CIT-AI – Intelligent Citizen Engagement Platform

# 1. INTRODUCTION

# 1.1 Project Overview

Citizen AI (CIT-AI) is an intelligent citizen engagement platform designed to bridge the gap between citizens and Indian government services. Built using HTML, CSS, JavaScript for the frontend and Flask (Python) for the backend, it leverages AI, voice recognition, and multilingual support to simplify access to essential government schemes and documents like Aadhaar, PAN, Voter ID. It integrates with Firebase for real-time data storage and user management.

# 1.2 Purpose

The purpose of CIT-AI is to enable citizens, especially from rural and semi-urban areas, to easily access, understand, and apply for government services without technical barriers.

# 2. IDEATION PHASE

# 2.1 Problem Statement

Citizens struggle with complex processes, language barriers, and fragmented portals when accessing government services. This often leads to incomplete applications or delays.

# 2.2 Empathy Map Canvas

- Says: “I don’t understand how to apply for services.”

- Thinks: “Will my application be rejected due to mistakes?”

- Does: Visits multiple offices or portals for one task.

- Feels: Confused, anxious, frustrated.

# 2.3 Brainstorming

Team generated ideas like multilingual AI chatbot, voice support, dashboard, Firebase storage, and real-time notifications.

# 3. REQUIREMENT ANALYSIS

# 3.1 Customer Journey Map

1. User logs in.

2. Selects a service.

3. Gets guided assistance.

4. AI chatbot helps fill forms.

5. Notifications sent.

6. Downloads status receipt.

# 3.2 Solution Requirement

- Multilingual AI chatbot

- Voice recognition (TTS/STT)

- Firebase backend

- Dashboard and reminders

# 3.3 Data Flow Diagram

[Insert diagram showing User → Flask Backend → AI/Voice → Firebase → Output]

# 3.4 Technology Stack

- Frontend: HTML, CSS, JavaScript

- Backend: Flask (Python)

- Database: Firebase Realtime Database

- AI Services: Google Gemini, IBM Watson

# 4. PROJECT DESIGN

# 4.1 Problem Solution Fit

The solution simplifies service access using AI and multilingual interfaces, reducing the complexity of government service portals.

# 4.2 Proposed Solution

A Flask-based web app with:

- Voice-enabled AI chatbot

- Service-specific guides

- Multilingual support

- Personalized dashboard

# 4.3 Solution Architecture

# WhatsApp Image 2025-06-27 at 12.44.18_6e53e0fd5. PROJECT PLANNING & SCHEDULING

# 5.1 Project Planning

- Week 1: Ideation

- Week 2: Frontend/Backend setup

- Week 3: AI and Firebase integration

- Week 4: Testing

# 6. FUNCTIONAL AND PERFORMANCE TESTING

# 6.1 Performance Testing

Tests conducted:

