**📝 Phishing URL Detector – Project Report Format**

**📄 1. Title Page**

yaml

CopyEdit

Project Title: Phishing URL Detector using Machine Learning

Submitted by: [Your Name]

Roll Number: [Your Roll No]

Department: [Your Department Name]

College: [Your College Name]

Academic Year: [2024 – 2025]

**📋 2. Abstract**

A short summary (5–6 lines):

This project presents a machine learning-based approach to detect phishing websites using URL analysis. By extracting 19 features from a given URL and applying the XGBoost classification algorithm, the model accurately predicts whether the URL is legitimate or phishing. The application is built using Flask, with MySQL used for data storage, and integrates a trained model stored via Pickle. The system aims to contribute to real-time phishing detection, helping users avoid fraudulent websites.

**📚 3. Introduction**

* What is phishing?
* Why is it dangerous?
* Why machine learning is useful in phishing detection?
* Real-world impact (banking, emails, SMS).

**🧰 4. Technology Stack**

| **Technology** | **Purpose** |
| --- | --- |
| Python | Core programming language |
| Flask | Web framework to build the UI and handle backend |
| XGBoost | Classification algorithm to detect phishing URLs |
| Pickle | For saving/loading the trained model |
| MySQL | Database to store URLs and predictions |
| HTML/CSS | For building frontend templates |

**⚙️ 5. System Architecture**

Include this diagram:

scss

CopyEdit

User Input → Feature Extractor → Trained Model (XGBoost) → Result Display

↓

MySQL Database (Optional Logging)

**🧪 6. Working & Flow**

1. User inputs a URL on the web page.
2. Features are extracted (length, use of '@', special characters, etc.).
3. Features are fed into the pre-trained XGBoost model.
4. Prediction is made (0 = Legit, 1 = Phishing).
5. Result shown on the frontend.
6. (Optional) Data stored in MySQL for analysis.

**📈 7. Dataset Used**

* Dataset contains phishing and legitimate URLs.
* Total features: 19
* Label: phishing (0 or 1)
* Features extracted using Python-based feature extractor.

**🧠 8. Machine Learning Model**

* Algorithm: XGBoost Classifier
* Reason for use:
  + High accuracy
  + Handles overfitting well
  + Works well on tabular data
* Training accuracy: ~[Your Accuracy]%
* Evaluation metrics: accuracy, precision, recall

**🧑‍💻 9. Code Snapshot (Optional)**

Include key code snippets:

* Feature extraction
* Model loading (pickle.load)
* Flask route handling
* Predict function

**📊 10. Results**

| **URL** | **Prediction** |
| --- | --- |
| http://example.com/login | Legitimate |
| <http://secure-update-paypal.com> | Phishing |

Graph (optional): Bar chart or pie chart of phishing vs legit URLs.

**🚀 11. Future Scope**

* Make a browser extension
* Add email or SMS scanner
* Use live blacklists and update model regularly
* Add deep learning (RNN/CNN)
* Dashboard for admin insights

**📚 12. References**

* XGBoost documentation
* Flask official docs
* Kaggle dataset (if used)
* Research papers/articles on phishing detection

**🧾 13. Appendix (Optional)**

* Full code printout
* Model performance graphs
* MySQL schema

Agar tu chahe toh main is full report ko **Word or PDF format** me exportable bana ke de sakta hoon. Chahta hai ki main ek ready-made editable report likh ke du?