AI Engineering Internship Assignment: Model Deployment with FastAPI

Topic Chosen: Deploying Machine Learning Models with FastAPI

This assignment will test your ability to:

- Learn a technical AI engineering concept (model deployment).
- Teach it back to your peers using a clear, well-structured presentation.
- Demonstrate deep technical understanding through a full code walkthrough.
- Solve a real-world challenge and communicate your solution effectively.

1. Learning Resources

Use the following curated references to master the topic:

Resource Type	Title/Description	Link
Documentation	FastAPI Official Documentation (core concepts, API building)	https://fastapi.tiangolo.com/
Blog	Deploying ML Models with FastAPI: End-to-End Guide	https://towardsdatascience.com/deploying-machine-learning-models-with-fastapi-6c3e6b7e5a3c
YouTube	Deploy ML Models as APIs with FastAPI (Full Tutorial)	https://www.youtube.com/watch?v=0sOvCWFmrtA
Blog	Production-Ready ML Model Deployment Using FastAPI and Docker	https://mlengineer.io/blog/fastapi-docker-ml-deployment
Blog	Serving ML Models with FastAPI and Docker	https://www.analyticsvidhya.com/blog/2021/06/serving-machine-learning-models-using-fastapi-and-docker/

2. Assignment Instructions

A. Learn & Summarize

- Study the above resources to understand:
    - What is FastAPI and why is it used for model deployment?
    - How do you build, test, and deploy an API for a machine learning model?
    - Best practices for productionizing ML models.

B. Teach the Concept

- Prepare a **PPT (8–12 slides)** that:
    - Explains FastAPI and its relevance in AI engineering.
    - Illustrates the steps to deploy a machine learning model as an API.
    - Uses diagrams, code snippets, and real-world analogies.
    - Is tailored for new joiners with basic Python/ML background.

C. Code Walkthrough

- Choose a simple ML model (e.g., Iris classifier, sentiment analysis).
- Build and save the model in Python (using scikit-learn or similar).
- Develop a FastAPI application to serve predictions.
- **Full Code Walkthrough:**
    - Explain each section: model loading, API endpoints, request/response structure, error handling, and testing.
    - Discuss the math behind the model (e.g., how logistic regression or decision trees work).
    - Highlight deployment considerations (e.g., Docker, scaling, monitoring).

D. Real-World Challenge

Scenario:
A client wants to automate flower species identification from petal/sepal measurements (Iris dataset). They need a REST API that receives measurements and returns the predicted species.

- Build the solution end-to-end:
    - Train a model on the Iris dataset.
    - Deploy it as a FastAPI service.
    - Demonstrate the API with sample requests.

E. Video Submission

- Record a **video [NO time constraint]** :
    - Teach the FastAPI deployment concept using your PPT.
    - Walk through your code in detail, explaining logic, math, and design choices.
    - Demo your API solving the real-world challenge.

- Upload the video to Google Drive (or similar) and share a link with **download or read access so that we could evaluate it.**

3. Submission Checklist in the drive

- PPT file (Google Slides or PDF)
- Well-commented code (GitHub or zip)
- Video link (Google Drive with download/read access)
- Short summary (2–3 sentences) of your project and learnings [.docx]

4. Evaluation Criteria

- Clarity and depth of concept explanation
- Quality and accuracy of code walkthrough (including math and technical reasoning)
- Effectiveness of teaching (PPT and video)
- Relevance and completeness of real-world solution
- Communication skills and ability to simplify technical content

Tip:
Teaching a concept simply is the best test of your own understanding. Focus on clarity, structure, and real-world relevance.