**Project Title:** Phishing URL Detection using Machine Learning

**1. Introduction:** Phishing is a prevalent form of cyberattack where malicious websites mimic legitimate ones to steal sensitive user data like login credentials and financial information. This project aims to detect phishing websites by analyzing their URLs using machine learning techniques. By training a model on various features extracted from URLs, we can effectively classify whether a given URL is legitimate or malicious.

**2. Objectives:**

* To develop a machine learning model that can detect phishing URLs.
* To automate the feature extraction process from raw URLs.
* To provide an easy-to-use interface for users to check URLs.
* To improve online safety by warning users of potentially harmful websites.

**3. Technologies Used:**

* Programming Language: Python
* Libraries: Pandas, Scikit-learn, Joblib, Tkinter, Flask
* IDE: VS Code / PyCharm / Jupyter Notebook
* Dataset: Collection of URLs labeled as "phishing" or "legitimate"

**4. Modules of the Project:**

* **Feature Extractor:** Extracts characteristics from a URL (e.g., presence of IP, use of HTTPS, suspicious words).
* **Model Trainer:** Uses extracted features to train a Random Forest Classifier and saves the trained model.
* **Predictor:** Loads the model and predicts if a URL is phishing based on extracted features.
* **GUI (Tkinter):** User interface to enter a URL and see the prediction result.
* **Web App (Flask):** Optional web interface for the same functionality.

**5. Methodology:**

* Collect and clean a labeled dataset of URLs.
* Extract relevant features from each URL.
* Split data into training and testing sets.
* Train a Random Forest model and evaluate its accuracy.
* Deploy the model with a GUI/web interface.

**6. Result:**

* Achieved high accuracy (~96-97%) on the test set.
* Successfully predicted phishing URLs using both GUI and web interfaces.
* Model performs well in real-time predictions.

**7. Conclusion:** The project demonstrates an effective way to combat phishing by analyzing URLs with machine learning. It offers a scalable solution for end-users to detect malicious websites in real-time.

**8. Future Enhancements:**

* Expand dataset size for better generalization.
* Integrate with browsers or email clients for real-time protection.
* Use deep learning models for more complex detection.
* Add support for mobile platforms.