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

Date	15 November 2022
Team ID	Team-591796
Project Name	Project - 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

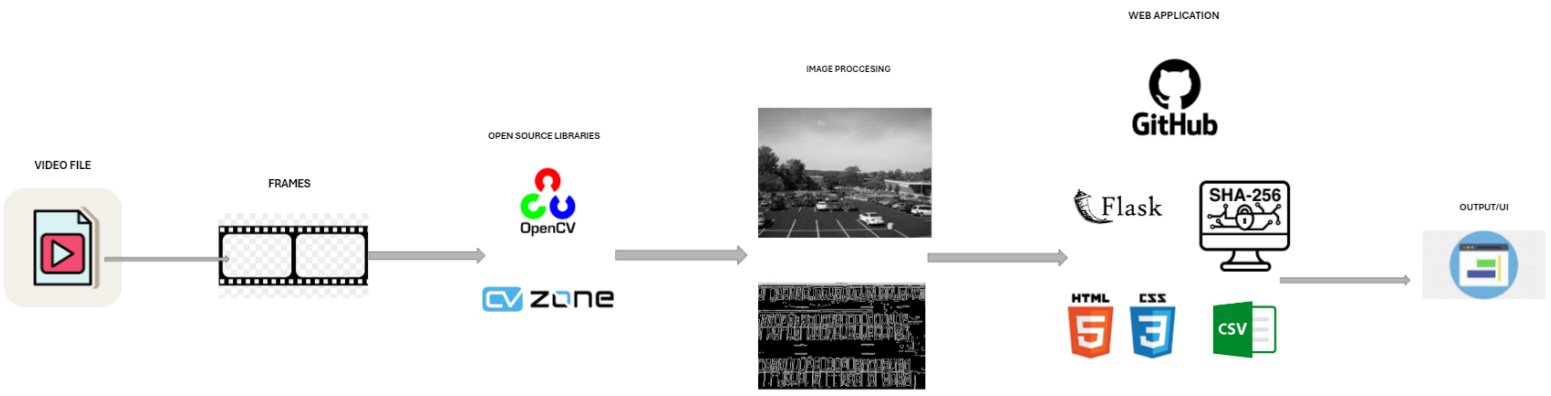

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web UI rendering	PYTHON ,FLASK
2	Video file reading	Video feed and parking space visualization	OpenCV, cvzone
3	Application Logic-1	Parking space occupancy analysis	Python
4	Application Logic-2	File storage and loading of parking positions	Python, pickle
5	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud:	Local Server Configuration, Cloud Server Configuration(GitHub)
6	Database	Data storage for parking space positions	Pickle file
7	User Interface	Front-end markup language for web interface & Styling language for web interface	HTML, CSS,
8	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
9	Login Credentials	Storage of user login information	CSV File
10	Password Encryption	Encryption method for password storage	SHA-256

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilization of open-source frameworks for image processing and UI development	OpenCV, cvzone, HTML, CSS,Flask
2.	Security Implementations	Implementation of SHA-256 encryption for password storage and access control for parking space information	Python (hashlib library for SHA-256), Authentication mechanisms

3.	Scalable Architecture	The architecture is designed for scalability through modularized image processing logic, enabling adaptability to varying parking lot sizes and configurations	Python (Modular code structure)
4.	Availability	Ensured availability through the use of load balancers and distribution across multiple servers, minimizing downtime and enhancing fault tolerance	Load balancers (Not explicitly implemented in the provided code, but suggested for availability)
5.	Performance	Designed for optimal performance, considering factors such as requests per second, implementing caching mechanisms, and utilizing Content Delivery Networks (CDNs) for efficient content delivery	Caching (Not explicitly implemented in the provided code), Efficient algorithm design