AI-Powered Phishing URL Detection System (Sample Implementation)

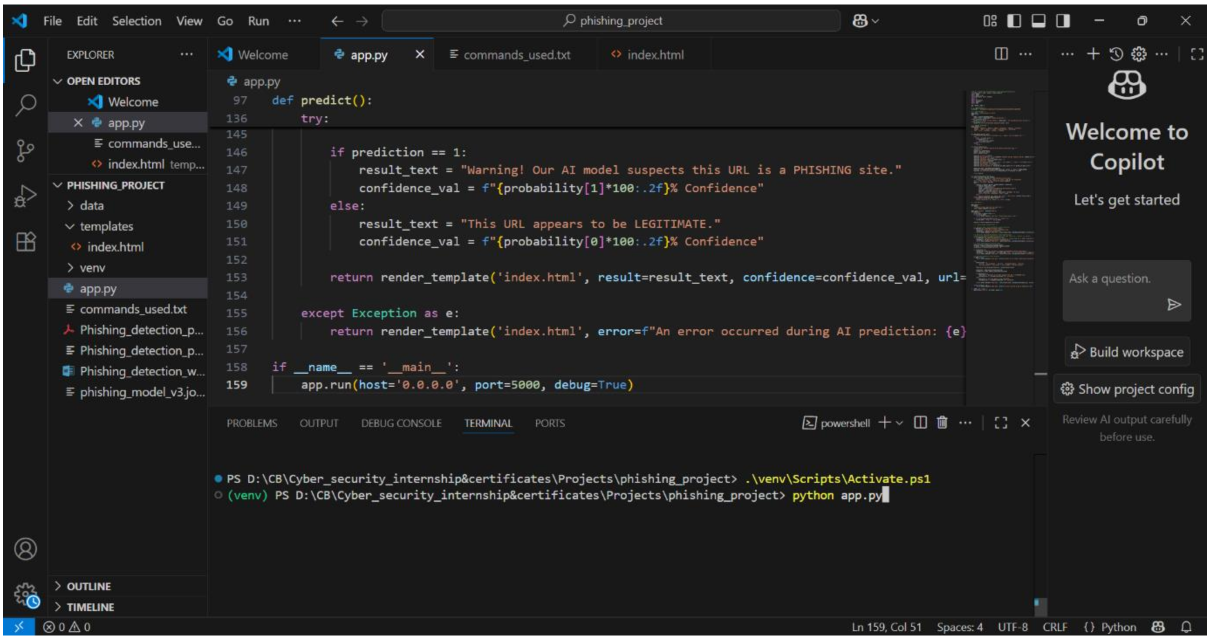
1. Introduction

Phishing attacks remain one of the most prevalent and effective cyber threats, tricking users into revealing sensitive information by impersonating legitimate websites. This project addresses this challenge by developing an intelligent, web-based application designed to detect and flag phishing URLs in real-time. The primary goal is to provide a user-friendly tool that leverages a hybrid approach of machine learning, heuristic analysis, and whitelist verification to assess the safety of a given web link, thereby enhancing user security against online fraud.

2. Project Description

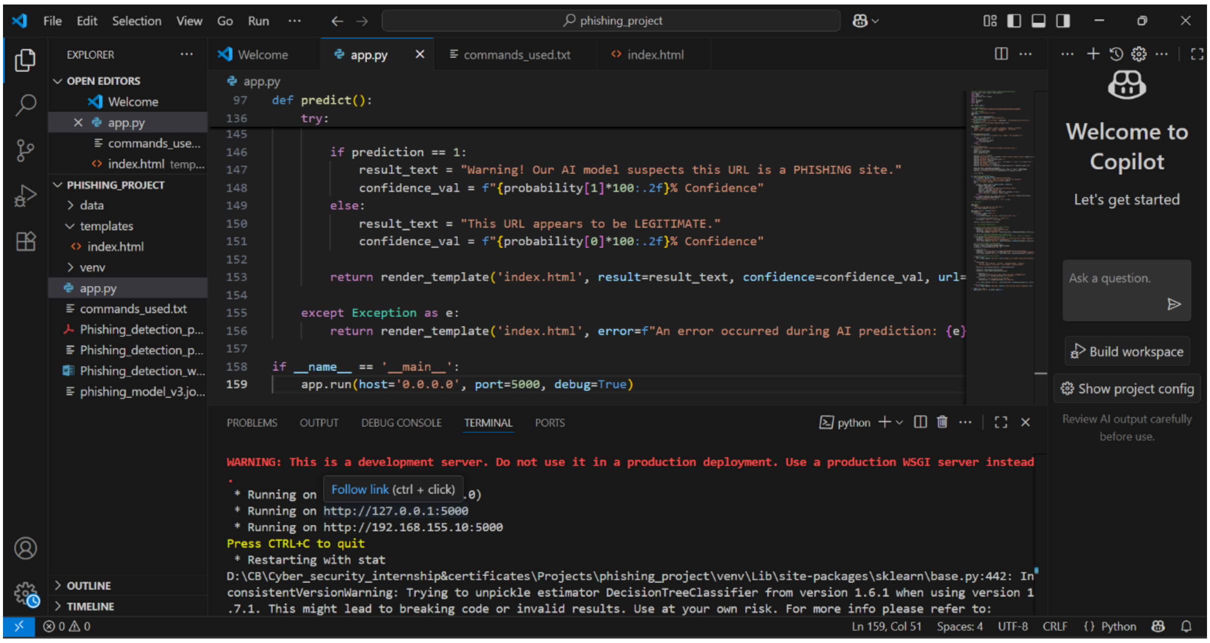
The project is a full-stack web application built using Python and the Flask framework. The system allows a user to submit a URL through a simple web interface, which is then processed by a sophisticated backend to determine its legitimacy. The development and execution process, as seen in the provided images, involves the following key components:

