Project Title: URL Safety & Authenticity Checker Web App

Problem Statement:

In today's digital world, users are frequently exposed to unsafe, suspicious, or fake websites. These sites can steal personal information, spread malware, or manipulate users. There is a need for an intelligent web-based tool that can determine whether a given URL is safe, suspicious, or fake using both rule-based and machine learning techniques.

College Name: Parul Polytechnic Institute

Mentor Name: Drashty Shah

Mentor Email: drashty.shah31524@paruluniversity.ac.in

Team Name: D-coders

Team Members:

- Team Leader: Manav Barot
- Member 1: Kush Patel
- Member 2: Heer Patel

Problem Domain: AI/ML for Networking

Used Technologies:

- Frontend: HTML, CSS, JavaScript
- Backend: Python, Flask
- Machine Learning: Scikit-learn

- Dataset: URL datasets with label annotations (safe, suspicious, malicious)

- Deployment: Localhost (can be extended to cloud)

Deployment:

- Project successfully deployed locally using Flask development server.

- Final version tested on multiple sample URLs for real-time classification.

- Web app shows prediction results instantly with a simple user interface.

Each Team Member’s Work:

- Manav (Team Leader):

- Developed and trained the ML model

- Created Flask backend logic

- Integrated model with frontend

- Resolved runtime issues during prediction

- Kush:

- Designed and styled the frontend (HTML/CSS)

- Built the input/output interface

- Handled user experience feedback implementation

- Heer:

- Collected and cleaned the dataset

- Performed feature extraction on URLs

- Assisted in model evaluation and testing

Architecture Diagram:

[Insert Architecture Diagram Image Here – Optional description below:]

- User submits URL → Flask API receives input → URL pre-processed and passed to ML model → Prediction returned to frontend → Status displayed (Safe/Suspicious/Fake)

Code Repository:

GitHub: https://github.com/barotmanav/-URL-Safety-Authenticity-Checker-Web-App

Errors Encountered and Solutions:

1. Error: Model not predicting correct classes for new inputs

- Solution: Rebalanced dataset and applied better feature extraction (length of URL, presence of symbols like '@', etc.)

2. Error: Flask server crashing on large inputs

- Solution: Added input validation and error handling for malformed URLs

3. Error: Mismatch between frontend and backend response format

- Solution: Standardized JSON response from backend and updated JS parser logic

4. Error: HTML not updating the URL status

- Solution: Fixed JavaScript DOM logic to properly reflect model result dynamically

Future Scope:

- Add database logging for history of checks

- Deploy to cloud using Render or Hugging Face Spaces

- Enhance model using deep learning URL embeddings

Conclusion:

The project successfully demonstrates an AI-powered web app capable of analyzing the

safety of URLs. It integrates multiple technologies to offer a real-time security utility for end users.

Prepared by Team D-coders for Hackathon 2025.