👨‍💻 Nav's Work (Completed)

1. **Dataset Setup**
    - Used the Kaggle `fake_job_postings.csv` dataset.
    - Added **synthetic LinkedIn-like company stats**:
        - `followers`, `employees`, `engagement`.
    - Combined job text fields (`title + description + requirements + company_profile`).
2. **Model Training**
    - Built a pipeline with **TF-IDF Vectorizer + Logistic Regression**.
    - Trained model → achieved **~97% accuracy**.
    - Saved trained model to `scam_detector.pkl` (binary file).
3. **Explainability Layer**
    - Added a **list of scammy keywords** (e.g., "work from home", "quick money", "earn $").
    - Model now highlights suspicious phrases if they appear.
    - Flags weak company stats (low followers, very few employees, low engagement).
4. **Helper Script (`scam_detector.py`)**
    - Contains `predict_job()` function.
    - Input: job ad details + company stats.
    - Output: JSON-like dict with:
        - `prediction` (scam/legit)
        - `confidence` (0–1)
        - `keywords_triggered` (list of scammy words found)
        - `weak_company` (True/False).
    - Added **self-test** → running `python scam_detector.py` shows one scam example and one legit example.

5. **Notebooks**
    - `scam_detector.ipynb` contains the full training + evaluation + explainability steps.
    - For reference only, in case retraining is needed.

📂 Folder Structure

```
job-scam-detector/
│
├── scam_detector.ipynb     # Full training + explainability (Nav's notebook)
├── scam_detector.pkl       # Trained ML model (binary, do not open in editor)
├── scam_detector.py        # Prediction + explainability helper
└── fake_job_postings.csv   # Dataset used
```

🟢 How to Run (Quick Start)

Install dependencies (Python 3.11 recommended):

```
pip install pandas numpy scikit-learn joblib
```

1.

Run the helper script:

```
python scam_detector.py
```

2.
3. You will see:
    - 🚨 One scam job flagged (with keywords + weak company stats).
    - ✅ One legit job flagged (no scam keywords, strong company).

🟡 Next Steps (Who Does What)

Dish (Backend)

- Wrap `predict_job()` from `scam_detector.py` in an API (FastAPI/Flask).
- API endpoint example:
    - **POST /predict** → returns scam/legit + explanation.

Vig (Data Enhancer)

- Mock or fetch LinkedIn-like data (followers, employees, engagement).
- Pass this data into the API alongside job text.
- Optionally, build a “Trust Score” (0–100) from company stats.

Kusuma (Frontend)

- Build simple UI → user pastes job ad link/text.
- Call backend API → display results:
    - Scam or Legit (red/green).
    - Confidence % bar.
    - Highlight scammy keywords in red.
    - Show company trust score dial.

🏁 Summary

- ✅ Nav delivered the **AI engine** (trained model + explainability).
- 🔜 Dish builds API → connects ML to the outside world.
- 🔜 Vig provides company stats → strengthens detection.
- 🔜 Kusuma builds UI → makes it visual & demo-ready.