- **Backend Framework:** The application is built on **Flask**, a lightweight Python web framework. The app.py script serves as the core of the application, handling web requests, processing data, and rendering results. The application is run locally on a development server (127.0.0.1:5000).



```python
def predict():
    try:

            if prediction == 1:
                result_text = "Warning! Our AI model suspects this URL is a PHISHING site."
                confidence_val = f"{probability[1]*100:.2f}% Confidence"
            else:
                result_text = "This URL appears to be LEGITIMATE."
                confidence_val = f"{probability[0]*100:.2f}% Confidence"

            return render_template('index.html', result=result_text, confidence=confidence_val, url=

        except Exception as e:
            return render_template('index.html', error=f"An error occurred during AI prediction: {e}

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

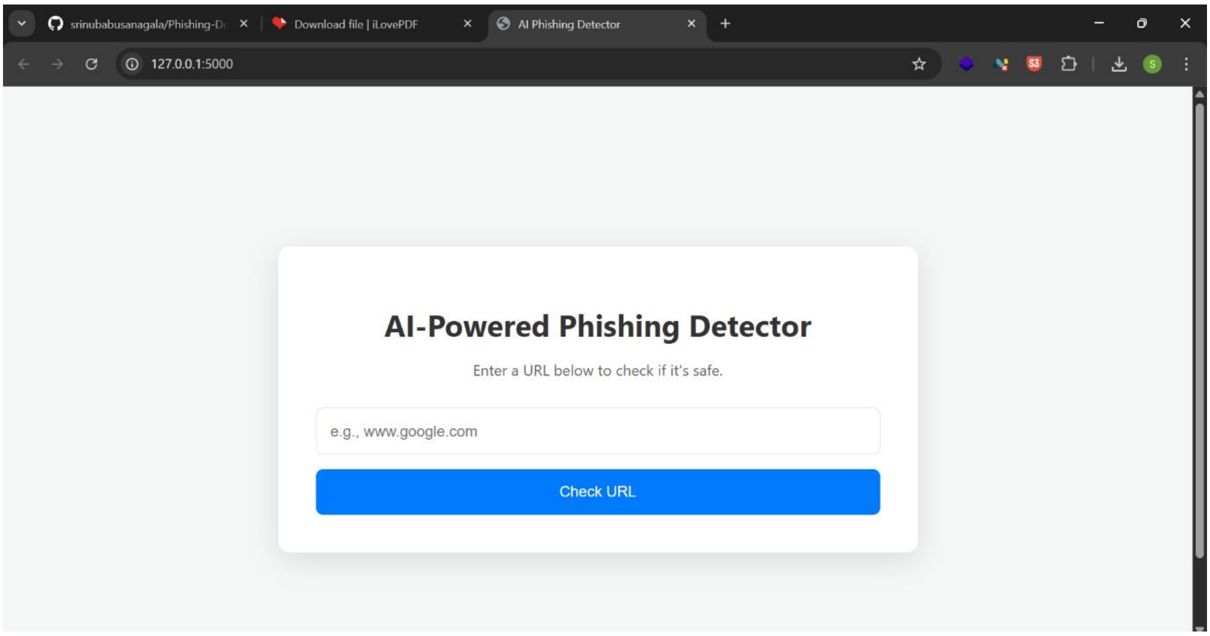
```
PS D:\CB\Cyber_security_internship&certificates\Projects\phishing_project> .\venv\Scripts\Activate.ps1
(venv) PS D:\CB\Cyber_security_internship&certificates\Projects\phishing_project> python app.py
```



