Artificial Intelligence for Business  
Decisions and Transformation

**Team Members:**

**Angel Antony**

**Arya Krishnan**

**Prasant Pradeep**

**1. Create Operational Protocols (SOPs)**

The following Standard Operating Procedures (SOPs) will ensure consistency, efficiency, and quality in executing tasks for the project:

**Data Collection and Preprocessing SOP:**

* **Objective**: Collect and preprocess data for training and testing the LSTM model.
* **Steps**:
  + Use **OpenCV** to capture live video data or utilize existing datasets for sign language gestures.
  + Apply **MediaPipe** to detect keypoints of hands, face, and body for gesture recognition.
  + Use **Numpy** to preprocess the extracted keypoint data for training the model, cleaning the data, and normalizing it.
  + Augment the data by flipping, rotating, or scaling to increase dataset diversity.
  + Store the preprocessed data in an organized format to ensure easy retrieval during model training.

**Model Training SOP:**

* **Objective**: Train the LSTM model using TensorFlow for gesture recognition.
* **Steps**:
  + Train the **LSTM model** using **TensorFlow** on preprocessed data containing gesture sequences.
  + Regularly monitor key metrics like **accuracy, loss, and response time** using **TensorBoard**.
  + Adjust hyperparameters such as learning rate and batch size to optimize performance.
  + Save model checkpoints and log files to track progress and revert changes when needed.

**System Integration SOP:**

* **Objective**: Seamlessly integrate the LSTM model with real-time gesture recognition and the user interface.
* **Steps**:
  + Use **OpenCV** to capture real-time video, which is processed for gesture detection.
  + Pass the processed keypoints from **MediaPipe** to the **LSTM model** in **TensorFlow** for real-time gesture recognition.
  + Ensure that the model output is integrated with the user interface to provide immediate feedback.
  + Test the complete system (video feed, model inference, and UI integration) for any performance bottlenecks or delays.

**2. Allocate Resources**

**Team Roles and Responsibilities**:

**Arya Krishnan – Data Collection and Preprocessing Lead:**

* **Responsibilities**:
  + Collect data using **OpenCV** and clean it using **Numpy**.
  + Apply augmentation techniques to increase data variation.
  + Use **MediaPipe** to extract keypoints for further processing.
  + Format the data for TensorFlow model training.

**Angel Antony – Model Training and Optimization Lead:**

* **Responsibilities**:
  + Train the **LSTM model** using **TensorFlow** and optimize it for real-time gesture recognition.
  + Monitor model performance using **TensorBoard**, ensuring accuracy and low response times.
  + Fine-tune hyperparameters to achieve the best model performance.

**Prasant Pradeep – UI/UX and System Integration Lead:**

* **Responsibilities**:
  + Design the user interface and integrate it with the gesture recognition model.
  + Ensure smooth integration of the real-time video feed from **OpenCV** and the model output for gesture detection.
  + Perform usability testing and refine the interface based on user feedback.

**Software Resources**:

* **Software**:
  + **TensorFlow**: For training and deploying the LSTM model.
  + **OpenCV**: For capturing and processing real-time video.
  + **MediaPipe**: For detecting hand, face, and body landmarks.
  + **Numpy**: For numerical processing and data management.

**3. Build Templates or Tools**

Instead of using Google Sheets, the following tools and templates will be managed via **Azure DevOps Boards**:

**Daily Task Tracking via Azure DevOps Boards:**

* **Azure DevOps Boards** will be used to create user stories, tasks, and sub-tasks for each element of the project.
  + Each task will have:
    - **Task description**
    - **Status (To Do, In Progress, Done)**
    - **Priority**
    - **Assignee**: (Arya, Angel, Prasant)
    - **Issues/Blockers**
    - **Next steps**
  + Tasks will be categorized by sprint and linked to specific project milestones.

**Performance Monitoring and Automation:**

* **Azure Pipelines** or **Azure DevOps Dashboards** will be set up to track project performance:
  + **Model accuracy and loss** will be monitored daily via TensorBoard logs integrated into **Azure Pipelines**.
  + **System response time** will be logged during real-time testing and reported on the dashboard.
  + The **dashboard** will display overall progress, real-time performance metrics, and current blockers.

**UI/UX Testing Checklist:**

* **Azure Boards** will contain an **UI/UX improvement checklist**:
  + Each task related to UI feedback will be added as a task or sub-task.
  + Metrics such as **responsiveness** (lag time, refresh rate) and **usability** (user feedback, accessibility features) will be tracked.
  + Changes made to the interface will be logged as updates in each sprint.

**4. Prototype or Mockup Solutions**

To validate the system’s workflow and operational decisions before full-scale deployment, we will develop the following prototypes and manage them in Azure DevOps:

**Data Preprocessing Prototype:**

* **Format**:
  + A **Python script** using **Numpy** and **MediaPipe** will simulate the data preprocessing workflow. This script will clean the raw video data, apply data augmentation techniques, and extract the necessary keypoints using MediaPipe for gesture recognition.

**LSTM Model Training Prototype:**

* **Format**:
  + A **TensorFlow prototype** will be developed to demonstrate the basic functionality of the LSTM model using a small subset of data. This will allow the team to quickly identify issues with data handling and model performance before scaling up.
  + **Azure Pipelines** will be used to automate the training and testing of this prototype.

**UI Mockup:**

* **Format**:
  + Use **Figma** or **Adobe XD** to create a prototype for the user interface, showing how recognized gestures will be displayed in real-time. Feedback from early tests will be logged as tasks in Azure DevOps to guide further development.

**5. Prepare for Integration**

To ensure seamless integration of the operational design with the overall system and any external systems, the following integration plan will be managed via Azure DevOps:

**Real-Time Video Feed and Model Integration:**

* **Objective**: Ensure smooth integration of **OpenCV**, **MediaPipe**, and the LSTM model in **TensorFlow**.
* **Steps**:
  + Capture real-time video using **OpenCV**, process it with **MediaPipe** to extract keypoints, and feed them into the trained LSTM model.
  + Perform regular testing to ensure minimal lag between the video input and the system’s prediction output, ensuring real-time performance.
  + Use **Azure Pipelines** to automate the real-time video feed testing and logging results.

**UI and Model Output Integration:**

* **Objective**: Ensure the system provides immediate and accurate feedback to users through the UI.
* **Steps**:
  + Connect the output from the LSTM model directly to the user interface.
  + Ensure the UI updates in real-time as gestures are recognized.
  + Test the system for performance issues, especially lag in updating the recognized gestures on the interface.
  + Log all integration issues in **Azure Boards** and link them to related tasks.

**Strategic Alignment:**

* **Scalability**: Design the system to be scalable across various platforms (web, mobile, desktop), allowing for easy integration with existing solutions or applications.
* **Adaptability**: Ensure the system can accommodate future additions, such as new gestures, languages, or features, without significant architectural changes.

**Conclusion and Next Steps**

1. **SOP Development**: Implement the standard operating procedures for data collection, model training, and system integration.
2. **Resource Allocation**: Ensure team members have clear roles and access to the necessary software and hardware.
3. **Azure DevOps Boards**: Use Azure DevOps Boards for detailed task management, tracking progress, assigning tasks, and managing blockers.
4. **Prototyping**: Develop prototypes for data preprocessing, model training, and the user interface, managing them via Azure Pipelines for automation.
5. **System Integration Preparation**: Use Azure DevOps to manage integration testing, real-time video feed, model predictions, and UI interactions. Ensure seamless communication and future scalability of the system.