CIT-AI – Intelligent Citizen Engagement Platform

# 1. Introduction

Project Title: CIT-AI – Intelligent Citizen Engagement Platform

Team Members:  
- k.Lakshmi Bramham– Frontend Developer (HTML/CSS/JS)  
- K.Vamshi – Backend Developer (Flask)  
- K.Sai Teja – Firebase Integration & AI  
- K.Pavan Thanai – UI/UX and Testing

# 2. Project Overview

Purpose:  
To create a centralized citizen engagement platform that allows users to access Indian government services using AI, voice, and multilingual support. The platform simplifies form submissions, provides service eligibility checks, and ensures real-time assistance.

Features:  
- AI chatbot with multilingual support  
- Real-time form guidance  
- Secure login and personalized dashboards  
- Firebase real-time data storage  
- Voice-based interface using TTS/STT  
- Government scheme tracking and reminders

# 3. Architecture

Frontend:  
Built using HTML, CSS, and vanilla JavaScript. Responsive layout with modular structure for reusability. Navigation through dynamic routing handled via templates.

Backend:  
Flask (Python) handles routing, form processing, user authentication, and integrates AI services and Firebase interactions.

Database:  
Firebase Realtime Database stores user data, chat history, scheme progress, and submission details.

# 4. Setup Instructions

Prerequisites:  
- Python 3.x  
- Flask  
- Firebase Project

Installation:  
1. Clone the repository  
2. Create a virtual environment and activate it  
3. Run `pip install -r requirements.txt`  
4. Set environment variables for Firebase and AI API keys in a `.env` file

# 5. Folder Structure

Client:  
- /static  
 - /css  
 - /js  
 - /assets  
- /templates  
 - login.html  
 - dashboard.html  
 - chat.html  
 - form\_pages.html

Server:  
- app.py  
- /routes  
- /ai\_modules  
- /firebase\_services  
- .env  
- requirements.txt

# 6. Running the Application

Backend + Frontend (Flask serves both):  
```bash  
python app.py  
```

# 7. API Documentation

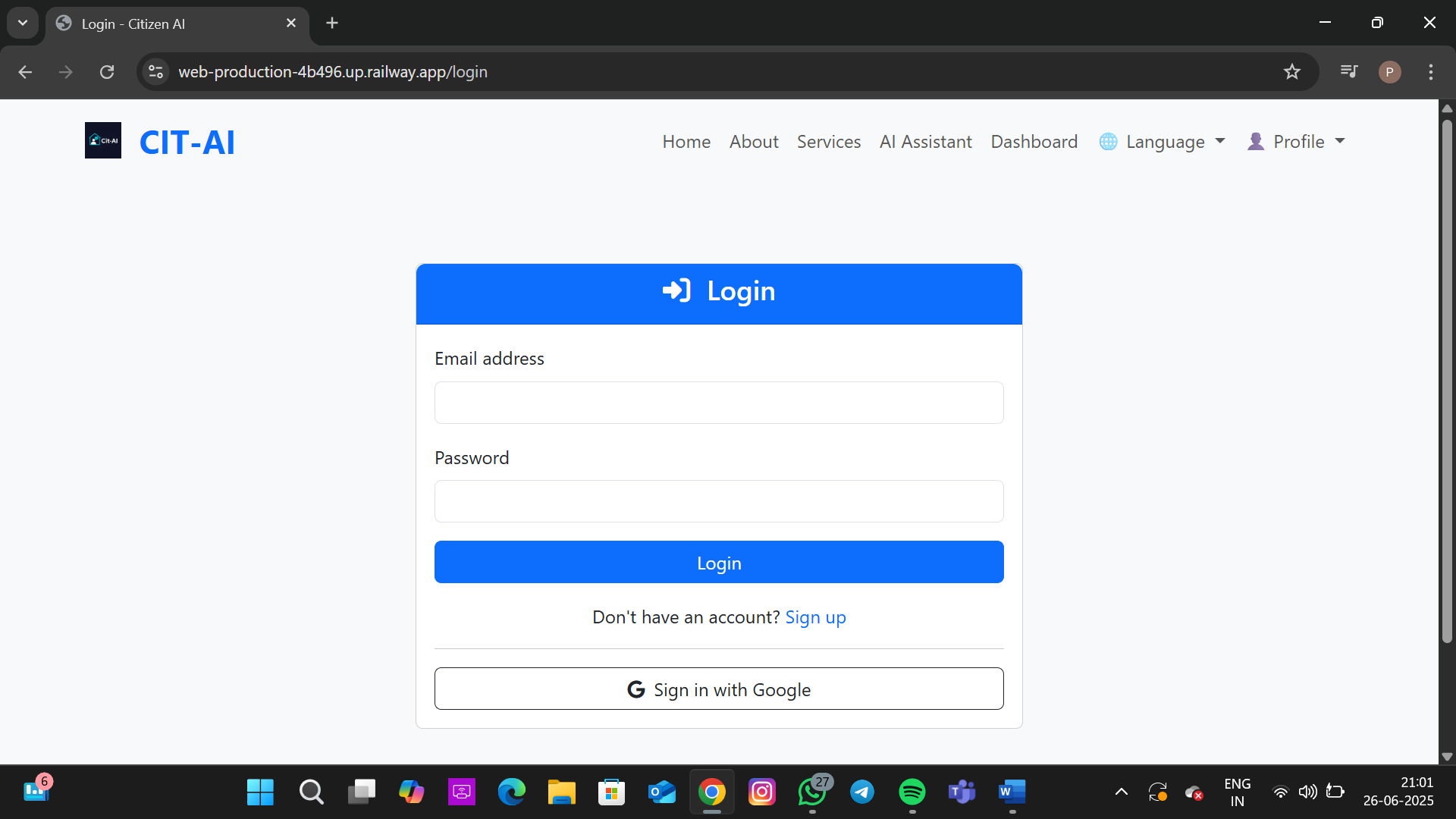
Example Endpoints:  
- POST /login – User login  
- POST /register – Register a new user  
- GET /dashboard – Load personalized dashboard  
- POST /submit\_form – Submit service form