```python
def predict():
    try:

        if prediction == 1:
            result_text = "Warning! Our AI model suspects this URL is a PHISHING site."
            confidence_val = f"{probability[1]*100:.2f}% Confidence"
        else:
            result_text = "This URL appears to be LEGITIMATE."
            confidence_val = f"{probability[0]*100:.2f}% Confidence"

        return render_template('index.html', result=result_text, confidence=confidence_val, url=

    except Exception as e:
        return render_template('index.html', error=f"An error occurred during AI prediction: {e}

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
.
 * Running on                            .0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.155.10:5000
Press CTRL+C to quit
 * Restarting with stat
D:\CB\Cyber_security_internship&certificates\Projects\phishing_project\venv\Lib\site-packages\sklearn\base.py:442: In
consistentVersionWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.6.1 when using version 1
.7.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
```

- **Machine Learning Model:** The system integrates a pre-trained machine learning model (likely a DecisionTreeClassifier from the scikit-learn library, as indicated by the terminal warning). This model analyzes various features of a URL to predict whether it is malicious. The model is saved as a file (phishing_model_v3.joblib) and loaded by the Flask application to make live predictions.
- **Hybrid Detection Engine:** The application does not rely solely on the AI model. It employs a multi-layered detection strategy for higher accuracy:
    - **Whitelist Verification:** It checks the URL against a list of known, legitimate brands. As shown with the amazon.in URL, this results in a 100% confidence score, confirming it as a "Verified Brand."
    - **Heuristic Rules:** The system incorporates rules to detect common phishing techniques. The example of amazonn.in triggers a "Typosquatting" rule, correctly identifying it as a highly suspicious imitation of a known brand.
    - **AI-Based Prediction:** For URLs that are not on the whitelist or caught by heuristic rules, the AI model performs a deeper analysis to classify them as either "PHISHING" or "LEGITIMATE," assigning a confidence score based on the model's prediction probability.
- **Frontend Interface:** The user interacts with a clean and intuitive web page created with HTML/CSS (index.html located in the templates folder). This interface features a simple input field for the URL and a button to initiate the check, with the results displayed clearly on the same page.



AI-Powered Phishing Detector

Enter a URL below to check if it's safe.

---

3. Key Features

- **Real-Time URL Analysis:** Provides immediate feedback on the safety of a submitted URL.
