

## Introduction

Hi there! Thank you for signing up to take this coding exercise. I wanted to take a moment to set the stage for what we're looking for and share some insights into how we'll evaluate your work.

This exercise isn't just about delivering a feature—it's about showcasing how you approach software design and development holistically. Let me explain.

---

### 1. Evaluating Your Design Skills

First, we'll be looking at your **design skills**. This includes how well you apply **object-oriented analysis and design (OOAD)**, your use of **design patterns**, and adherence to **SOLID principles**. We're particularly interested in how you create **modular components** that are **decoupled** and maintainable over time. Your design choices should demonstrate thoughtful architecture that can evolve as the application grows.

---

### 2. Evaluating Your Development Skills

Next, we'll evaluate your **development skills**. This isn't just about writing code—it's about how you **evolve your implementation** and how you manage the process. We'll be looking at:

- **Git usage:** How you organize commits, use branches, and collaborate with version control.
- **Code structure:** Is it clear, well-organized, and intuitive?
- **Documentation:** Are your code and processes well-documented for other developers to understand?
- **Automation testing:** Have you included sufficient test coverage to ensure code reliability?
- **Deployment:** We want to see deployment instructions that are clear and functional, both for **developers** and **production environments**.

Remember, we value quality over quantity. It's better to submit a small, complete, and polished solution than a feature-complete one that lacks these essential aspects.

---

### 3. Evaluating Your Understanding of Technology and Concepts

We're also evaluating your grasp of **technology and concepts**. This includes your familiarity with:

- **Frameworks and languages:** Node.js, NestJS, Angular, Python, FastAPI.
- **Core web technologies:** HTTP, cloud, and containerization.
- **Critical software considerations:** Authentication, security, and scalability.

Show us that you understand not just the "how" but the "why" behind your technology choices.

---

### 4. What We Are Not Looking For

Now, let me clarify what's **not acceptable**:

- **Procedural implementations:** We expect a structured, object-oriented or modular approach.
- **Missing automation tests:** Testing is non-negotiable.
- **Lack of documentation:** Your work should be easy to understand and maintain.
- **No deployment instructions:** Ensure deployment is clear and reproducible.
- **Improper Git usage:** Your Git history should reflect thoughtful, professional practices.

This isn't about rushing to get "feature complete." Instead, focus on making sure whatever you implement is complete **in all aspects**—from design to development to deployment.

---

### Conclusion

Finally, remember that this exercise is your opportunity to showcase how you think, design, and build. Take your time, approach it thoughtfully, and don't hesitate to showcase your best practices.

Good luck, and we look forward to seeing your work!