# 8. Authentication

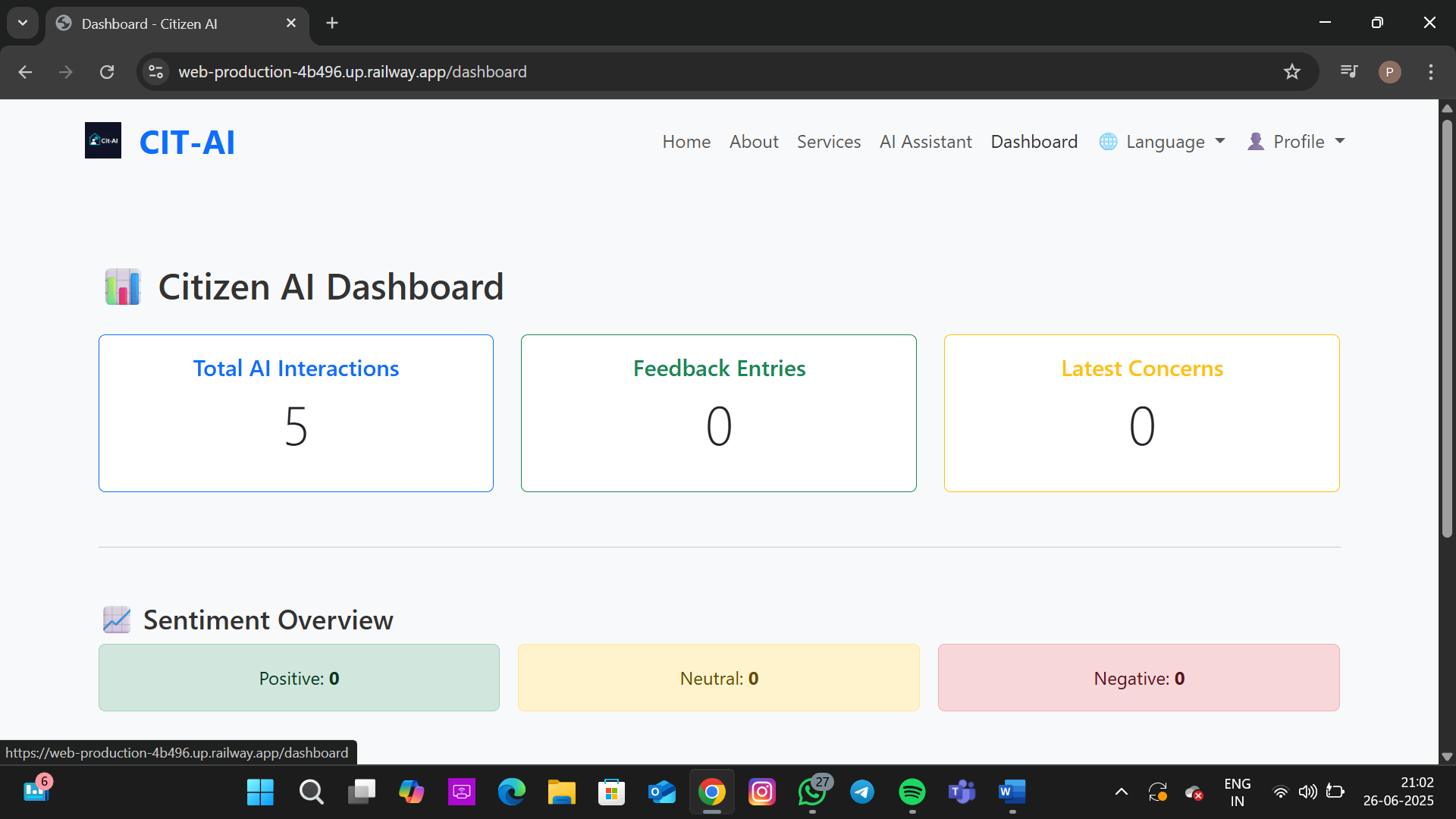
- Email/password-based login  
- Sessions managed using Flask-Login  
- Firebase used for authentication and data storage

