**🔁 Phase 1: Data Science Core (Jupyter OK here)**

Get insights and build a strong model — fast and clean.

1. **Problem Framing**
   * Define the business problem: *“Predict which customers are likely to churn so the company can intervene.”*
   * State assumptions and goals clearly.
2. **Data Exploration**
   * Understand columns, types, missing values.
   * Visualize churn vs. features.
   * Summarize insights.
3. **Data Cleaning + Preprocessing**
   * Handle nulls, encode categoricals, scale numerical values.
   * Split data (train/test), set random seed for reproducibility.
4. **Modeling**
   * Try Logistic Regression, Random Forest, XGBoost.
   * Evaluate using:
     + Accuracy, Precision, Recall, F1, AUC
     + Confusion Matrix
   * Choose best model.
5. **Feature Importance + Interpretation**
   * Use SHAP or model feature importances to explain predictions.
   * Visualize top drivers of churn.

**🚀 Phase 2: Build a Real App Backend (API)**

No more notebook — you’re moving to real engineering.

1. **Export the model**
   * Use joblib or pickle to save your trained model.
2. **Create FastAPI App**
   * Load model
   * Define endpoint /predict that accepts user input (JSON) and returns churn prediction.
3. **Test API locally**
   * Use Postman or curl to test your prediction endpoint.

**💻 Phase 3: Frontend or Dashboard**

Make it accessible to non-tech users.

1. **Create a simple frontend**
   * Option A: Build a clean **Streamlit app** with input form + prediction display.
   * Option B: Plain HTML/JS page + AJAX that hits the FastAPI endpoint.
2. **Optional: Dashboard**

* Use Streamlit or Dash to create a dashboard showing:
  + Churn rates
  + Predictions
  + Feature impacts

**☁️ Phase 4: Deployment**

Put it on the internet. This is what separates you from notebook warriors.

1. **Containerize (Optional but strong)**

* Use Docker to package the app.

1. **Deploy**

* Option A: Use **Render.com** for FastAPI + Streamlit
* Option B: Use **Railway**, **Heroku**, or **AWS EC2**

1. **Test + Monitor**

* Confirm app is live and working
* Add basic logging to API

**🧾 Phase 5: Documentation + GitHub Polish**

Make your repo recruiter-ready.

1. **README**

* Problem → Solution → Tools used → How to run the app → Demo link

1. **Code Structure**

* /app, /model, /notebooks, /static folders
* requirements.txt or pyproject.toml

1. **Optional Extras**

* Add unit tests to your API
* Add a CI pipeline with GitHub Actions
* Link to LinkedIn and medium article if you write one