Creating Custom APIs Using AI APIs with Node.js

Duration: 5 Days

Day 1: RESTful API Foundations with Node.js

1.1 Introduction to APIs

- What is an API?
- REST vs GraphQL
- API request/response flow
- Status codes, headers, and content types

1.2 Setting Up Node.js Project

- Initializing package.json
- Installing Express, nodemon, cors
- Project folder structure: routes/, controllers/, services/

1.3 Creating First REST API

- Basic routes: GET, POST, PUT, DELETE
- Parsing JSON payloads
- Using Postman to test endpoints

1.4 Error Handling and Middleware

- Centralized error handler
- Custom error messages
- Global middleware

Lab 1:

Build a simple task-manager API with:

- Create, Read, Delete tasks
- Use middleware for logging and error handling

Day 2: API Security, Modularization, and Environment Setup

2.1 Environment Configuration

- Storing API keys securely with. env
- Using dotenv in Node.js

2.2 API Structuring Best Practices

- Creating modular routes and controllers
- Reusable services
- Organizing validation and constants

2.3 CORS, Rate Limiting & Security Basics

- Enabling CORS for frontend apps
- Adding basic security headers
- Rate limiting with express-rate-limit

2.4 Using Git & GitHub for Version Control

- Creating a GitHub repo
- Pushing project code
- Managing .env in .gitignore

Lab 2:

Refactor the task-manager API:

- Move logic to controllers/ and services/
- Add .env, rate limiting, and GitHub versioning

Day 3: OpenAI API Integration (ChatGPT, DALL·E)

3.1 Introduction to OpenAI APIs

- Overview: GPT, DALL·E, Whisper
- OpenAI API documentation walkthrough
- Prompt engineering fundamentals

3.2 ChatGPT Integration (gpt-3.5-turbo)

- Install axios for HTTP calls
- Secure API key with .env
- Create /chat endpoint that accepts prompt and returns a GPT response

3.3 Image Generation with DALL·E

- Understanding DALL·E parameters (prompt, size, format)
- Creating /generate-image endpoint

3.4 Prompt Engineering Practice

- Writing system vs user prompts
- Controlling tone, format, creativity

Lab 3:

Create a content-generator API with:

- /blog-summary – summarizes blog posts
- /image-cover – generates an image cover using DALL·E

Day 4: Google Cloud AI & Whisper API Integration

4.1 Using Google Cloud Vision API

- Enabling APIs in Google Cloud
- Creating service account and API key
- Analyzing:
    - Text (OCR)
    - Labels and objects

4.2 Using Google Natural Language API

- Analyzing sentiment, entities, syntax
- Creating /analyze-text endpoint

4.3 Uploading Files with Multer

- Handling audio/image uploads

- Saving to disk or using memory buffer

4.4 Whisper API for Speech-to-Text

- Upload .mp3 or .wav files
- Send to OpenAI Whisper API
- Create /transcribe-audio endpoint

Lab 4:

Build a "Media Intelligence API":

- /analyze-image: returns labels/text from image
- /transcribe-audio: returns text from uploaded audio
- /summarize-transcript: sends transcribed text to ChatGPT for summary

Day 5: Final Project, Testing, and Deployment

5.1 Combining AI APIs in Workflow

- AI workflow: audio → text → analysis → summary
- Chaining multiple AI APIs in one request
- Create /smart-report:
    - Accepts audio
    - Transcribes → summarizes → returns JSON summary

5.2 Input Validation & Robust Error Handling

- Using express-validator or custom checks
- Handling timeouts, malformed inputs, API quota errors

5.3 API Documentation with Postman

- Creating Postman collections
- Documenting request/response schemas
- Exporting and sharing collections

5.4 Deployment

- Deploy to Render or Vercel
- Securing deployed API

Final Lab 5:

Build a “Smart AI Assistant API”:

- Features:
    - /chat: ChatGPT response
    - /transcribe: Whisper audio transcription
    - /analyze: Vision + NLP
    - /smart-assist: Uploads audio/image, performs AI operations, and summarizes results