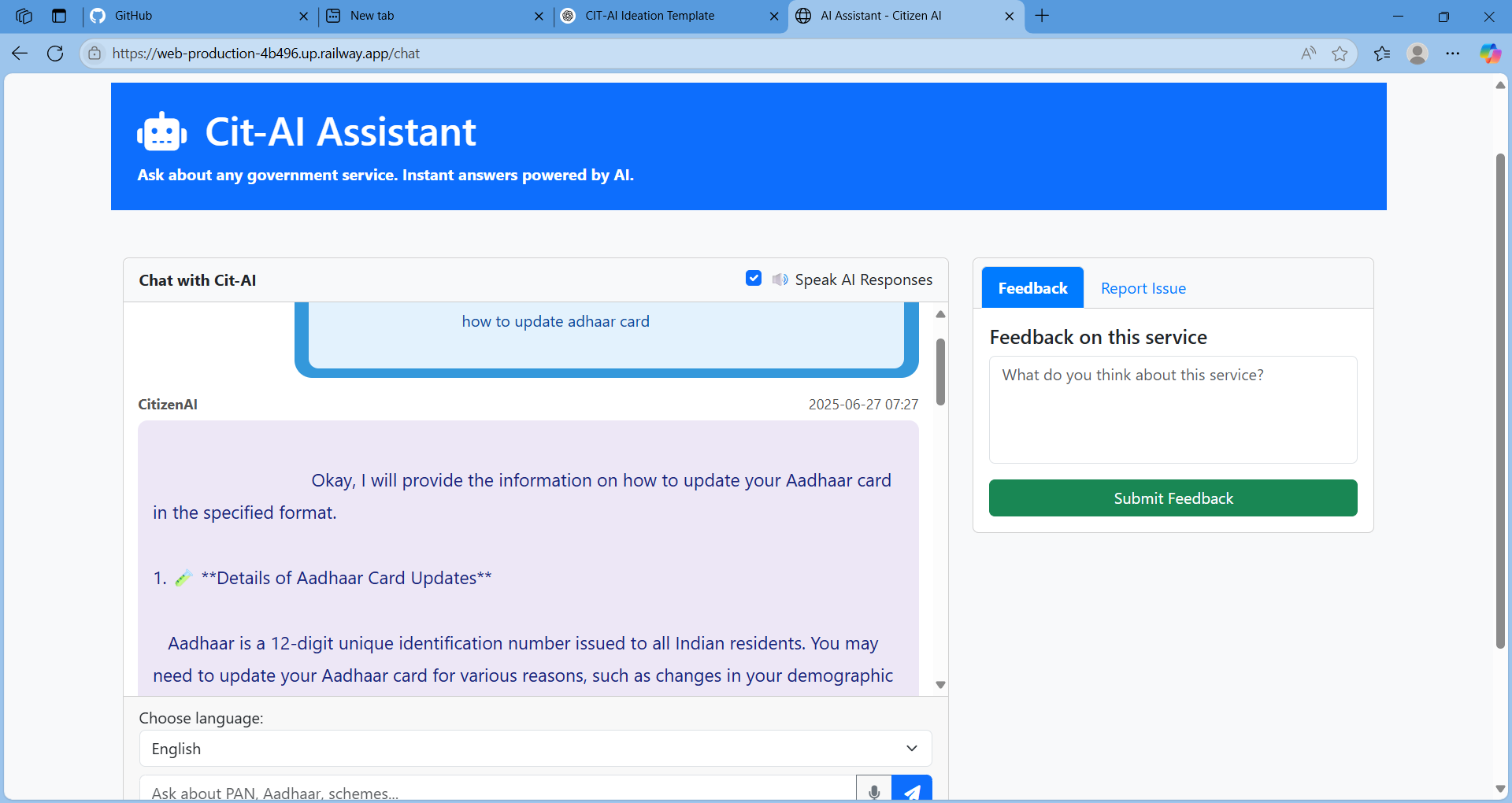
# 9. User Interface

Screenshots :  
- Login Page

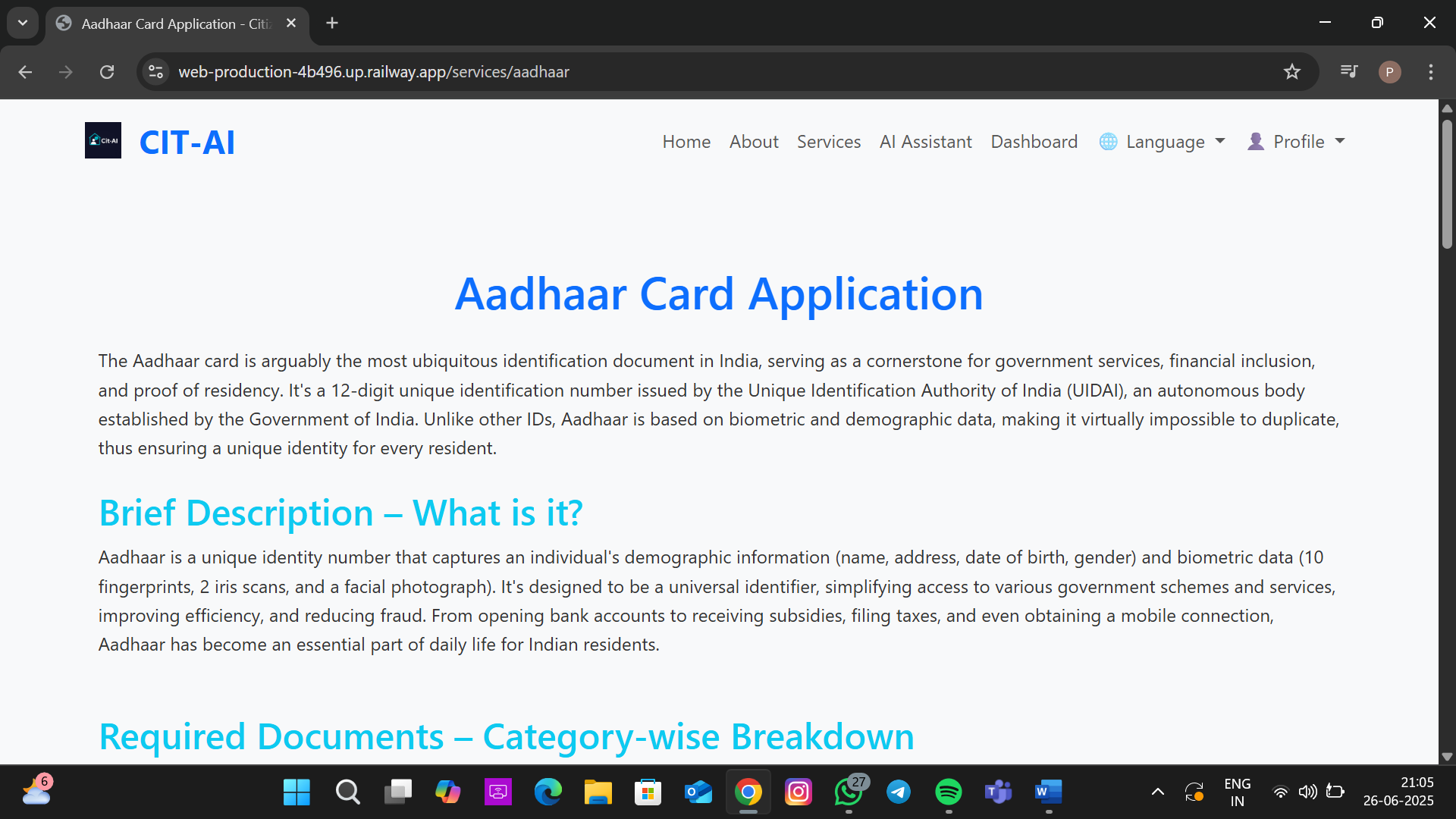


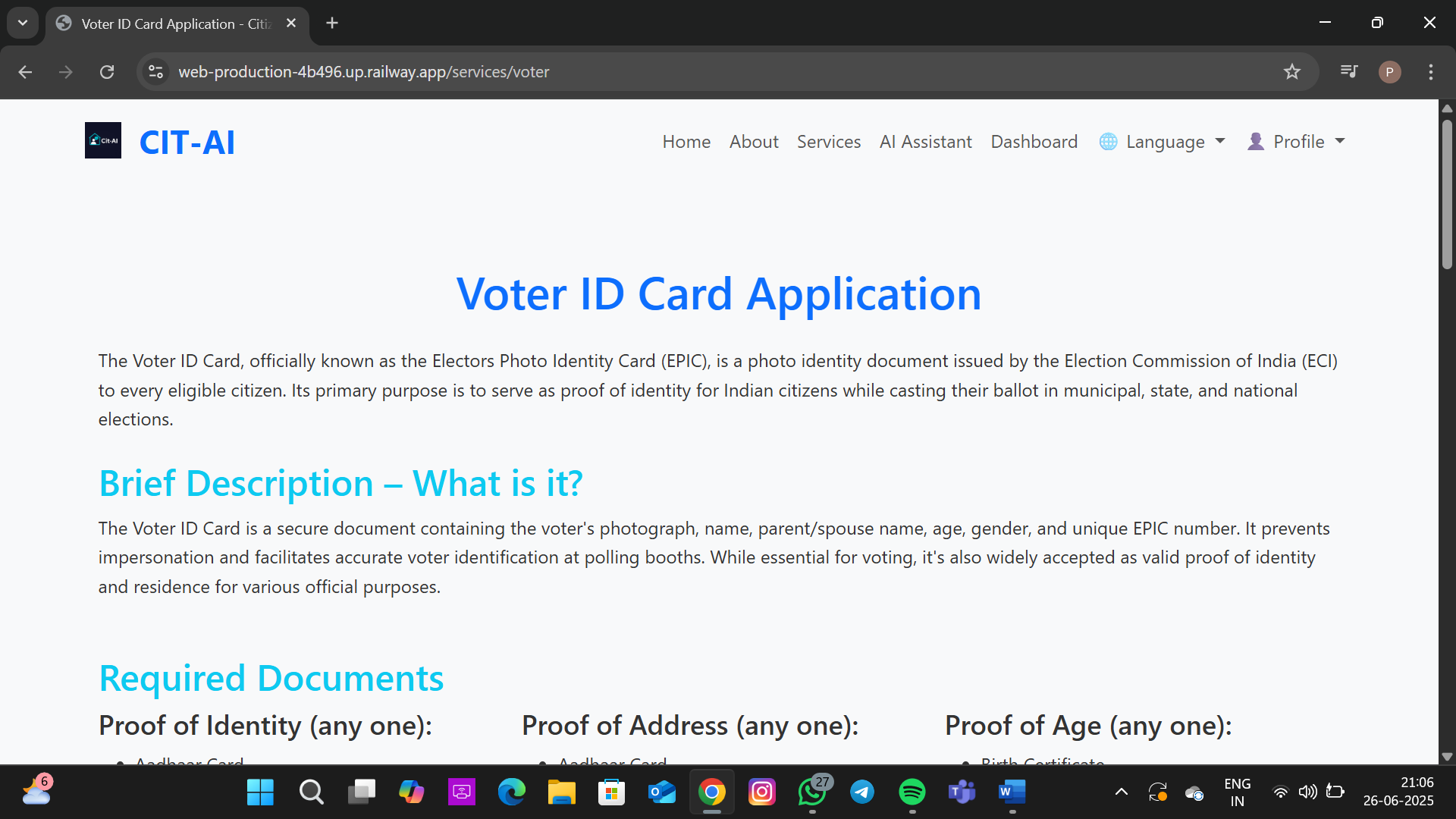
- Dashboard Page

  
- AI Chatbot Interaction



