Hiring Assignment

For a pre-hiring assignment, you could design a task that evaluates the candidate's proficiency in Python FastAPI, PostgreSQL, Docker, and Pydantic. Here's an idea for an assignment:

Assignment: Build a Simple Book Management API

Task Overview:
You are required to build a small API for managing a collection of books using Python FastAPI, PostgreSQL, Docker, and Pydantic. The API should allow users to perform CRUD (Create, Read, Update, Delete) operations on a book database.

Requirements:

- ☐ API Endpoints:
  - ☐ POST /books/: Add a new book to the collection. The book should have the following attributes:
    - ☐ Title (string)
    - ☐ Author (string)
    - ☐ Published Year (integer)
    - ☐ ISBN (string)
    - ☐ Available (boolean)
  - ☐ GET /books/: List all books in the collection.
  - ☐ GET /books/{book_id}: Get details of a specific book by ID.
  - ☐ PUT /books/{book_id}: Update details of a specific book by ID.
  - ☐ DELETE /books/{book_id}: Delete a specific book by ID.
- ☐ Database:
  - ☐ Use PostgreSQL to store book data.
  - ☐ Design the database schema accordingly (using migrations is a plus, but not mandatory).
- ☐ Data Validation:
  - ☐ Use Pydantic for data validation and data handling.
- ☐ Docker:
  - ☐ Create a Dockerfile and docker-compose.yml to run the FastAPI app and PostgreSQL in containers.
- ☐ Documentation:
  - ☐ The API should have interactive documentation available at /docs using FastAPI's automatic documentation generation feature (Swagger).

Bonus Points:

- [ ] Include unit tests for key functionality.
- [ ] Implement pagination for the list of books.
- [ ] Provide detailed instructions in the README on how to set up and run the project using Docker.

This assignment tests their ability to:

- [ ] Set up a basic FastAPI project
- [ ] Work with PostgreSQL for data persistence
- [ ] Containerize applications using Docker
- [ ] Validate data using Pydantic
- [ ] Write clean, structured, and maintainable code
- [ ] Knowledge of SOLID principles