# 🚨 Phishing URL Detection App

An end-to-end machine learning application that predicts whether a given URL is phishing or legitimate.

## 📌 Features

- 🧠 ML model with reduced feature set

- 📊 Explainability using SHAP and LIME

- ⚙️ FastAPI backend with REST API endpoints

- 🎨 Streamlit frontend for user interaction

- 📦 Dockerized for easy deployment

---

## 🗂️ Project Structure

```

phishing\_app/

│

├── main.py # FastAPI backend

├── ui/stream.py # Streamlit frontend

├── model/ # Saved ML models and scalers

├── utilitis/ # Custom utility modules

│ └── explain.py # SHAP and LIME explanation logic

├── requirements.txt # Python dependencies

├── Dockerfile # Docker build instructions

├── start.sh # Entrypoint script

└── README.md / docs.md # Project documentation

```

---

## 🧪 How it Works

1. User inputs a URL in the Streamlit frontend.

2. Frontend sends a request to FastAPI's `/predict` endpoint.

3. Backend returns a prediction (Phishing/Legitimate).

4. User can choose explainability:

- SHAP: Global feature impact

- LIME: Local explanation for one prediction

---

## 🐳 Docker Instructions

### 📦 Build the Docker Image

```bash

docker build -t phishing-app .

```

### 🚀 Run the App

```bash

docker run -p 8501:8501 -p 8000:8000 phishing-app

```

- Access the file : [http://localhost:8501](http://localhost:8501)

---

## 🛠 API Endpoints (FastAPI)

| Endpoint | Method | Description |

|------------------|--------|-----------------------------------|

| `/predict` | POST | Returns prediction from model |

| `/explain/shap` | POST | Returns SHAP explanation |

| `/explain/lime` | POST | Returns LIME explanation |

Example `POST` body:

```json

{

"url": "http://example.com/login"

}

```

---

## 📋 Requirements

fastapi

uvicorn

pydantic

scikit-learn

xgboost

joblib

pandas

numpy

matplotlib

shap

lime

streamlit