- Aadhaar Application Guide Page

  
- Voter ID Application Page



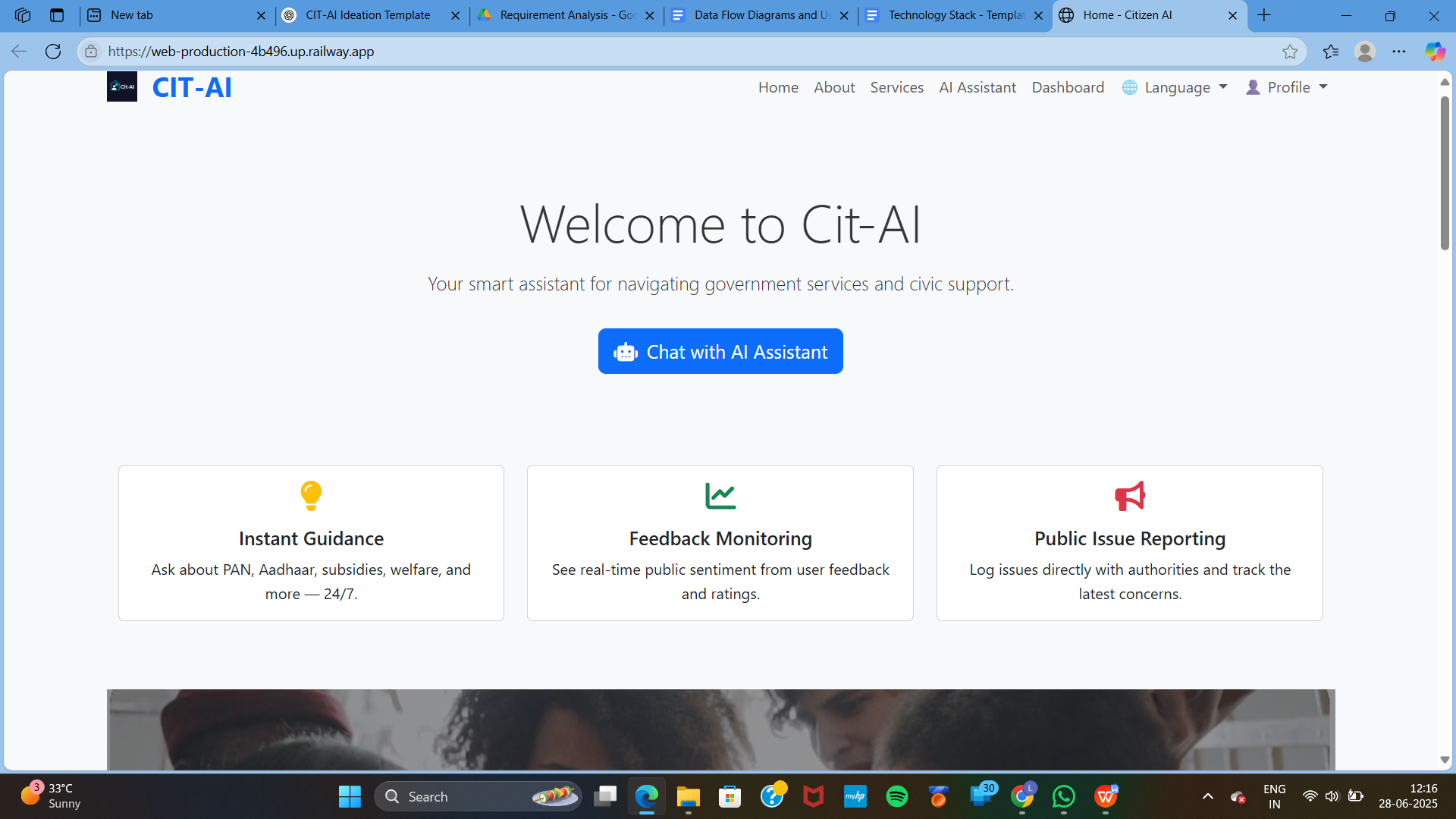
# 10. Testing

Testing Strategy:  
- Manual testing in different browsers  
- Console-based testing for API responses  
- Firebase read/write testing