- **Intuitive Web Interface:** A simple, user-friendly design ensures accessibility for non-technical users.
- **AI-Powered Detection:** Utilizes a trained machine learning model to identify complex phishing patterns that are difficult to spot manually.
- **Heuristic-Based Checks:** Implements rules to identify common phishing tactics like typosquatting, enhancing detection capabilities.
- **Confidence Scoring:** Quantifies the prediction by providing a confidence percentage, helping users gauge the level of risk.
- **Clear and Actionable Results:** Displays results in an easy-to-understand format (e.g., green for safe, red for warning) with a brief explanation.

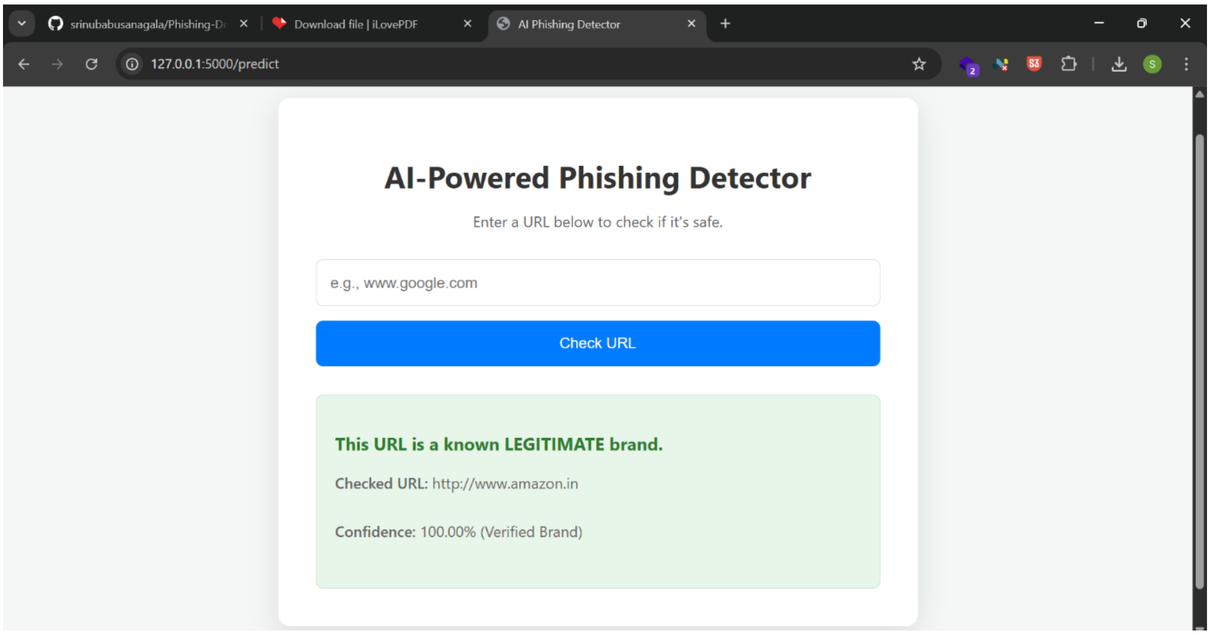
---

4. Results & Analysis

The screenshots demonstrate the system's effectiveness in handling different types of URLs:

- **Legitimate URL Test:**
  - **Input:** http://www.amazon.in

- **Output:** "This URL is a known LEGITIMATE brand."
- **Analysis:** The system correctly identified a well-known, legitimate domain, likely through its whitelist, and assigned it the highest confidence.



AI-Powered Phishing Detector

Enter a URL below to check if it's safe.

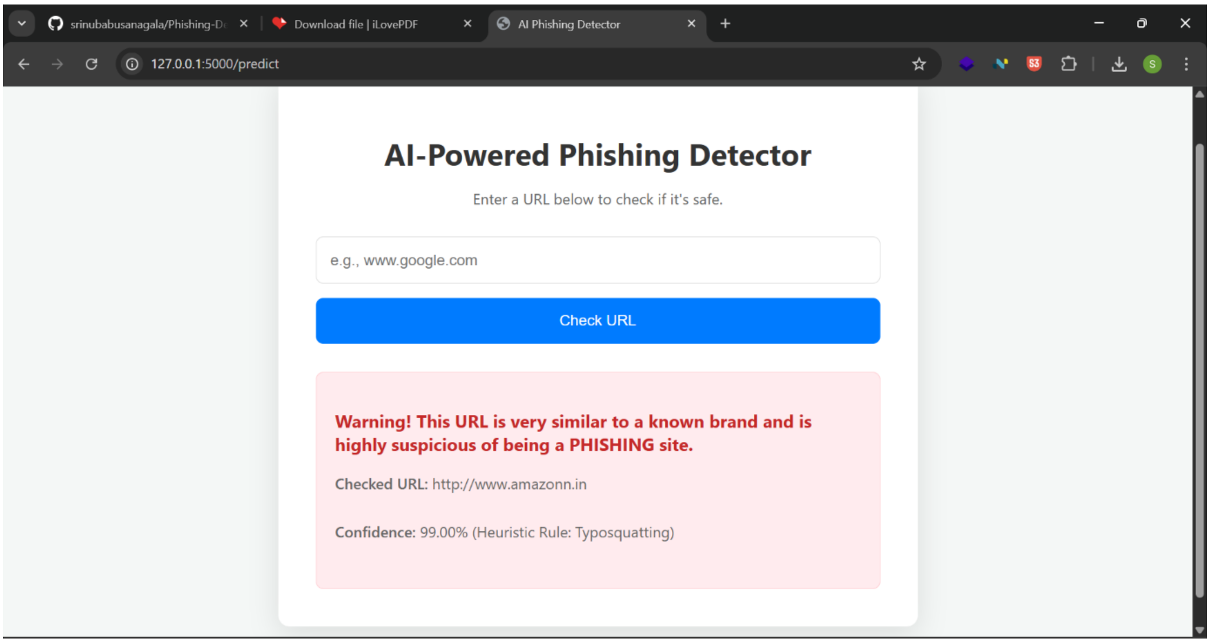
This URL is a known LEGITIMATE brand.

Checked URL: http://www.amazon.in

Confidence: 100.00% (Verified Brand)

- **Suspicious (Typosquatting) URL Test:**
  - **Input:** http://www.amazonn.in
  - **Output:** "Warning! This URL is very similar to a known brand and is highly suspicious of being a PHISHING site."
  - **Analysis:** The application successfully detected a classic typosquatting attempt. The high confidence score (99.00%) and the specific reason ("Heuristic Rule: Typosquatting") showcase the strength of its rule-based engine.

These results validate the project's hybrid approach, which allows for both high accuracy in identifying known entities and robustness in catching common, deceptive attack patterns.



AI-Powered Phishing Detector

Enter a URL below to check if it's safe.

e.g., www.google.com

Warning! This URL is very similar to a known brand and is highly suspicious of being a PHISHING site.

Checked URL: http://www.amazonn.in

Confidence: 99.00% (Heuristic Rule: Typosquatting)

---

5. Conclusion

This AI-Powered Phishing Detector project successfully combines machine learning with other security techniques to create a practical and effective tool for combating phishing threats. It serves as a strong proof-of-concept for a system that can be further developed into a robust security solution, such as a browser extension or an integrated API for email clients. The project effectively demonstrates how modern AI can be applied to solve critical real-world cybersecurity problems, offering users a valuable layer of protection in their daily online activities.