

# CareConnect - Voice Based Virtual Health Assistant

---

Shreeya Singh

UG4

IIIT Hyderabad

Link to Repo: [Github Repo Link](#)

Loom Link to Video: [Demo Video Link](#)

Drive Link to Video: [Demo Video Drive Link](#)

## Table of Contents

---

1. [Introduction](#)
2. [Project Overview](#)
3. [Components](#)
  - [Frontend](#)
  - [Flask Application](#)
  - [Plivo API Integration](#)
4. [Installation](#)
  - [Frontend Setup](#)
  - [Flask Application Setup](#)
  - [Ngrok Setup](#)
5. [Usage](#)
6. [Demo](#)
7. [Design Overview](#)
  - [Call Flow](#)
  - [Symptom-to-Disease Model](#)
8. [Benefits](#)
9. [Future Scope](#)

---

## 1. Introduction

---

Welcome to CareConnect, an innovative healthcare assistant leveraging Plivo APIs to provide accessible healthcare advice through phone calls. This document provides comprehensive information about the project, its components, installation instructions, and usage guidelines.

---

## 2. Project Overview

---

CareConnect aims to make healthcare advice easily accessible by utilizing Plivo's Voice API. It enables users to seek medical advice through phone calls, offering a seamless and convenient experience.

---

## 3. Components

---

### Frontend

The frontend is responsible for providing the user interface. It allows users to initiate a call, input their number which is then used to initiate an outbound call to the user.

### Flask Application

The Flask application acts as the backend server. It handles the initiation of outbound calls, processes user input, interfaces with the Hugging Face API for symptom-to-disease prediction, and communicates with the Plivo API for voice interactions.

### Plivo API Integration

Plivo's Voice API is a crucial component of CareConnect. It facilitates voice interactions between users and the system, enabling seamless communication.

It takes the speech input from the user and converts it to text using the Speech-to-Text API. The text is then used to predict a diagnosis based on the symptoms provided by the user. The diagnosis is then converted to speech using the Text-to-Speech API and communicated to the user.

Also used is the Plivo Number API, which is used to purchase a Plivo number for the outbound call.

---

## 4. Installation

---

Install required dependencies using

```
pip install -r requirements.txt
```

### Frontend Setup

1. Clone the repository: `git clone <repo-url>`
2. Navigate to the `frontend` directory using the following command:

```
cd CareConnect/src/frontend
```

3. Run the following command to start a local server:

```
python3 -m http.server
```

# Flask Application Setup

1. Navigate to the `src` directory:

```
cd CareConnect/src
```

2. Run the Flask application: `python main.py`.

## Ngrok Setup

1. Download and install Ngrok from <https://ngrok.com/download>.
2. Open a terminal and run Ngrok: `

```
ngrok http 5000
```

3. Ngrok will provide a secure tunnel to the locally running Flask app. Use the generated URL in your Plivo application by pasting it in the `.env` file (which is created to store the auth details and huggingface token).

---

## 5. Usage

1. Access the frontend interface via `http://0.0.0.0:8000/index.html`.
2. Click on "Start Call" to initiate a call.
3. Follow the prompts to input your symptoms using voice.
4. Wait for the system to provide a diagnosis based on your symptoms.

---

## 6. Demo

A live demonstration of the project is available at [Demo Video Link](#).

---

## 7. Design Overview

### Call Flow

1. Outbound call is initiated when the user clicks "Start Call".
2. User provides symptom input via voice.
3. Symptom-to-Disease model predicts a diagnosis using the speech input.

### Symptom-to-Disease Model

- The model is based on a Transformer-based encoder-decoder architecture.
  - Trained on a dataset of 100,000 symptom-disease pairs.
  - [Link To Model](#)
-

## 8. Benefits

---

- Quick and accessible healthcare advice.
  - Efficient processing of speech input.
  - Reliable diagnoses based on a comprehensive dataset.
- 

## 9. Future Scope

---

- Integration with Plivo Messaging API for follow-up information.
  - Implementation of Call Recording for reference.
  - Call Forwarding to Specialist Doctors for advanced consultation.
-