- Chatbot response speed

- Voice recognition accuracy

- Page loading time

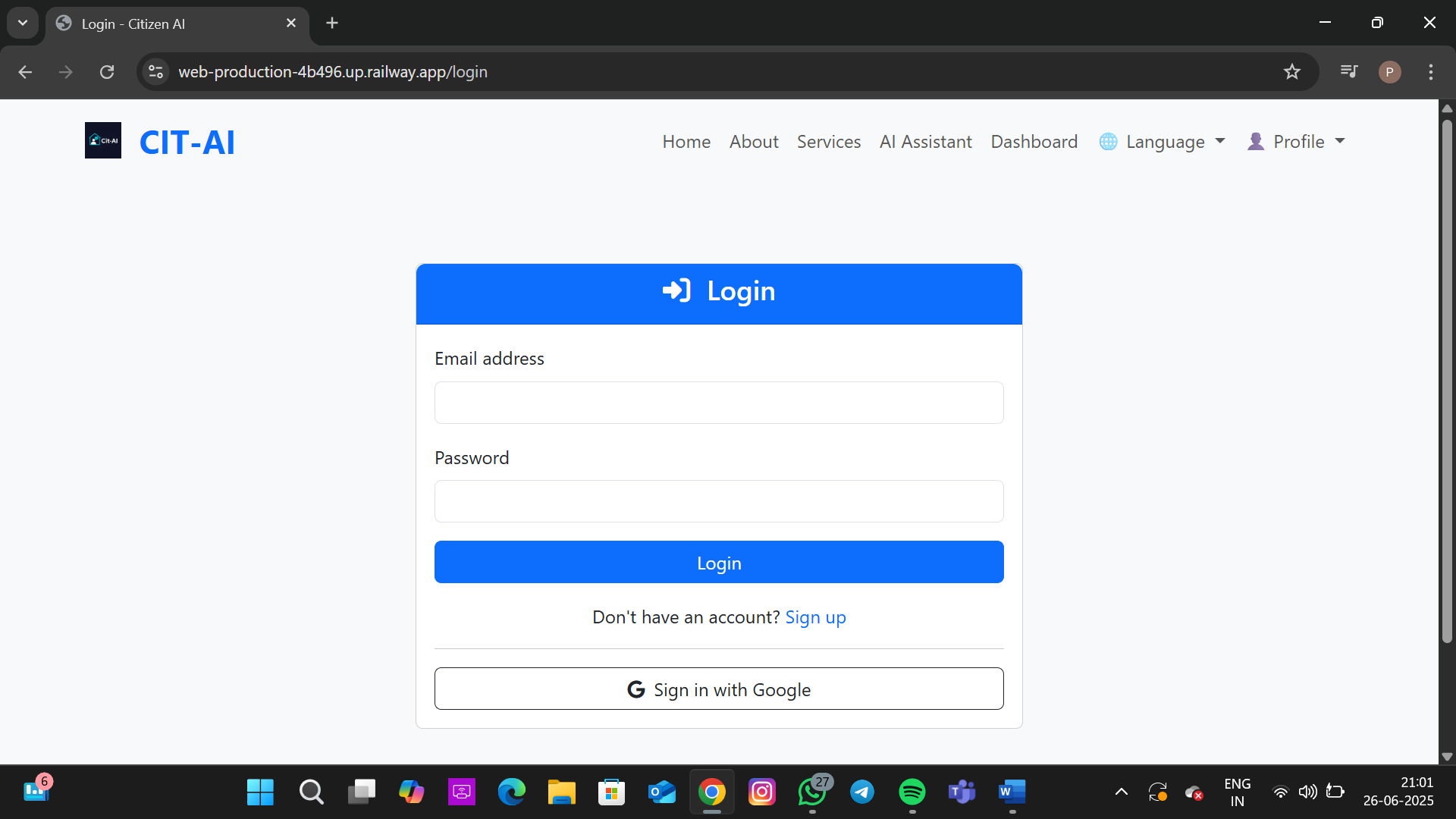
- Firebase data sync

# 7. RESULTS

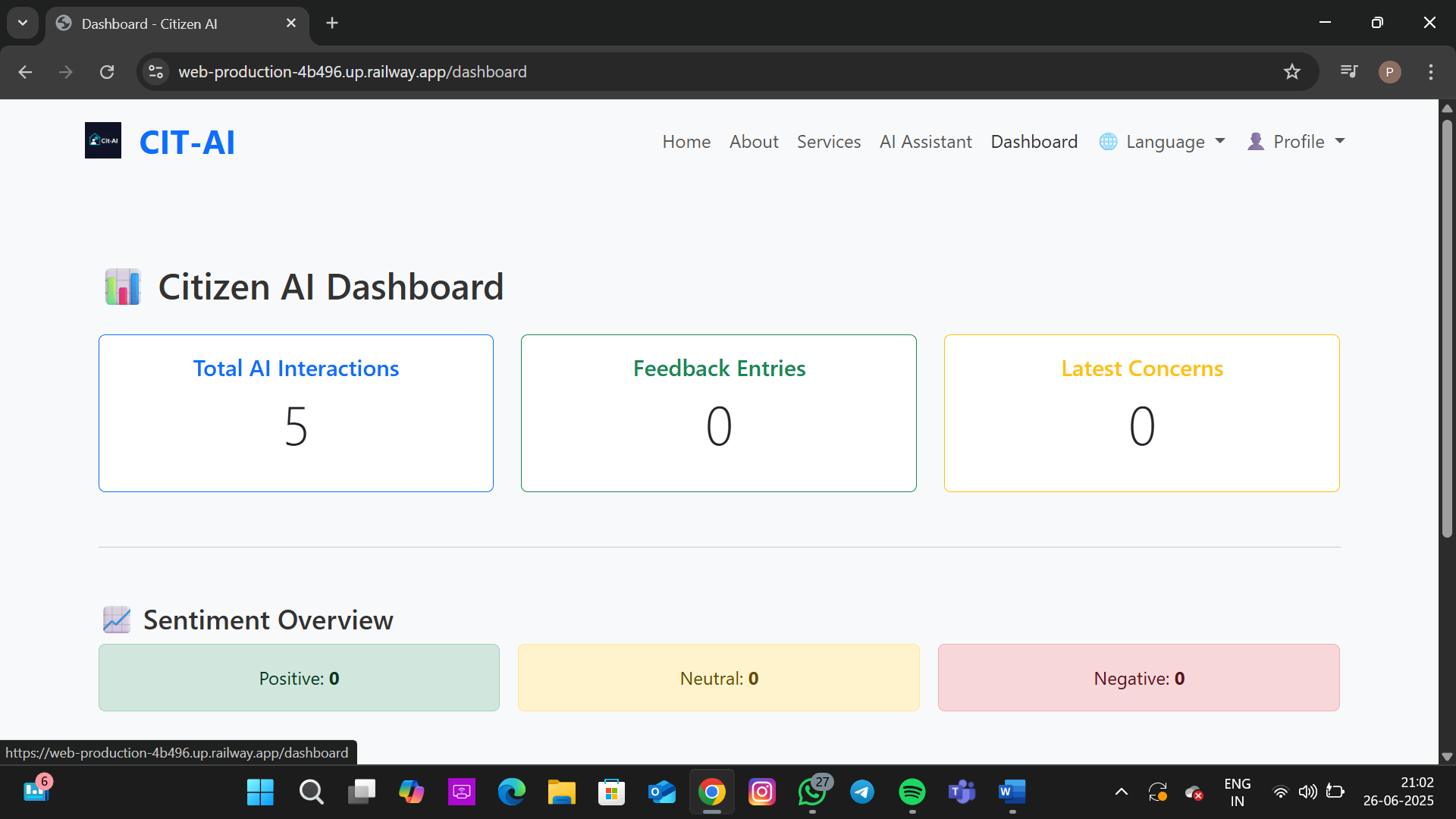
