# Group 9: Phishing Website Detector Solution

Here’s a modular architecture for the phishing website detection solution, using **MCMC-GAN** for augmentation and **Stacking Ensemble** for classification.

**Phishing Detection System Architecture**

**1. Data Pipeline Layer**

**Input:**

* Raw dataset (CSV or SQL):
  + Features like URL tokens, domain age, SSL cert, JS usage, etc.
  + Label: 0 = legitimate, 1 = phishing

**Preprocessing Steps:**

* Feature engineering (if needed)
* SelectKBest with mutual\_info\_classif → Select Top 30 features
* Save processed dataset

**2. Data Augmentation Layer**

**MCMC-GAN Module:**

* Input: Imbalanced dataset
* Output: Balanced dataset (50% phishing, 50% legitimate)

**Key Components:**

* GAN Generator & Discriminator
* MCMC sampler to improve diversity
* Logging module to record synthetic sample stats

**3. Model Training Layer**

**Stacking Ensemble Model**

**Base Models:**

* Logistic Regression
* Random Forest
* XGBoost

**Meta-learner:**

* Logistic Regression or XGBoost (your choice)

**Training Flow:**

* Stratified train-test split
* 5 or 10-fold cross-validation
* Evaluate with:
  + Accuracy, Precision, Recall, F1-score, ROC-AUC

**4. Evaluation & Reporting Layer**

**Metrics Logged:**

* Training/Test accuracy
* Confusion matrix
* Precision, recall, F1, ROC-AUC
* Runtime per model

**Visualization:**

* ROC curves
* Feature importance plot
* Augmentation effect comparison

**5. Inference / Deployment Layer (Optional for real-time use)**

**Interface:**

* Web API (Flask or FastAPI)
* Input: Website URL (or features)
* Output: Phishing or Legitimate + confidence score

**Hosting:**

* Could be containerized (Docker) and hosted on a cloud platform (AWS/GCP/Heroku)

**Folder Structure Suggestion**

phishing\_detector/

├── data/

│ └── raw\_dataset.csv

├── preprocessing/

│ └── feature\_selector.py

├── augmentation/

│ └── mcmc\_gan.py

├── models/

│ ├── base\_models.py

│ ├── stacking\_model.py

├── evaluation/

│ └── evaluate\_metrics.py

├── deploy/

│ └── api\_server.py

├── reports/

│ └── metrics\_log.csv

├── main.py

**Bonus: Traceable Flow Diagram (Simplified)**

[ Raw Dataset ]

↓

[ Feature Selection (Top 30) ]

↓

[ MCMC-GAN Augmentation ]

↓

[ Balanced Dataset ]

↓

[ Train Stacking Ensemble ]

↓

[ Evaluate ]

↓

[ Save model / Deploy via API ]

