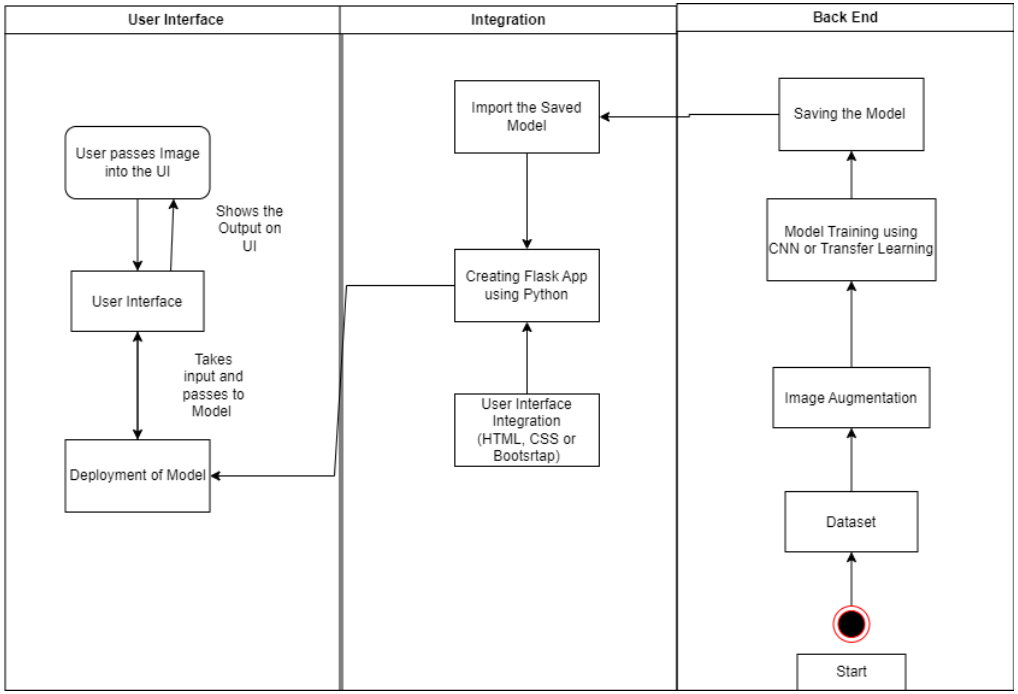
Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	27-10-2023
Team ID	Team-593009
Project Name	AI Enable car parking using OpenCV
Maximum Marks	4 Marks

Technical Architecture:

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



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, Bootstrap
2.	Application Logic-1	Logic for a process in the application	Python
3.	Database	Collect the Dataset Based on the Problem Statement	File Manager
4.	File Storage/ Data	File storage requirements for Storing the dataset	Local System, Google Drive
5.	Frame Work	Used to Create a web Application, Integrating Frontend and Back End	Python Flask
6.	Deep Learning Model	Purpose of Model	OpenCV, CNN
7.	Infrastructure	Application Deployment on Local System	Local Server

Table: Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python's Flask
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	e.g., SHA-256, Encryptions, IAM Controls, OWASP etc.

3.	Scalable Architecture	Justify the scalability of architecture (3 – tier,Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g., use ofload balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Python's Flask or any basic web framework	Python's Flask
2.	Security Implementations	Basic encryption for data transmission, simple access controls	Basic Encryption (e.g., HTTPS), Basic Access Control
3.	Scalable Architecture	Simple, non-distributed architecture	Single-Server Setup
4.	Availability	Local server setup for development and testing	Local Server
5.	Performance	Basic performance considerations for local use	Basic Caching Strategies, Local Testing