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

Date	21 November 2023
Team ID	Team-592184
Project Name	ASL-Alphabet Image Recognition
Maximum Marks	4 Marks

Technical Architecture:

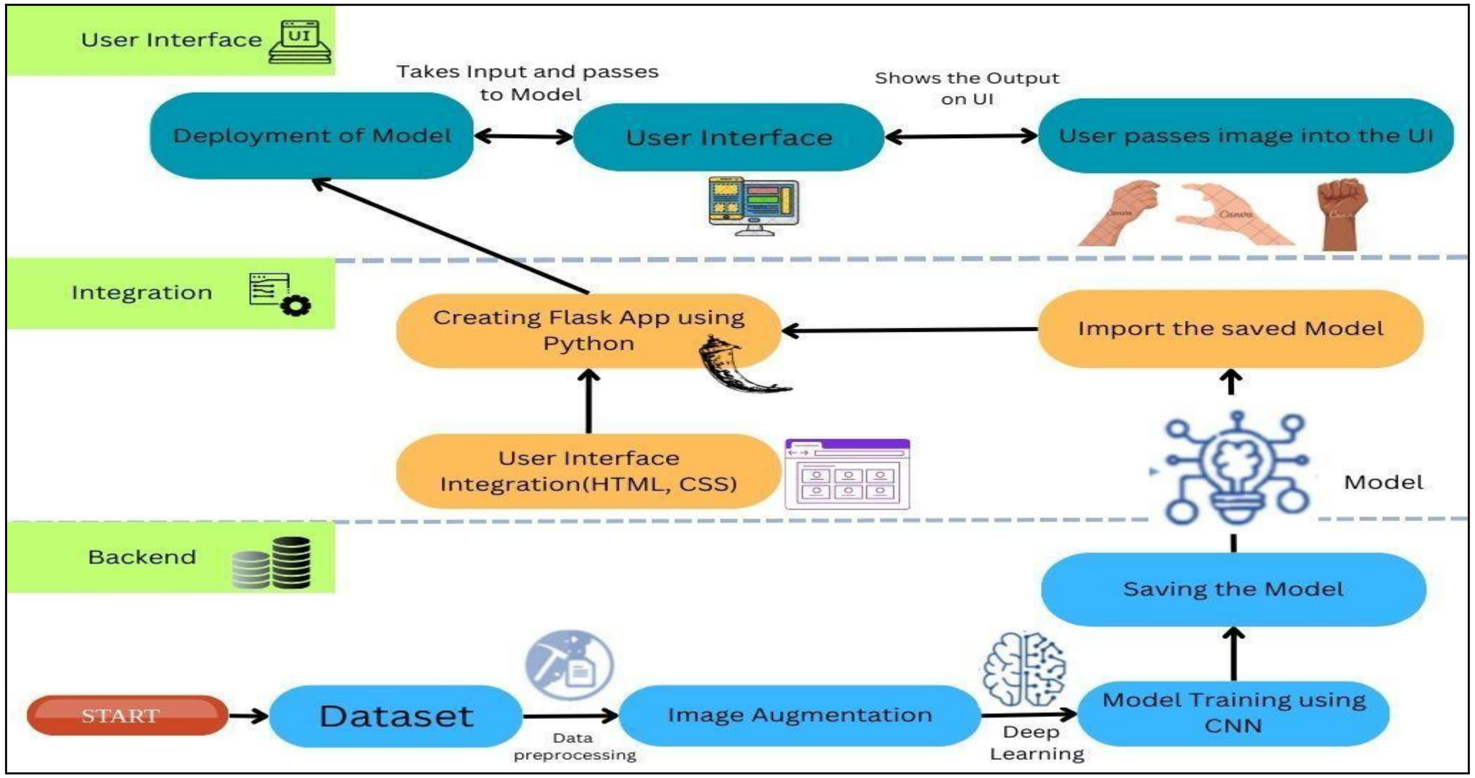

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Image Acquistion	Acquire or gather pictures depicting American Sign Language (ASL) alphabet signs.	Image databases,smart phones or Digital cameras.
2.	Preprocessing	Preprocess the images for recognition by applying a range of transformations and enhancements.	Open CV(Computer Vision Library),python.
3.	Image Segmentation	Isolate hand signs from the background and, if required, identify individual fingers for enhanced recognition.	Open CV, Image processing Techniques.
4.	Feature Extraction	Capture relevant characteristics from the segmented images, encompassing aspects like shape, color, and texture.	Feature extraction algorithms (e.g., Histogram of Oriented Gradients, Color Histograms), Python.
5.	Machine Learning Model	Develop a machine learning model capable of identifying ASL alphabet signs using features extracted from the data.	Tensor Flow, PyTorch, Scikit-Learn, Keras, or a custom model using deeplearning or traditional machine learning algorithms.
6.	Training Data	A dataset of labeled ASL alphabet sign imagesfor model training.	ASL image datasets, data augmentation techniques.

7.	Model Evaluation	Evaluate the model's performance by examining metrics such as accuracy, precision, recall, F1 score, and other pertinent indicators.	Cross-validation, evaluation metrics in Python.
8.	Model Deployment	Deploy the trained model for the real-time or batch processing of ASL signs in a deployed environment.	Cloud platforms (e.g., AWS, Azure, GCP), web servers, APIs.
9.	User Interface	Design a user-friendly interface that allows users to interact effortlessly with the ASL alphabet recognition system.	Web development (HTML, CSS, JavaScript), mobile app development(e.g., React Native, Flutter).
10.	Integration With Sign Language Interpreter	Integrate the ASL recognition system seamlessly with a sign language interpreter to offer translations or responses.	APIs, libraries for natural language processing (NLP).
11.	Infrastructure (Server / Cloud)	Consistently enhance and refine the system through the collection of user feedback, ensuring regular updates to improve the model.	Agile development practices, versioncontrol (e.g., Git).

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Using open-source frameworks can greatly expedite development, cut down costs, and take advantage of a collaborative community for mutual benefits.	Python for machine learning and image processing (NumPy, OpenCV,Scikit-Learn),

			TensorFlow or PyTorchfor deep learning, and Flask or Django for web application development.
2.	Security Implementations	Ensuring the security of both user data and the system is paramount. Implement a range of security measures to safeguard against data breaches and unauthorized access.	SSL/TLS for secure data transmission and implementencryption for data at rest.
3.	Scalable Architecture	Architecting the system to accommodate growing loads and user demands by scaling horizontally or vertically as required.	Docker for packaging applications andKubernetes for container orchestration, Auto-scaling on cloud platforms to dynamically allocate resources based on demand, tools like Nginx or HAProxy for distributing traffic across multiple instances.
4.	Availability	Ensuring that the system is always accessible and minimizes downtime.	Setting up failover mechanisms and replicate critical components for high availability, tools likeAWS CloudWatchto monitor system health and performance. Implement backup and recovery strategies to restore the system incase of failures.Use CDNs to distribute content andreduce latency.

5.	Performance	Enhance system performance by optimizing for rapid response times and efficient utilization of resources.	Implement caching mechanisms (e.g.,Redis, Memcached) for frequently accessed data. Tools like Python's cProfile to identifybottlenecks.Apache Spark or Hadoop fordistributed data processing.