# 7.1 Output Screenshots

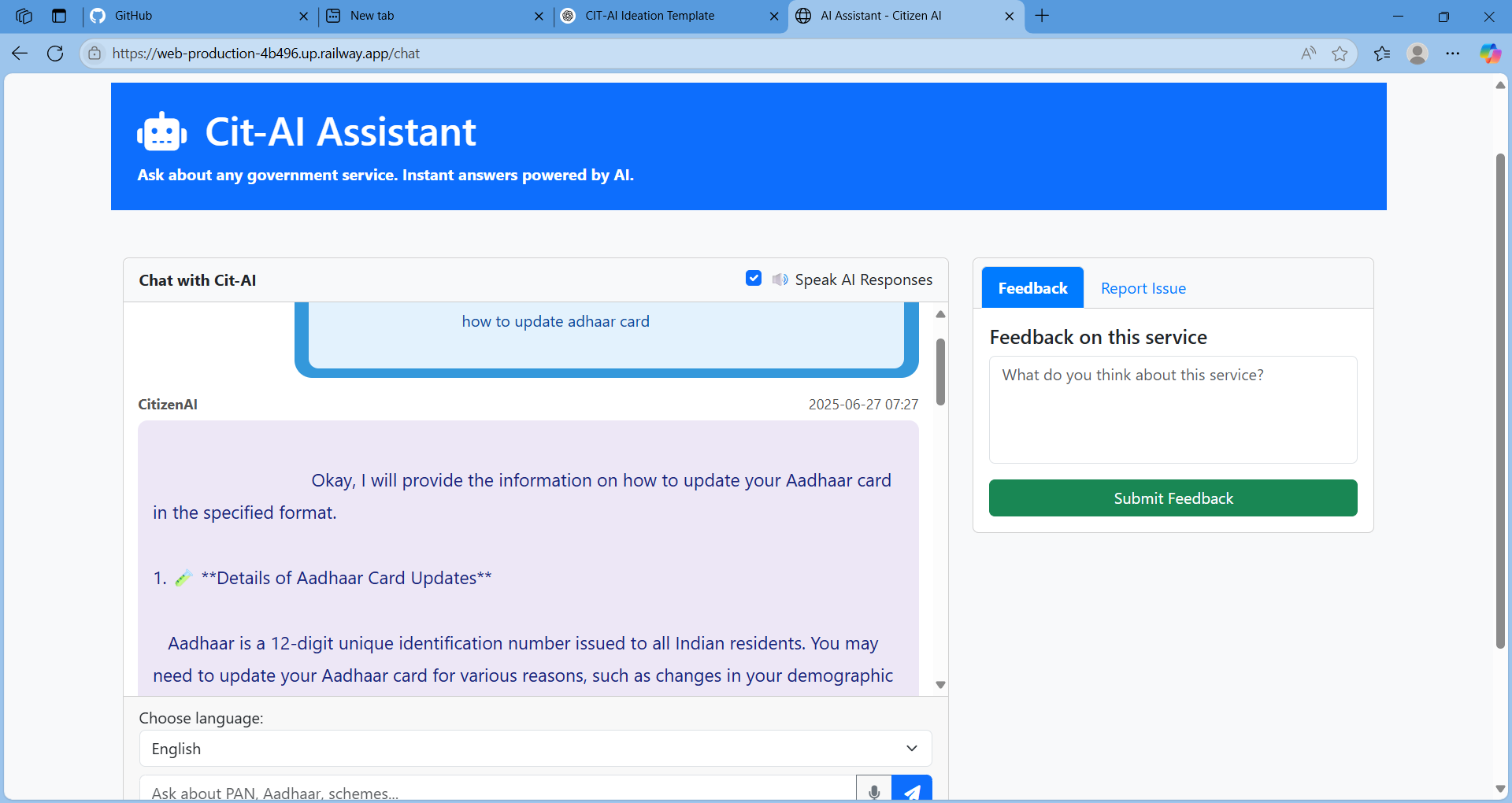
Screenshots:

- Login Page

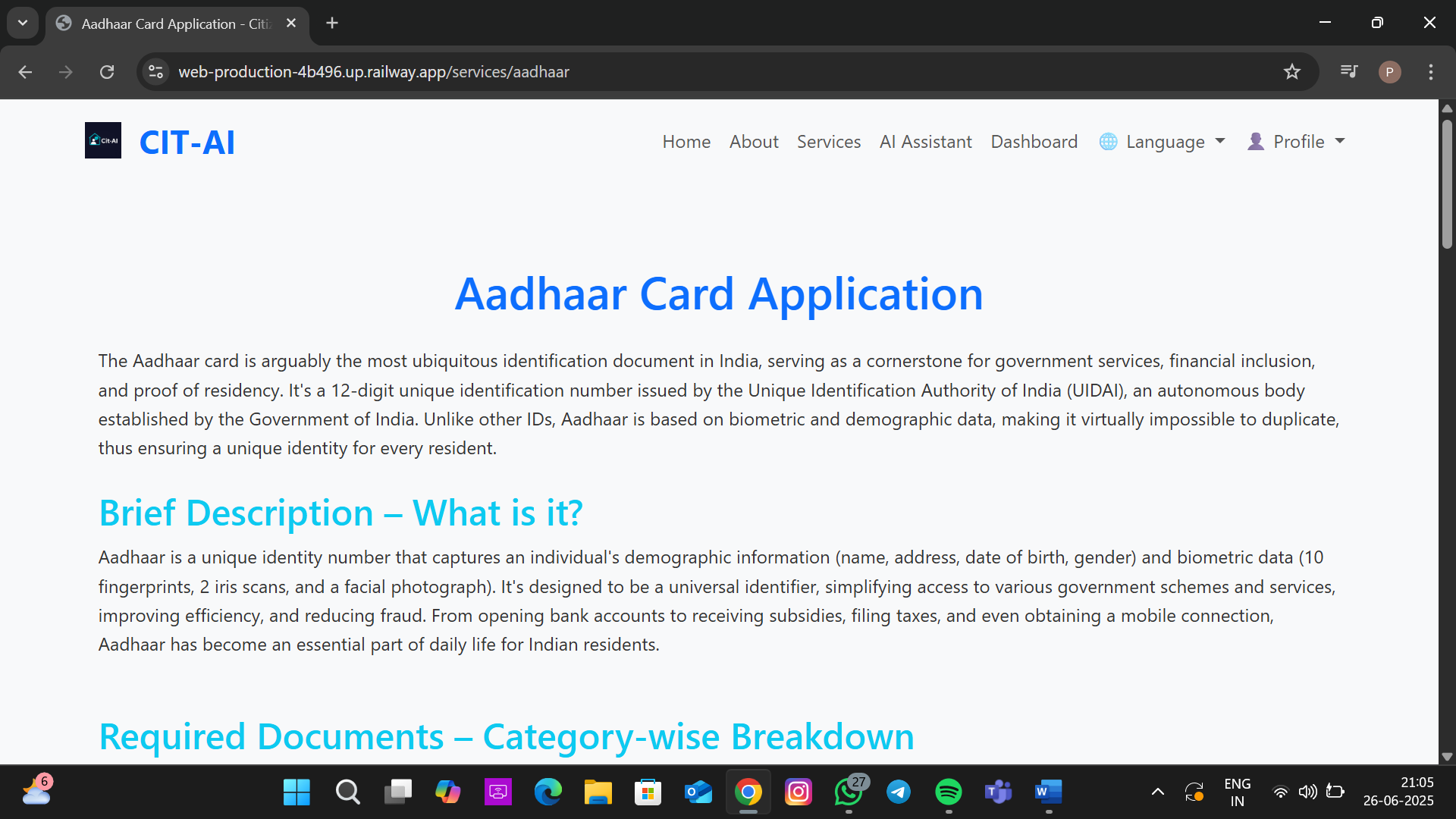


- Dashboard Page

  
- AI Chatbot Interaction



- Aadhaar Application Guide Page

  
- Voter ID Application Page

# 8. ADVANTAGES & DISADVANTAGES

\*\*Advantages:\*\*

- User-friendly for rural citizens

- Multilingual and voice-supported

- Personalized experience

\*\*Disadvantages:\*\*

- Internet dependency

- Learning curve for voice use

# 9. CONCLUSION

CIT-AI empowers citizens by providing seamless, AI-driven access to government services.

# 10. FUTURE SCOPE

- Mobile app

- Offline mode

- Visual guides in regional languages

- Smarter AI suggestions

# 11. APPENDIX

# 12. SOURCE CODE

## Example Flask Backend (app.py)

from flask import Flask, render\_template, request, jsonify

import firebase\_admin

from firebase\_admin import credentials, db

cred = credentials.Certificate("firebase-config.json")

firebase\_admin.initialize\_app(cred, {

'databaseURL': 'https://<your-database-name>.firebaseio.com'

})

app = Flask(\_\_name\_\_)

@app.route("/")

def home():

return render\_template("index.html")

@app.route("/submit\_form", methods=["POST"])

def submit\_form():

data = request.json

ref = db.reference("/forms")

ref.push(data)

return jsonify({"message": "Data submitted successfully!"})

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

## Example HTML Template (index.html)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>CIT-AI Platform</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

</head>

<body>

<h1>Welcome to CIT-AI</h1>

<button onclick="submitForm()">Submit Dummy Data</button>

<script src="{{ url\_for('static', filename='js/script.js') }}"></script>

</body>

</html>

## Example JavaScript (script.js)

function submitForm() {

const data = {

name: "Test User",

service: "Aadhaar Application"

};

fetch("/submit\_form", {

method: "POST",

headers: {

"Content-Type": "application/json"

},

body: JSON.stringify(data)

})

.then(response => response.json())

.then(result => {

alert(result.message);

})

.catch(err => console.error(err));

}

## Example CSS (style.css)

body {

font-family: Arial, sans-serif;

background-color: #f2f2f2;

text-align: center;

margin-top: 50px;

}

button {

padding: 10px 20px;

font-size: 16px;

}

## Firebase Rules

{

"rules": {

".read": "auth != null",

".write": "auth != null"

}

}

## Environment Variables (.env)

FIREBASE\_CREDENTIALS=path/to/firebase-config.json

SECRET\_KEY=your\_flask\_secret

**GitHub:** <https://github.com/lakshmibramham/citi-ai>

**Project Demo Link:**