## **Project Title:**

**TRANSLINGUA:AI Powered Multi-Language Translator**

## **Team Name:**

Techno Divas

## **Team Members:**

* V. Spoorthi Reddy
* K. Swabhavika
* S. Keerthana
* K. Sanjana

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

TransLingua aims to bridge language barriers by providing real-time translation across multiple languages. It enhances communication by supporting various input formats, including text, voice, and PDFs, ensuring accessibility for diverse users.

### **Key Points:**

1. **Problem Statement:**

* Language barriers create communication challenges in education, business, healthcare, and daily life.
* Existing translation tools often lack multi-format input support, leading to accessibility issues for users needing voice or document-based translation.

1. **Proposed Solution:**
   * TransLingua is an AI-driven translation system that:
   * Supports real-time multilingual translation.
   * Accepts text, voice, and document inputs (PDFs).
   * Uses advanced AI models for context-aware translation.
   * Ensures user-friendly accessibility through web and mobile platforms.
2. **Target Users:**
   * Travelers and tourists
   * Students and educators
   * Businesses and professionals
   * Content creators and researchers
3. **Expected Outcome:**
   * Seamless multilingual communication
   * Enhanced accessibility and inclusivity
   * Improved efficiency in global collaboration
   * AI-driven, context-aware translations for accurate communication

## **Phase-2: Requirement Analysis**

### **Objective:**

Define the technical and functional requirements for the Auto Sage App.

### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Frontend: **Vite, React, javaScript**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   * Voice input for real-time translation.
   * Text input for instant translation.
   * PDF or image uploads for text extraction and translation.
3. **Constraints & Challenges:**
   * Accuracy Issues: AI-based translations may not always capture the nuances of languages.
   * Performance & Latency: Real-time translation for voice input requires efficient processing.
   * Data Privacy :Handling user input (text, voice, and PDFs) requires secure encryption.
   * OCR Limitations: Extracting text from handwritten or low-quality scanned documents may be challenging.
   * Multi-language UI: Ensuring a smooth experience for users in various languages.

## **Phase-3: Project Design**

1. **System Architecture:**
   * User enters vehicle-related query via UI.
   * Query is processed using **Google Gemini API**.
   * AI model fetches and processes the data.
   * The frontend displays **vehicle details, reviews, and comparisons**.
2. **User Flow:**
   * Step 1: User enters a query (e.g., "Best motorcycles under ₹1 lakh").
   * Step 2: The backend **calls the Gemini Flash API** to retrieve vehicle data.
   * Step 3: The app processes the data and **displays results** in an easy-to-read format.
3. **UI/UX Considerations:**
   * **Minimalist, user-friendly interface** for seamless navigation.
   * **User friendly** UI design

## 

## 

## **Phase-4: Project Planning (Agile Methodologies)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** |
| Sprint 1 | Set up Vite + React + TypeScript, Design reusable components | 🔴 High | 3 hours (Day 1) | End of Day 1 | Spoorthi Reddy |
| Sprint 2 | Implement Routing, Develop Login & Signup forms, State management | 🔴 High | 3 hours (Day 1) | End of Day 1 | Keerthana , Swabhavika |
| Sprint 3 | Fetch & display mock data, Add interactivity & animations, Performance Optimization | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Sanjana |
| Sprint 3 | Testing, Debugging, Final optimizations & Deployment | 🟢 Low | 1 hour (Day 2) | Mid-Day 2 | Entire Team |

### 

### **Sprint Planning with Priorities**

### **Sprint 1 – Setup & Integration (Day 1)**

**(🔴 High Priority)** Set up the **environment** & install dependencies.  
 **(🟡 Medium Priority)** Build a **basic UI with input fields**.

### **Sprint 2 – Core Features & Debugging (Day 2)**

**(🔴 High Priority)** Implement **search & comparison functionalities**.  
 **(🔴 High Priority)** Debug API issues & handle **errors in queries**.

### **Sprint 3 – Testing, Enhancements & Submission (Day 2)**

**(🟡 Medium Priority)** Test API responses, refine UI, & fix UI bugs.  
 **(🟢 Low Priority)** Final **demo preparation & deployment**.

## **Phase-5: Project Development**

### **Objective:**

Implement core features of the Auto Sage App.

### **Key Points:**

1. **Technology Stack Used:**
   * Programming Language: **Python**
   * Frontend: **Vite, React, javaScript**
   * Database: **Not required initially (API-based queries)**
2. **Development Process:**
   * Design UI page using Html and Java Script.
   * Implement Page navigation using React Router.
   * Implement Jest/React Testing Library for component testing.
3. **Challenges & Fixes:**
   * **Challenge:** Data Handling Without a Backend.  
      **Fix:** Use local JSON files, external APIs
   * **Challenge:** Deployment Issues  
      **Fix:** Ensure vite.config.js has proper base paths, and all dependencies are installed before deployment.

## **Phase-6: Functional & Performance Testing**

### **Objective:**

Ensure that the Trans lingua-Multi language Translator works as expected.

|  |  |  |
| --- | --- | --- |
| **Type of Testing** | **Focus** | **Tools** |
| Functional Testing | UI behaviour, navigation, state, API calls | Jest, React Testing Library, Cypress |
| Performance Testing | Load time, re-renders, bundle size, optimization | Lighthouse, WebPageTest, React Dev Tools. |

## **Final Submission**

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**