# README: AI-Powered Phishing & Threat Intelligence Detection System

## Problem Statement

Phishing remains one of the most prevalent cyber threats worldwide. It targets individuals and organizations via deceptive emails, malicious links, and SMS messages to steal sensitive information. Traditional spam filters and blacklists are reactive and ineffective against zero-day phishing tactics. This project aims to develop a proactive, AI-driven detection system that flags phishing attempts with high accuracy and minimal latency.

## Objectives

* Detect phishing attempts in:  
   - Emails (subject, body, sender)  
   - URLs (reputation, structure, redirect behavior)  
   - SMS/text messages
* Use NLP and AI models for classification
* Integrate public threat intelligence APIs (e.g., VirusTotal, AbuseIPDB)
* Provide real-time inference via RESTful API
* Visualize threats and logs via a dashboard
* Add a browser extension for live detection (Stretch Goal)

## Features

* AI/ML-powered phishing detection (emails, URLs, messages)
* Real-time backend API with JSON responses
* Threat enrichment via intelligence APIs
* Admin dashboard for analytics and monitoring
* Secure data handling and logging
* Stretch Goals (in progress): Browser extension, Feedback loop

## Tech Stack

Programming: Python  
ML/NLP: Scikit-learn, BERT, HuggingFace Transformers  
Data: Pandas, NLTK, PhishTank, VirusTotal API  
Backend API: FastAPI / Flask  
Dashboard: React.js / Streamlit  
Database: SQLite / PostgreSQL  
DevOps: Git, Docker (optional), Postman

## System Architecture

1. Data Collection (PhishTank, scraped examples)  
2. Preprocessing & Feature Engineering (NLP-based)  
3. Model Training (ML & Deep Learning Models)  
4. Inference API (FastAPI or Flask)  
5. Threat Intelligence Lookup (VirusTotal, AbuseIPDB)  
6. Logging, Alerting, and Visualization

## Getting Started

1. Clone the Repository:  
 git clone https://github.com/your-username/phishing-threat-intel-system.git  
 cd phishing-threat-intel-system  
  
2. Create and Activate a Virtual Environment:  
 python -m venv venv  
 source venv/bin/activate # Windows: venv\Scripts\activate  
  
3. Install Dependencies:  
 pip install -r requirements.txt  
  
4. Run the Application:  
 python app.py # or use uvicorn for FastAPI  
  
5. Access Dashboard:  
 http://localhost:8000 (or as specified)

## Example API Request

POST /predict  
  
{  
 "email\_subject": "Urgent: Your Account is Suspended",  
 "email\_body": "Click here to verify your account...",  
 "url": "http://suspicious-link.com"  
}  
  
Response:  
{  
 "is\_phishing": true,  
 "confidence\_score": 0.94,  
 "threat\_report": {  
 "domain\_reputation": "blacklisted",  
 "url\_risk\_score": 92  
 }  
}

## Project Structure

📦 phishing-threat-intel-system/  
├── data/  
├── models/  
├── api/  
│ ├── app.py  
│ └── inference.py  
├── dashboard/  
├── utils/  
│ └── preprocessing.py  
├── requirements.txt  
├── README.md  
└── LICENSE

## Security Considerations

- Secure API access using authentication keys  
- Encrypted storage of logs and sensitive inputs  
- CORS and input sanitization implemented in API layer

## Future Improvements

- Browser extension for live phishing alert  
- Multilingual phishing detection  
- Real-time user feedback loop  
- Deployment on cloud (Render, AWS, Heroku)

## Contributors

- 👨‍💻 Cybersecurity Lead: Your Name  
- 📊 Data Scientist: Friend 1 Name  
- 🤖 AI Engineer: Friend 2 Name

## License

This project is licensed under the MIT License - see the LICENSE file for details.