**Technical Proposal: Mobile App for Diagnostic Translation with Free AI**

**App Name:** To Define  
**Platform:** Native Android (Android Studio)  
**Languages:** Kotlin/Java (frontend), PHP (backend, optional), CSS (styling)  
**Translation AI:** Free solutions (OPUS-MT, LibreTranslate, or MyMemory API)

**1. Executive Summary**

**Goal:**  
Develop a native Android app that enables technicians to translate diagnostic content (manuals, error codes, instructions) into regional languages using **free AI**, working both **offline and online**.

**Problem Solved:**

* Language barriers in technical documentation (e.g., English manuals for other language speakers).
* Time wasted and errors due to misinterpretations.

**2. Tech Stack**

| **Component** | **Technology Selected** |
| --- | --- |
| **Frontend** | Android Studio (Kotlin/Java + XML/CSS) |
| **Backend** | PHP (optional for storing translations) |
| **Database** | SQLite (local) / MySQL (remote) |
| **Translation** | OPUS-MT (offline) or LibreTranslate (API) |
| **OCR** | Firebase ML Kit (free) |

**3. Key Implementation**

**A. App Workflow**

1. **Language Selection:** Technician picks their regional language.
2. **Input Options:**
   * 📷 **Camera Scan:** Uses ML Kit to extract text from manuals and translates it.
   * ✍ **Manual Input:** User types text to translate.
   * 📁 **File Upload:** Supports PDFs/Images (text extraction).
3. **Translation:**
   * **Online:** LibreTranslate API (free, requires internet).
   * **Offline:** Embedded OPUS-MT model (no internet needed).
4. **Result:** Displays translated text + Text-to-Speech audio option.

**B. Free AI for Translation**

* **OPUS-MT:**
  + Pre-trained downloadable models (e.g., English → Other Regional Language).
  + Integrated via TensorFlow Lite in Android.
  + **Pros:** Works offline.
  + **Cons:** Increases APK size (needs compression).
* **LibreTranslate API:**
  + Open-source API (can be self-hosted on a PHP server).

**C. Storage**

* **SQLite:** Stores frequent translations for offline access.
* **PHP + MySQL (optional):** For syncing across devices.

**4. Technical Requirements**

* **Android Libraries:**
  + org.tensorflow:tensorflow-lite (for OPUS-MT offline).
  + com.google.mlkit:text-recognition (OCR).
  + com.android.volley (REST API calls).
* **Server (optional):**
  + Basic PHP hosting (000webhost, InfinityFree) for translations.

**5. MVP (Minimum Viable Product)**

1. **Screens:**
   * Language selection.
   * Camera/text input for translation.
   * Results with audio playback option.
2. **Supported Languages:** 3-5 (e.g., English, Spanish, Hindi).
3. **Offline Functionality:** Uses embedded OPUS-MT.

**6. Hackathon Deliverables**

* 📱 Functional APK with 2+ languages.
* 📹 Demo Video Showing:
  1. Scanning an English error code.
  2. Translating to other language in seconds.
* 📝 Slides Explaining:
  1. Why free AI was chosen (cost vs. efficiency).
  2. How to scale (e.g., adding more offline models).