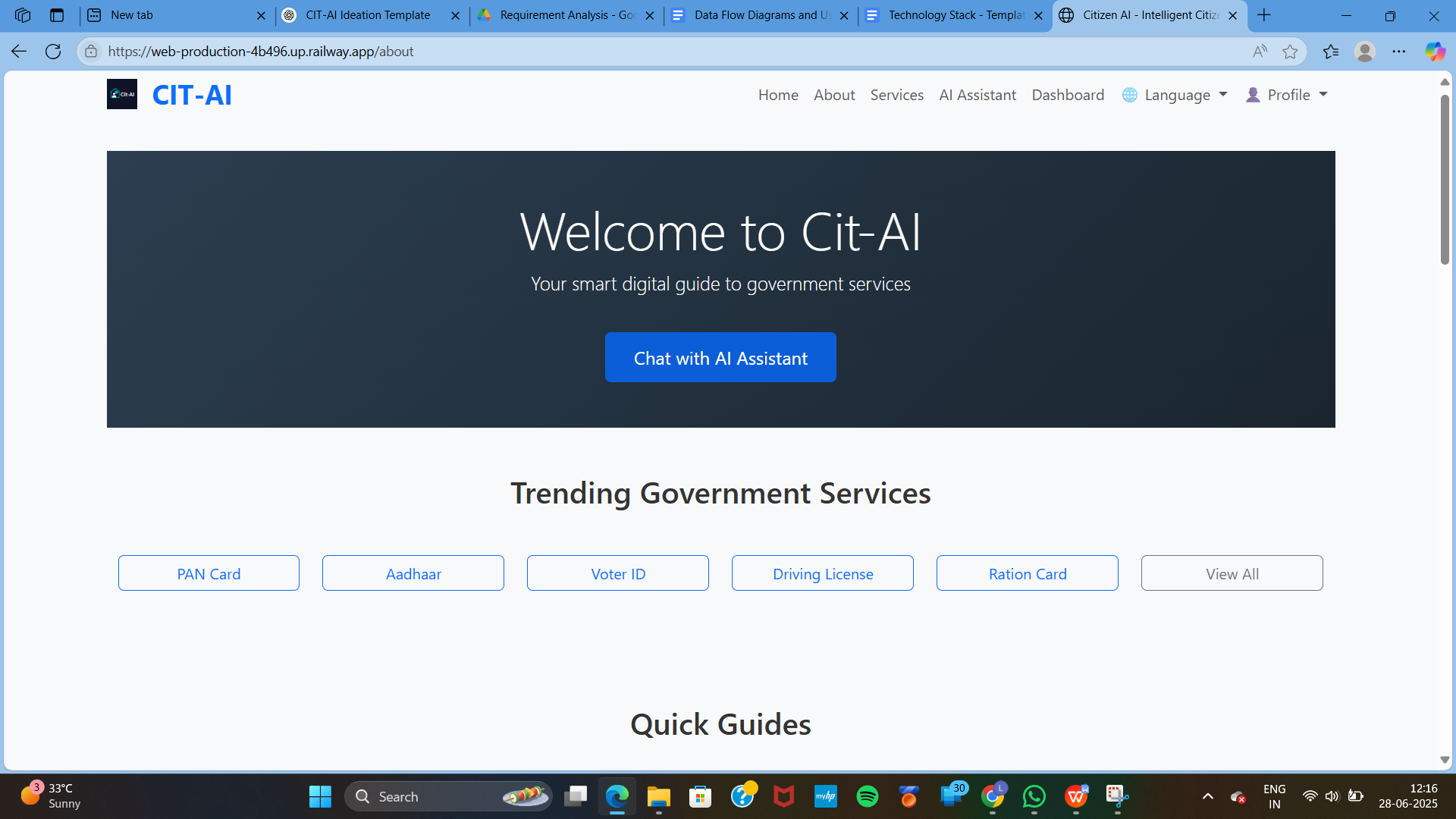
# 11. Screenshots or Demo

Screenshots :

