NIRA: Network Incident Response Assistant

NIRA is an advanced AI-powered system designed to detect network intrusions in real-time and generate human-readable incident reports.

NIRA performs:

- **Real-time intrusion detection** using an ensemble ML model (XGBoost + Random Forest + Logistic Regression)
- **Automated incident reporting** (via GPT-4o-mini, optional)
- **Live threat visualization** on a browser dashboard

This README provides **simple, clean, step-by-step instructions** for running the full project.

Project Structure

```
NIRA/
├── nira_backend/
│   ├── main_v2.py
│   ├── reporter_v2.py
│   ├── traffic_simulator_v2.py
│   ├── UNSW_NB15_testing-set.csv
│   ├── nira_ensemble_model.joblib
│   ├── nira_preprocessor.joblib
│   ├── nira_label_encoder.joblib
│   └── nira_feature_lists.joblib
│
└── nira_frontend/
    └── index.html
```

Prerequisites

Please install:

- Python 3.8+
- Modern web browser
- OpenAI API Key for improved incident reports

Backend Setup

Navigate to:

```
nira/nira_backend
```

1. Create & activate virtual environment

Windows

```
 python -m venv venv
venv\Scripts\activate
```

macOS / Linux

```
 python3 -m venv venv
source venv/bin/activate
```

---

2. Install dependencies

Use this exact installation command:

```
pip install fastapi "uvicorn[standard]" websockets requests pandas xgboost scikit-learn==1.6.1 joblib httpx openai
```

Add OpenAI API Key

1. Open `reporter_v2.py`
2. Modify:

```
openai_api_key = "sk-xxxx"
```

HOW TO RUN THE PROJECT

STEP 1 — Start the Backend Server

In the backend folder, run:

```
uvicorn main_v2:app --reload
```

Expected output:

```
 Server running on http://127.0.0.1:8000
Models loaded successfully
```

Keep this terminal open.

STEP 2 — Open the Frontend Dashboard

Open this file:

```
nira/nira_frontend/index.html
```

(You can double-click it.)

If backend is running, the dashboard will show:

- **Connected**
- Empty alert log (initially)
- World map loaded

STEP 3 — Start the Traffic Simulator

Open a **second terminal**, then re-activate the virtual environment:

Windows

```
venv\Scripts\activate
```

macOS / Linux

```
source venv/bin/activate
```

Run:

```
python traffic_simulator_v2.py
```

You should now see:

```
 Packet sent
Packet processed
Alert detected
```

And the dashboard will begin displaying alerts in real time.
