

PROJECT DESIGN PHASE-II TECHNOLOGY STACK (ARCHITECTURE & STACK)

Date	23 October 2022
Team ID	PNT2022TMID21985
Project Name	WEB PHISHING DETECTION
Maximum Marks	4 Marks

TECHNICAL ARCHITECTURE:

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

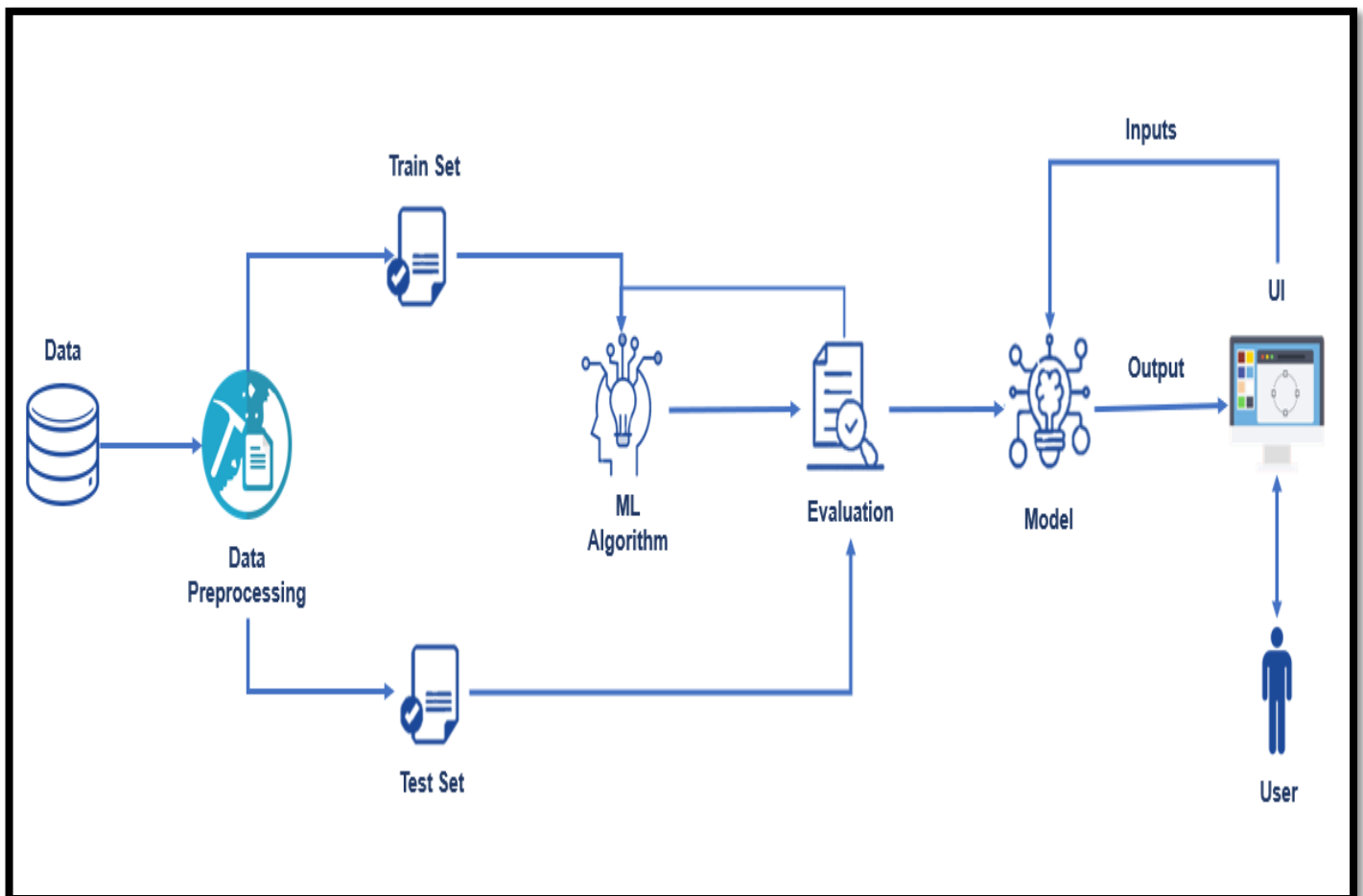


Table-1: Application Components:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI etc.	HTML, CSS, Python flask etc
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL etc.
6.	Cloud Database	Database Service on Cloud	IBM Database etc.
7.	File Storage	File storage requirements	IBM Cloud Object Storage
8.	External API-1	Purpose of External API used in the application	IBM API Connect, etc.
9.	External API-2	Purpose of External API used in the application	Nil
10.	Machine Learning Model	Purpose of Machine Learning Model	IBM Watson Studio
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local System, IBM Watson etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python Flask, Jupyter Notebook and Python Libraries etc
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. Anti-Phishing Protection and Anti-Spam Software etc
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Python Libraries
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	IBM Watson Machine Learning
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Flask