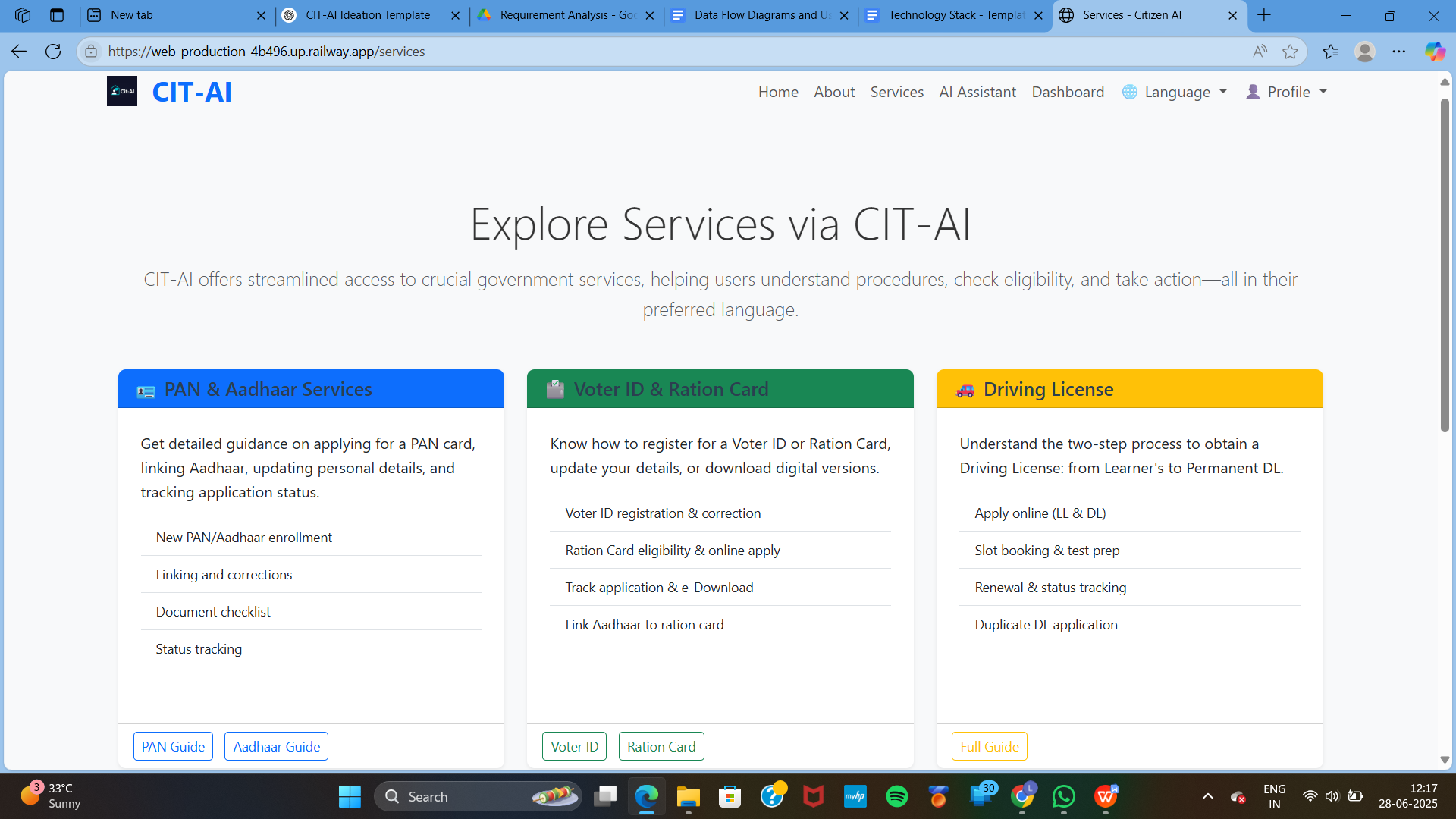
- Home Page



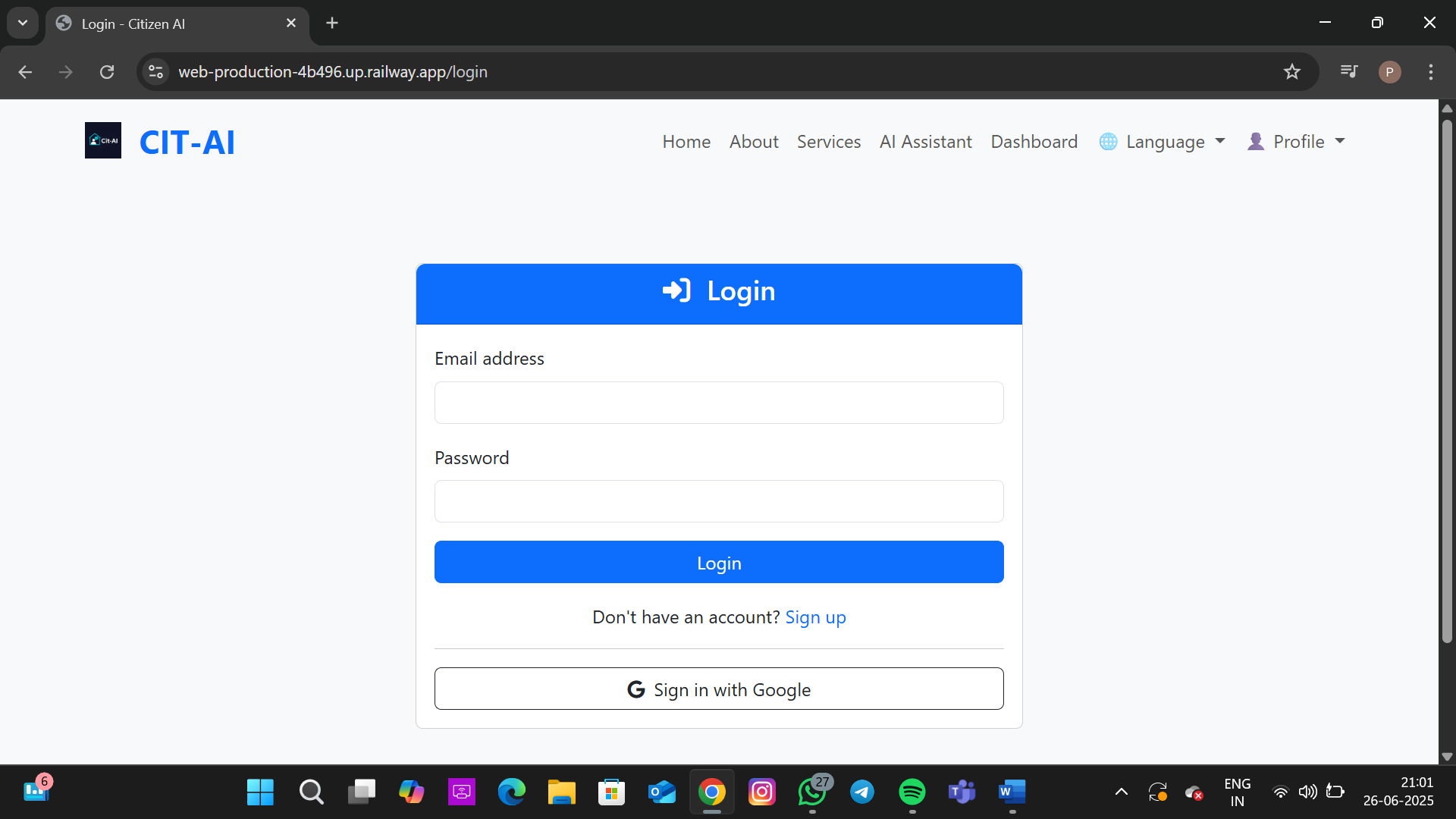
- About Page



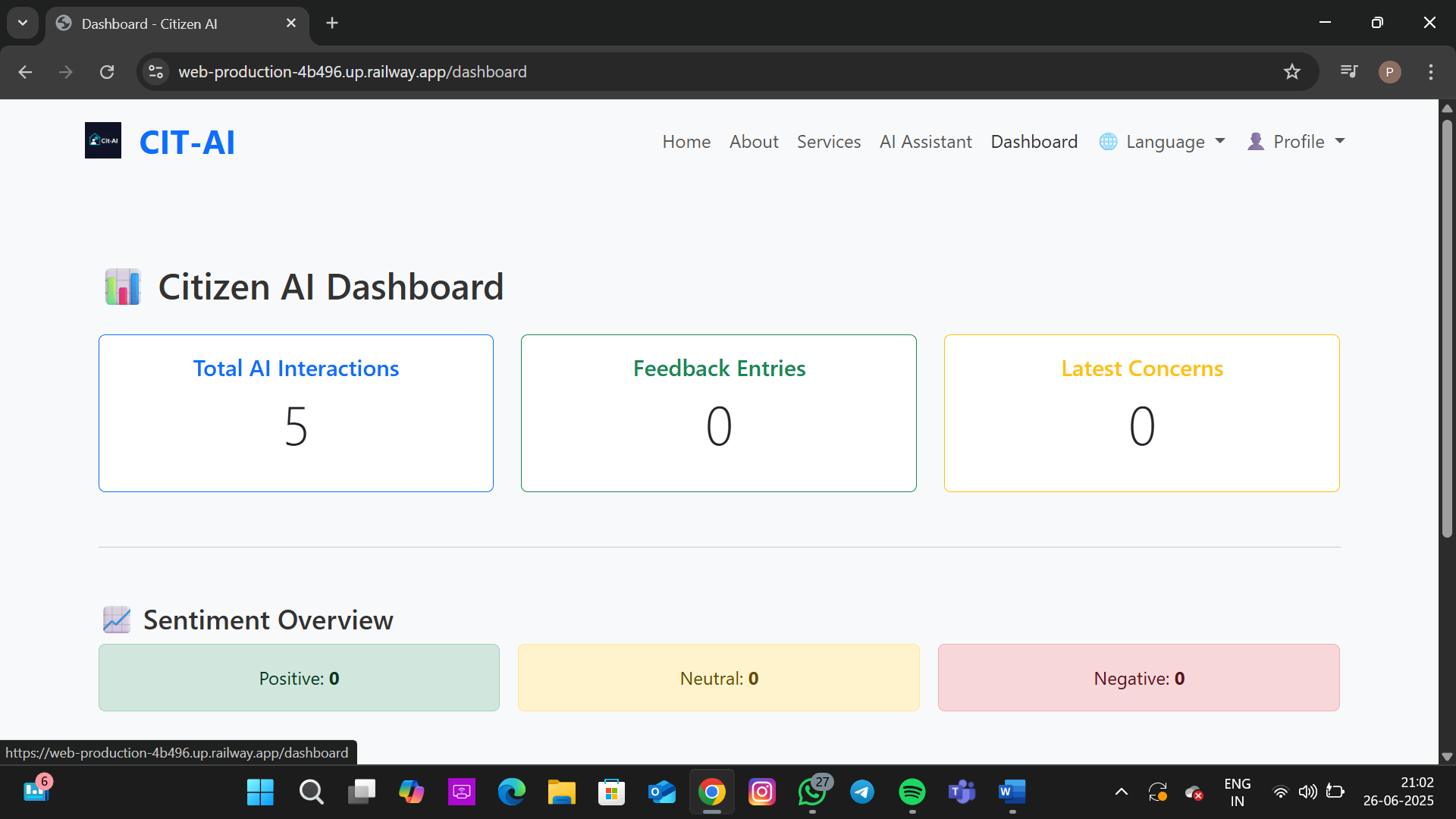
- Services Page

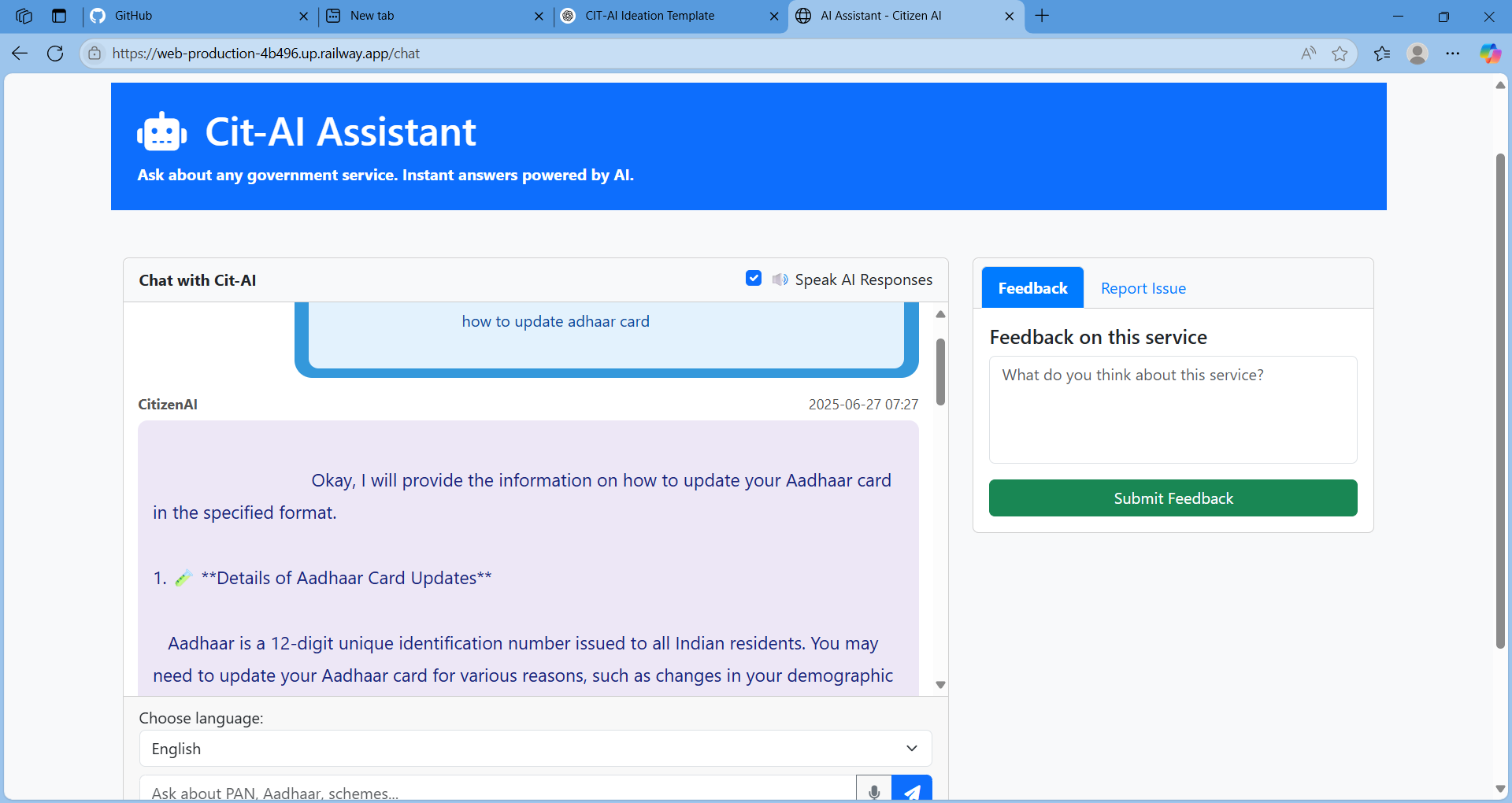


- Login Page

