**🧭 Project: Stock Price Movement Predictor (Java + ML)**

**Goal:** Predict if a stock will go up or down tomorrow using a Python ML model + Java backend.

**📁 PHASE 1: Setup & Plan (Day 1)**

**✅ Tasks:**

1. Create a GitHub repo
2. Decide on project folder structure (Java + Python in separate folders)
3. Write a README with:
   * Goal of the project
   * Tools/tech stack
   * Planned features
4. Install Postman (for testing APIs)

**📚 Resources:**

* [How to create a README](https://www.makeareadme.com/)
* Postman download

**🟦 PHASE 2: Java Backend (Day 2–5)**

**✅ Tasks:**

1. Set up a Spring Boot project
2. Create REST endpoint /predict
3. Use RestTemplate or WebClient to call Python FastAPI service
4. Accept ticker input (e.g., AAPL), and send it to Python
5. Receive prediction from Python and show in response

**📚 Resources:**

* [Spring Initializr (to create project)](https://start.spring.io/)
* Spring Boot REST API tutorial
* Calling external API from Spring Boot using RestTemplate

**🟨 PHASE 3: ML API with FastAPI (Day 6–8)**

**✅ Tasks:**

1. Set up Python project
2. Install FastAPI + Uvicorn
3. Install yfinance to fetch stock data
4. Use Logistic Regression to predict UP/DOWN (I’ll provide base code)
5. Expose a POST API endpoint /predict that takes ticker and returns prediction + confidence

**📚 Resources:**

* [FastAPI Crash Course (YouTube)](https://www.youtube.com/watch?v=0sOvCWFmrtA)
* [yfinance docs](https://pypi.org/project/yfinance/)
* Basic ML with scikit-learn

**🔄 PHASE 4: Java ↔ Python Integration (Day 9)**

**✅ Tasks:**

1. Make sure your FastAPI is running locally (on localhost:8000)
2. From Java, call the FastAPI endpoint with ticker data
3. Receive prediction result
4. Display result in a JSON or HTML page

**📊 PHASE 5: Bonus Features (Optional)**

| **Feature** | **Tools to Learn** |
| --- | --- |

|  |  |
| --- | --- |
| Show stock price graph | * + Chart.js (JS) or use table in Java |

|  |  |
| --- | --- |
| Store user searches | * + MongoDB / PostgreSQL |

|  |  |
| --- | --- |
| Deploy ML API - online | Render / Railway / Heroku |

|  |  |
| --- | --- |
| Authentication | * + Spring Security |