Context

Software engineering is already changing with the use of LLMs. LLMs can now write a whole game with just one prompt, and within a year they will be able to develop a fully working app/website…

So the role of the software engineer will be to use LLMs effectively and efficiently, and to make sure the code quality is good by:

1- Dividing a large project into smaller chunks that can be better understood, tested, and reused, and making sure that the code has a lot of error catching and handling to make the development easier.

2- Creating tests with great coverage (by asking the LLMs to create the tests and modifying them)

3- Setting up a powerful development environment, between code repos, continuous integration, and automatic collection of bugs. Again, this would be done by guiding LLMs to do the details while the engineer is focusing on the quality.

That is the role we are hiring for, and so the task is focused on the larger process and the quality of the overall approach.

Objective

Create a simple REST API service that manages a to-do list, utilizing LLMs throughout the development process. The focus is on demonstrating your ability to effectively use LLMs for various stages of the software development lifecycle.

Requirements

<u>1. API Development</u>

- Develop a RESTful API with the following endpoints:
  - GET /tasks: Retrieve all tasks
  - POST /task: Create a new task
  - PUT /task/{id}: Update a task
  - DELETE /task/{id}: Delete a task
- Use any backend framework of your choice (e.g., Flask, FastAPI for Python, or Express for Node.js)

2. LLM Usage

- Utilize an existing LLM API (we recommend GPT4o or Claude) for the following processes:
  - Code generation
  - Test case creation
  - Documentation writing
  - Code review suggestions
  - Monitoring alert analysis

3. Development Process

Demonstrate the use of LLMs in the following stages:

a. Code Generation
- Use the LLM to generate the initial code structure and implementation for your API
- Show how you prompt the LLM and how you integrate its output into your development process

b. Testing
- Generate unit and integration tests using the LLM
- Implement the generated tests and show how you validate them

c. Documentation
- Use the LLM to generate API documentation, including endpoint descriptions and usage examples

d. Deployment
- Create a deployment script or configuration using LLM-generated suggestions
- Implement a simple CI/CD pipeline, possibly using LLM for configuration file generation

e. Monitoring
- Set up basic monitoring for your API (e.g., request rate, error rate)
- Use the LLM to generate and interpret example monitoring alerts

f. Iterative Improvement
- Simulate a scenario where you need to add a new feature or fix a bug
- Show how you use the LLM to assist in understanding the problem, generating a solution, and implementing the change

4. Process Documentation

- Document your entire process, including:
  - How you prompt the LLM for each task
  - How you validate and refine LLM-generated content

- Any challenges you faced and how you overcame them
- Reflections on the effectiveness of using LLMs in different stages of development

** PS, if you keep track of what you're doing, you can have the LLM write all of that for you.

Evaluation Criteria

1. Effective use of LLMs across all stages of development
2. Quality and correctness of the final code and API
3. Thoroughness of testing and documentation
4. Clarity and efficiency of the development process
5. Critical thinking in validating and refining LLM outputs
6. Creativity in leveraging LLMs for software development tasks

Submission

- Provide a GitHub repository with your code, tests, and documentation
- Include a detailed README.md explaining your process and how to run the project
- Be prepared to discuss your approach and decision-making process in a follow-up interview

Time Frame

You have 1 week to complete this task. Quality is more important than speed, but try to showcase a streamlined, LLM-assisted development process.