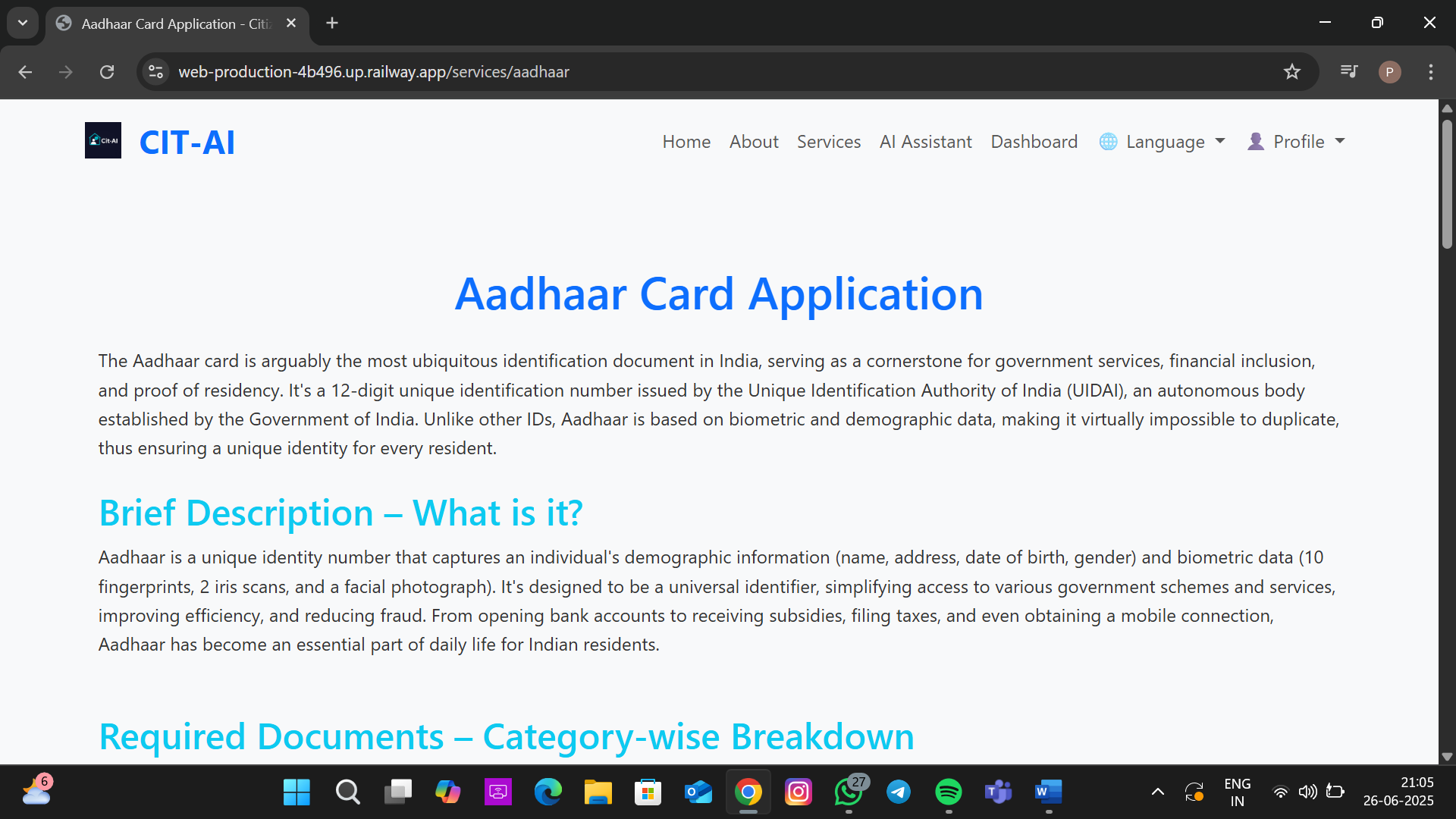


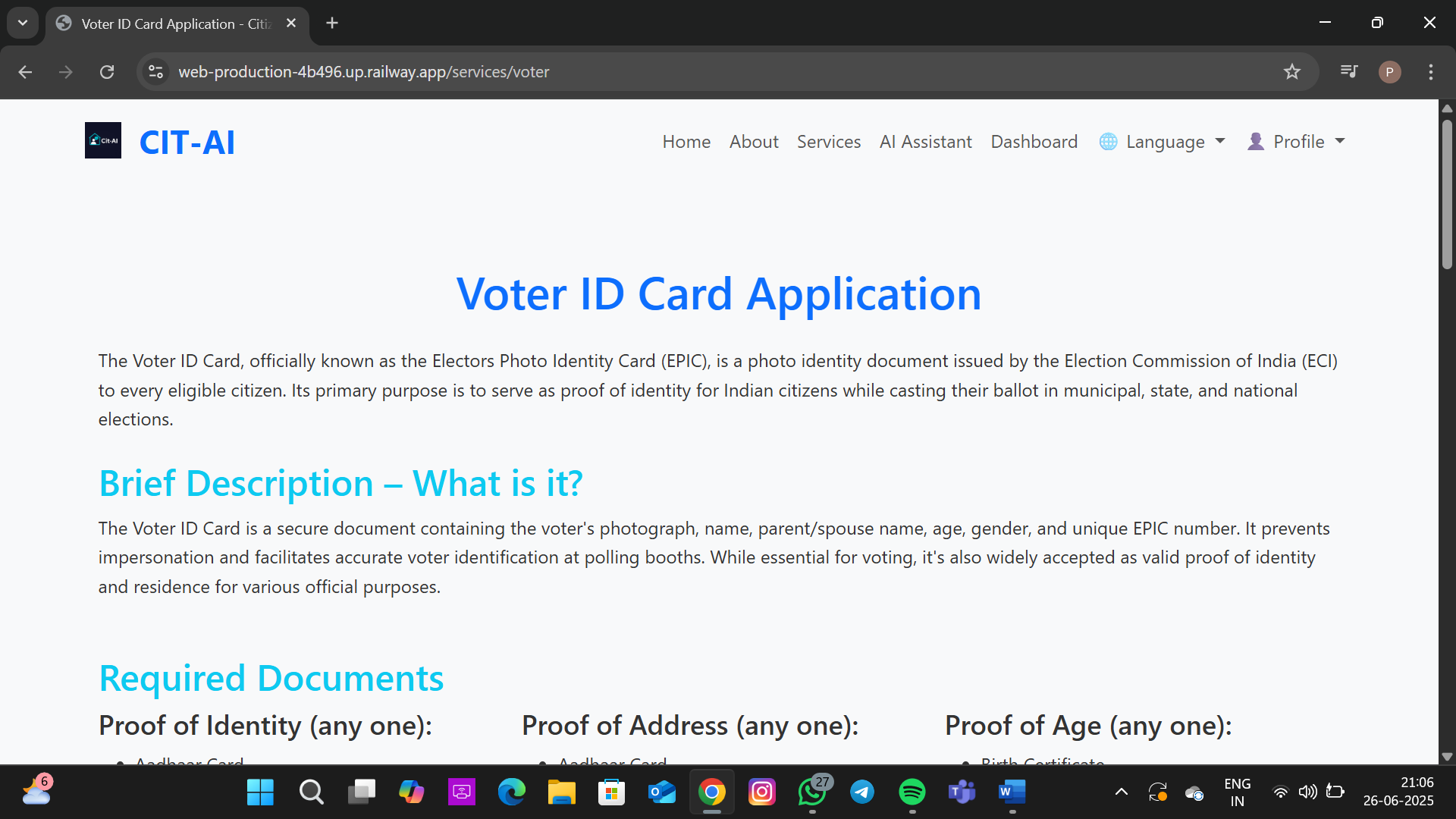
- Dashboard Page

  
- AI Chatbot Interaction



- Aadhaar Application Guide Page

  
- Voter ID Application Page



# 12. Known Issues

- Voice input may lag under slow internet  
- Firebase session timeout needs better handling

# 13. Future Enhancements

- Mobile app version  
- Offline document submission preview  
- Regional language voice support  
- Visual service walkthroughs  
- AI-based user intent prediction