Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	Please Enter the date
Team ID	Please Enter your Team ID
Project Name	Garbage Classification Using Deep Learning
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

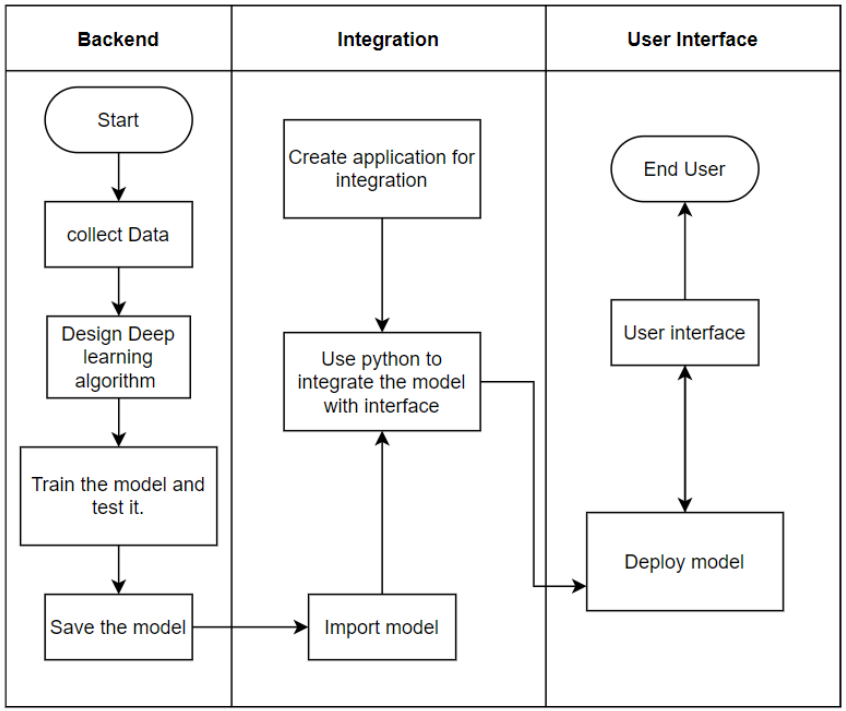

Table-1 : Components & Technologies:

S.NO	Component	Description	Technologies
1.	Data Collection	Gather malware samples and benign files for training	Crawlers, honeypots, virus total APIs
2.	Data Preprocessing	Prepare and clean the collected data	Python (Pandas, NumPy), data cleaning tools
3.	Feature Extraction	Extract relevant features from data	N-grams, byte sequences, API calls, PEinfo
4.	Machine Learning Models	Train models for malware classification	Deep Learning (e.g., CNN, RNN, LSTM), SVM

5.	Feature Selection	Identify important features	Recursive Feature Elimination (RFE), PCA
6.	Model Evaluation	Assess model performance	Cross-validation, ROC, F1 score
7.	Real-time Data Streaming	Handle incoming data for live detection	Apache Kafka, Apache Flink
8.	Anomaly Detection	Detect anomalies in network traffic	Unsupervised learning (e.g., Isolation Forest)
9.	API and Endpoint Monitoring	Observe and analyze behavior at endpoints	Elastic Stack, Sysmon, Suricata
10.	Dynamic Analysis	Execute malware in a controlled environment	Virtualization (e.g., VMware, VirtualBox)

11.	Signature-based Detection	Identify known malware signatures	Snort, ClamAV, YARA rules
12.	Behavioral Analysis	Analyze behavior patterns for malware detection	Cuckoo Sandbox, Sysinternals Suite
13.	Threat Intelligence	Stay updated with the latest threat information	Threat feeds, open-source threat intel tools

Table-2: Application Characteristics:

S.NO	Application Characteristic	Description	Technologies

1.	Real-time Detection	Provides immediate malware identification	Apache Kafka, Storm, real-time data processing
2.	Scalability	Ability to handle large datasets and traffic	Hadoop, Spark, distributed computing platforms
3.	Multi-platform Support	Works across various operating systems	Cross-platform development tools
4.	Adaptive Learning	Self-improves by adapting to new malware threats	Reinforcement learning, online learning
5.	Signature-less Detection	Identifies malware without predefined signatures	Anomaly detection, heuristics, behavioral analysis
6.	Cloud Integration	Utilizes cloud resources for processing	AWS, Azure, Google Cloud, cloud computing services

7.	Threat Intelligence	Integrates external threat information	Threat intelligence feeds, APIs, open-source data
8.	Cross-layer Analysis	Analyzes network, endpoint, and file data	Network traffic analysis, endpoint monitoring
9.	Explainability	Provides explanations for detection decisions	Explainable AI techniques, model interpretability
10.	Compliance Monitoring	Ensures adherence to security regulations	Compliance auditing tools, regulatory frameworks
11.	Reporting and Logging	Generates reports and logs for analysis	SIEM solutions, ELK Stack, log management tools
12.	Automatic Updates	Keeps the system up to date with new threats	Automated model training, continuous integration

13.	Integration with SOAR	Links with Security Orchestration Automation	SOAR platforms (e.g., Palo Alto Cortex XSOAR)

References:

https://c4model.com/ **https://www.leanix.net/en/wiki/ea/technical-architecture**

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d