AI Development Workflow Project Summary

1. Overview

This project simulates a hospital readmission prediction system using a Flask-based API. It is part of a practical AI development workflow, including data processing, model logic, API mockup, and final documentation. The use case focuses on predicting whether a patient will be readmitted based on key indicators such as age and diagnosis.

2. Project Components

- 📄 `readmission_model.py`: Core logic for predicting readmission.
- 🧪 `readmission_model.ipynb`: Experimental notebook used for development and testing.
- 🗃️ `sample_readmission_data.csv`: Sample dataset with patient age and diagnosis.
- 🌐 `app.py`: Flask API mock simulating hospital readmission predictions.
- 📝 `README_AI_Workflow.md`: Descriptive GitHub documentation.
- 📄 Word report and PDF: Contains written reflections, theoretical parts, critical thinking, and visuals.
- 📊 AI_Workflow_Diagram.png: Visual representation of the AI workflow.

3. Running the Flask Mock API

To run the app locally:
1. Activate the virtual environment: `source .venv/bin/activate`
2. Install dependencies: `pip install flask pandas`
3. Run the API: `python app.py`
4. Visit: http://127.0.0.1:5000/
5. To predict: Send POST request to `/predict` with JSON body:

```json
{
  "age": 70,
  "diagnosis": "diabetes and high blood pressure"
}
```

4. Deployment (Optional)

For remote deployment, you can use Replit or Render.com.
- On Replit: Import GitHub repo → Add `flask` and `pandas` in Packages → Run `app.py`
- On Render.com: Create new web service → Connect GitHub → Select repo → Set `Start Command`: `gunicorn app:app`

5. Submission Details

- ✅ GitHub repository: [Your link will be inserted here]
- ✅ Word report: `AI_Development_Workflow_Final_With_Critical_Thinking.docx`
- ✅ PDF copy: Shared to PLP Academy community.
- ✅ Flask mock server tested and running at http://127.0.0.1:5000/