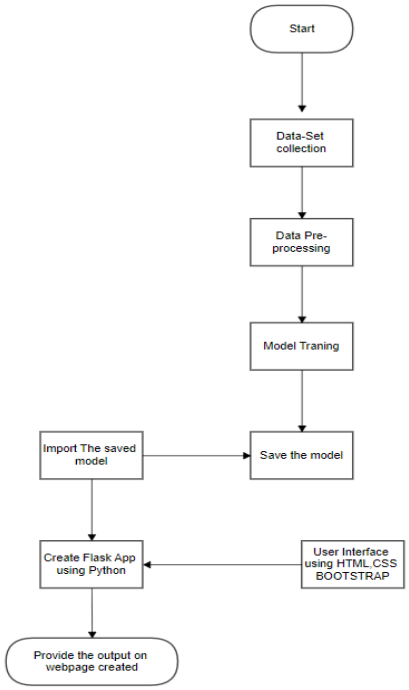
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	27-10-2023
Team ID	PNT2022TMID592946
Project Name	Project – Travel Insurance Predication using Machine Learning
Maximum Marks	4 Marks

Technical Architecture:



Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Database	Collect the Dataset Based on the Problem Statement	File Manager, MySQL, NoSQL, etc.
4.	File Storage/ Data	File storage requirements for Storing the dataset	Local System, Google Drive Etc
5.	Frame Work	Used to Create a web Application, Integrating Frontend and Back End	Python Flask, Django etc
6.	Machine Learning Model	Purpose of Model	CNN, Transfer Learning etc.
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilize open-source frameworks for development, machine learning, and data analysis.	Python's Flask, Scikit-Learn
2.	Security Implementations	Implement security measures to protect data and user interactions within the application.	SSL/TLS, Encryption, Authentication.
3.	Scalable Architecture	Design the architecture to be scalable, allowing the application to handle growing data and user loads.	Cloud Services (e.g., AWS Auto Scaling), Load Balancing
4.	Availability	Ensure high availability of the application, minimizing downtime and disruptions	Redundancy, Failover, Monitoring and Alerting
5.	Performance	Optimize application performance for responsiveness and efficient use of resources	Caching, Database Indexing, Efficient Algorithms

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