"
}
]
}

Bonus Challenges

Add a DELETE route (DELETE /api/videos/:id) to remove a video by its ID.
Add an UPDATE route (PUT /api/videos/:id) to modify video details.
Use Postman or Thunder Client to test all endpoints.
Deploy your API to Render, Vercel, or Railway.

What to Submit:

- GitHub repository link containing:
  - server.js
  - package.json
  - README.md describing:
    - How to install and run the project
    -
    - API endpoints with sample requests

Deadline: Submit your project by **November 6th, 2025 , 11:59pm,** in this submission link - *(https://forms.gle/jVXwmFbzGuXWnSXQ8)*