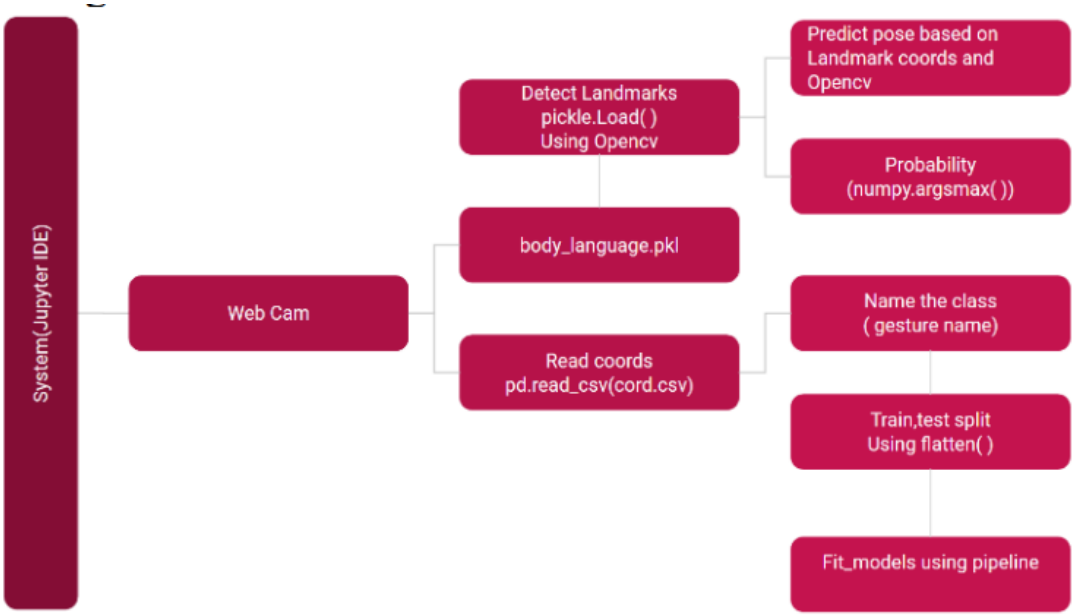
Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	19 November 2023
Team ID	Team-591988
Project Name	Project - AI Body Language Detector Using Media pipe
Maximum Marks	10 Marks

Technical Architecture:

- **Flowchart of working model using IDE:**



- **Overview of technical Architecture:**

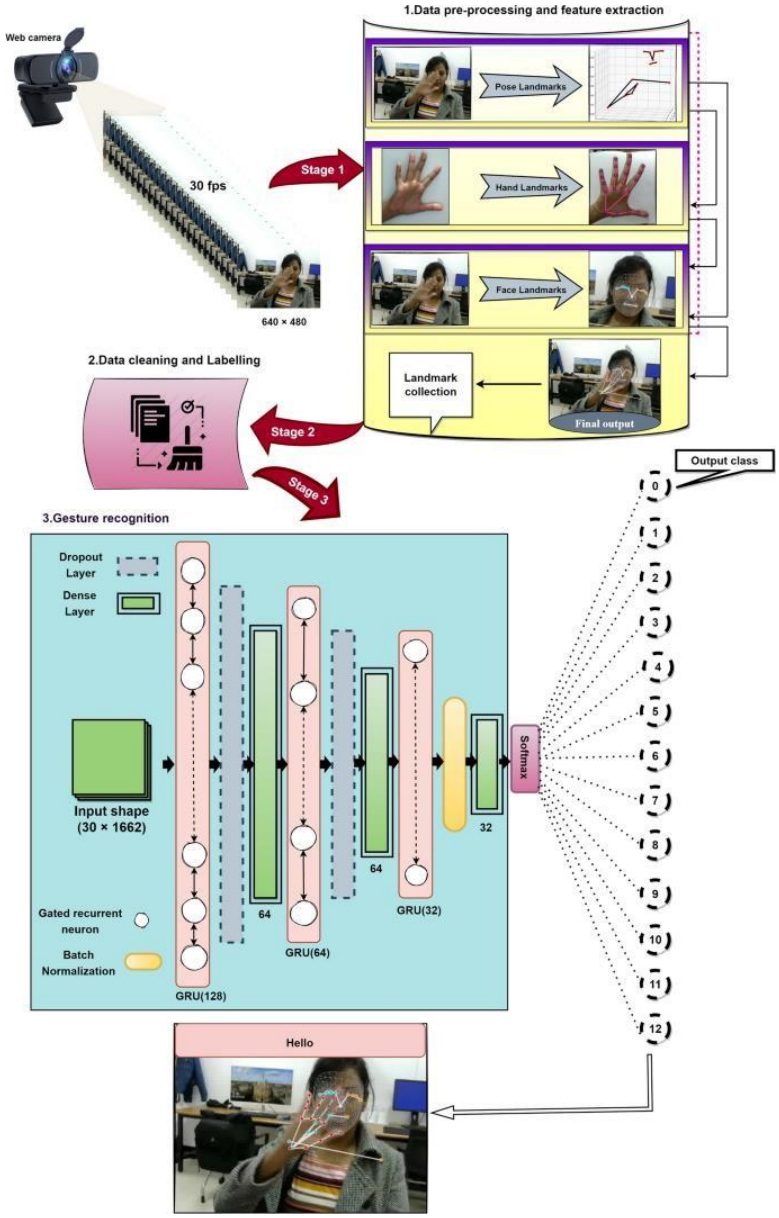


Table-1: Components & Technologies:

Component	Description	Technology
Mediapipe Integration	Core library for body language detection.	Mediapipe
Backend Framework (Optional)	Handles server-side logic and communication.	Flask or Django
Frontend Framework (Optional)	Provides the user interface for interaction.	React or Vue.js
Real-Time Communication (Optional)	Enables real-time updates between server and client.	WebSockets
Database (Optional)	Stores historical body language data.	SQLite, PostgreSQL, or MongoDB

Cloud Services (Optional)	Provides hosting, storage, and other cloud-based services.	Google Cloud Platform (GCP) or AWS
Containerization (Optional)	Packages the application and its dependencies.	Docker
Version Control	Manages code versions and facilitates collaboration.	Git
CI/CD Tools	Implements automated testing and deployment pipelines.	Jenkins, GitLab CI, or GitHub Actions
Integrated Development Environment (IDE)	Supports code writing and testing.	PyCharm, VS Code, or Jupyter Notebooks
RESTful API (Optional)	Facilitates communication with other services.	RESTful API

Table-2: Application Characteristics:

Component	**Description**	**Technology**
Real-time Analysis	Provides instant analysis of body language gestures and poses.	Mediapipe, WebSockets (for real-time updates)
User Interface (Optional)	Offers a user-friendly interface for interaction and feedback.	React or Vue.js (Frontend Framework)
Scalability (Optional)	Scales to handle increased load and user base.	Google Cloud Platform (GCP) or AWS (Cloud Services)
Data Storage and Retrieval (Optional)	Stores historical body language data for analysis and learning.	SQLite, PostgreSQL, or MongoDB (Database)
Cross-platform Compatibility	Runs seamlessly on various devices and platforms.	Python (Mediapipe is platformindependent)
Integration with Other Services (Optional)	Allows communication with external services.	RESTful API (for integration)
Security Considerations	Implements security measures to protect user data and privacy.	Secure coding practices, SSL/TLS for data transmission
Containerization (Optional)	Facilitates deployment and management across different environments.	Docker (Containerization)
Continuous Monitoring (Optional)	Monitors application performance and user interactions.	Application Performance Monitoring (APM) tools
Continuous Integration/Continuous Deployment (CI/CD)	Enables automated testing and deployment.	Jenkins, GitLab CI, or GitHub Actions (CI/CD tools)
Documentation and Training Materials	Provides comprehensive documentation and training resources.	Wiki, Documentation Platforms
Accessibility (Optional)	Ensures accessibility for users with disabilities.	Compliance with accessibility standards