

# Coding Challenge Options for Flavio Espinoza

## Overview

Hi Flavio, thank you for meeting with us this week! For the next step, please select one of the following coding challenges to complete. Each challenge is sized to be completed in about 4-6 focused hours. You'll have up to 3 calendar days to submit your solution. We care most about your problem-solving, architecture, code quality, and how you approach AI integration.

## General Instructions

**Stack:** Use what you are comfortable with. Examples: React, PHP or Next.js for front end, Node or Next API routes for back end, Python FastAPI if you prefer, PHP or .NET if that is your strength.

**Secrets:** Use an environment variable for API keys. Do not hardcode secrets in code. Provide a .env.example file with variable names.

**Timebox:** Please limit core work to 4 to 6 hours. If you add extra features, list them under Bonus in your README. We expect submissions within 3 days — focus on clarity and working features, not polish.

**Deliverables:** Git repository link or zip, a README with setup steps and design notes, and optional short Loom walkthrough or deployment link.

## OpenAI API Key for this challenge

Use this key via environment variable `OPENAI_API_KEY`. We monitor usage and will revoke the key after the challenge. Do not commit the key to the repository.

```
OPENAI_API_KEY= sk-proj-  
shb7WbCEQcksdSiq6XtG6aY5xMBkz2lh6FyqgP4NspcmA-3lMBuUTcxR3o1n3OVtWXTSz23fZHT3BibkFJX4pkwldxfIBYGuBbBDkq4ff  
mUPJ_m8jc0xMtPs1g2jfmITBtNVBIEeGMA9pSRe3qOguhiKMuoA
```

## What we evaluate

1) Core functionality and UX. 2) Code quality and structure. 3) AI integration and prompt design. 4) Data model and API design. 5) Developer experience and documentation.

## Submission

Reply with the repo link or zip, plus any optional video or deploy URL. For questions, email [leonmoyer@ftmedia.com](mailto:leonmoyer@ftmedia.com). Please include your start and end timestamps and note roughly how long you spent.

## 1. AI Powered Text Summarizer

**Goal:** Build a web app that allows users to paste in long text and get an AI generated summary.

### Requirements:

- React front end with a text area input and Summarize button
- Backend endpoint in Node Express or a Next.js API route calling the OpenAI API

- Store original text and summary in a database such as SQLite, Postgres, or MongoDB
- Display a list of past summaries below the form

**Bonus Ideas:**

- Adjustable summary style such as concise, detailed, or bullet points
- Basic rate limiting or simple abuse guard
- Search or filter on saved summaries

**Skills Tested:** API integration, async handling, data modeling, UI and UX design, persistence layer

## 2. AI Blog Title Generator and Mini CMS

**Goal:** Build a tool that generates blog title ideas from a topic and allows saving favorites.

**Requirements:**

- Input field for the topic idea such as online poker strategies
- Call OpenAI to generate five title suggestions
- Click to save a title which is stored in the database
- Optional tag or category per saved title

**Bonus Ideas**

- Regenerate five more that avoid duplicates
- Edit and delete saved titles
- Simple search or tag filter

**Skills Tested:** AI integration, CRUD operations, state management, small CMS style app design

## 3. Mini SEO Report Generator

**Goal:** Create a tool that takes a website URL, pulls metadata and page title, and gives AI generated SEO feedback.

**Requirements:**

- Input for website URL
- Backend fetches metadata using cheerio or puppeteer, or an equivalent approach
- Send the data to OpenAI with a prompt asking for SEO improvement tips
- Display the AI response and save the report to a database

**Bonus Ideas:**

- Basic checks such as title length and missing description
- Save and compare historical reports
- Export the report to Markdown or PDF

**Skills Tested:** Scraping, backend logic, AI prompt design, practical app use case

## 4. Interactive Options Strategy Calculator

**Goal:** Simulate a simple options profit calculator.

**Requirements:**

- Form with inputs for strike price, premium, and stock price at expiration
- Calculate and display profit or loss across price levels
- Chart the payoff curve using Chart.js or Recharts

**Bonus Ideas:**

- AI explanation that describes the strategy type and likely sentiment
- Multiple strategies such as covered call and protective put

**Skills Tested:** Frontend logic, chart rendering, optional AI reasoning layer

## 5. Internal Team Dashboard with AI Assistant

**Goal:** Create a dashboard for an internal team with an AI assistant feature.

**Requirements:**

- Login free dashboard
- Ask Assistant text box that sends a prompt to OpenAI
- Recent messages and answers listed below
- Button to save any answer to a knowledge base
- Store Q and A pairs in a database and display them in a sidebar

**Bonus Ideas:**

- Tagging and search for saved Q and A
- Pinned answers or simple categories
- Basic content moderation

**Skills Tested:** Full stack coordination, internal tooling mindset, saving structured data, conversational UI