**PROJECT REPORT ON**

# **AI-Powered Solution for Assisting Visually Impaired Individuals**

**Problem Statement**

This project aims to leverage Generative AI to assist visually impaired individuals in perceiving and interacting with their surroundings.

Visually impaired individuals often face challenges in understanding their environment, reading visual content, and performing tasks that rely on sight.

# **Project Overview**

“OptiAssist” is an AI-powered web application designed to assist visually impaired users by interacting with images in meaningful ways. The app offers a variety of features that help users understand their environment, process text from images, and provide task-specific guidance, all via audio.

**Features:**

* **Scene Understanding:** Describes the content of uploaded images.
* **Text-to-Speech:** Converts scene descriptions or extracted text from images into speech.
* **Object & Obstacle Detection:** Detects and warns about objects and potential obstacles in an image.
* **Personalized Assistance:** Provides task-specific guidance based on the uploaded image, such as recognizing objects or reading labels.

**Implementation Details:**

Technologies Used are…

* Python 3.7+
* ***LangChain:*** Conversational AI capabilities.
* ***Streamlit:*** User-friendly web application interface.
* ***Google Generative AI (Gemini API):*** Generates detailed scene descriptions.
* ***Pytesseract:*** Optical Character Recognition (OCR) to extract text from images.
* ***pyttsx3:*** Text-to-Speech engine for audio output.

**Step 1: Install Python Dependencies**

1. **Create a virtual environment** (optional but recommended):
   * This helps to manage dependencies for your project.
2. **Activate the virtual environment**:
   * On **Windows**: Activate the virtual environment to work in an isolated environment.
3. **Install required dependencies**:
   * Install all the necessary Python packages for running the application.
4. **Tesseract OCR Setup**:
   * Download and install [Tesseract OCR](https://github.com/tesseract-ocr/tesseract) on your machine.
   * Ensure the path to tesseract.exe is set correctly in your script.

**Configuration**

**Google Generative AI Setup**

To enable AI-based functionality using Google's Gemini model, follow these steps:

1. **Create a Google Cloud Project** and enable the **Generative AI API**.
2. **Obtain an API Key** for access to the **Gemini API**.
3. **Set the environment variable** for the API key in your system or directly in your script.

**Usage**

**Running the Application**

1. **Run the app** using Streamlit:
   * Launch the application using Streamlit in the terminal.
2. **Access the application**:
   * Once the app is running, you can access it from your browser.
3. **User Interface**:
   * **File Upload**: Users can upload images (formats like .jpg, .jpeg, .png).
   * **Action Buttons**:
     + **Describe Scene**: This button generates and displays a description of the uploaded image.
     + **Text-to-Speech**: Converts the scene description into audio.
     + **Personalized Assistance**: Provides task-specific guidance based on recognized objects or labels.
4. **Audio Output**: After each operation, the app will provide audio feedback for visually impaired users.

**Code Structure**

**Main File (app.py)**

This file contains all the functionalities for the application.

**Key Sections:**

1. **Imports**: Required libraries for image processing, AI, and speech.
2. **Setup and Configuration**: Configuration of Tesseract and API keys.
3. **Main Functionality**:
   * **File Upload Handling**: Users upload images to be processed.
   * **Scene Description Generation**: AI generates a description of the scene in the image.
   * **Text-to-Speech**: Converts generated text to speech for visually impaired users.
   * **Personalized Assistance**: Offers guidance based on image contents for daily tasks.

Submitted by:

VADDI UDAY KUMAR

(IN9240143)