CHIDAMBARAM PILLAI COLLEGE FOR WOMEN

CITIZEN AI- INTELLIGENT CITIZEN ENGAGEMENT PLATFORM

Team Leader : Dhanusiya U

Team Member : Hemalatha M

Team Member : Kanaga E

Team Member : Karpagam T

Team Member : Karthika V

INTERNAL EXTERNAL

ABSTRACT

Citizen AI is an intelligent citizen engagement platform designed to revolutionize how governments interact with the public. Leveraging Flask, IBM Granite models, and IBM Watson, Citizen AI provides real-time, AI-driven responses to citizen inquiries regarding government services, policies, and civic issues. The platform integrates natural language processing (NLP) and sentiment analysis to assess public sentiment, track emerging issues, and generate actionable insights for government agencies. A dynamic analytics dashboard offers real-time visualizations of citizen feedback, helping policymakers enhance service delivery and transparency. By automating routine interactions and enabling data-driven governance, Citizen AI improves citizen satisfaction, government efficiency, and public trust in digital governance.

PROJECT DESCRIPTION



INTRODUCTION TO PYTHON

Python is a high-level, general-purpose programming language known for its readability and ease of use. Created by Guido van Rossum and first released in 1991, it has gained significant popularity across various domains due to its versatile nature and extensive ecosystem.

INTRODUCTION TO AI

Artificial Intelligence (AI) is the creation of computer systems that can perform tasks typically requiring human intelligence, such as learning, reasoning, decision-making, and problem-solving. AI works by analyzing data to recognize patterns and make informed decisions, enabling machines to understand human language, recognize objects, and learn from new information. Applications range from personalized recommendations and smart assistants to self-driving cars and advanced data analysis, enhancing efficiency and productivity across various fields like healthcare, finance, and entertainment.

INTRODUCTION TO IBM GRANITE

IBM Granite is a family of open-source, generative AI models designed for enterprise use, offering a high degree of control, customization, and transparency. Available under the, these models can be downloaded, modified, and fine-tuned for specific needs, making them ideal for businesses concerned with data security and controlling their AI infrastructure. Granite models support various tasks, including code intelligence, and come in different sizes and specialized versions like time-series and multimodal models.

FRONT END

HTML:

HTML is the standard markup language used to create the structure and content of web pages. It uses a system of "tags" to define different elements within a document, such as headings, paragraphs, images, links, and lists. These tags provide semantic meaning to the content, indicating what each part represents.

CSS:

CSS is a style sheet language used to control the presentation and visual appearance of HTML documents. While HTML provides the structure, CSS dictates how that structure is displayed, including aspects like colors, fonts, spacing, layout, and responsiveness.

BACK END

FLASK:

**Flask** is a **micro web framework** written in **Python**. It's used to build web applications quickly and with minimal overhead. Flask is designed to be lightweight, flexible, and easy to get started with, making it a great choice for small to medium-sized web applications or APIs.

IBM GRANITE AI MODEL

INIBM Granite is a family of open-source, enterprise-focused large language models (LLMs) and other AI tools designed for tasks involving both language and code, available. These decoder-only models are built on trusted, curated data for business applications like customer service, HR, and code modernization, offering various sizes and capabilities for specific tasks such as question answering, summarization, text and code generation, and classification.

Coding:

import random

class CitizenAI:

def \_\_init\_\_(self, citizen\_name):

self.citizen\_name = citizen\_name

def govt\_services(self, query):

services = {"aadhaar": "You can apply/update Aadhaar at UIDAI website: https://uidai.gov.in", "passport": "You can apply for Passport at https://www.passportindia.gov.in", "voter": "Register for Voter ID at https://www.nvsp.in",

"pan": "Apply for PAN at https://www.onlineservices.nsdl.com }

for key, value in services.items():

if key in query.lower():

return value

return "Sorry, I don't have information on that service."

def report\_issue(self, query):

issues = { "road": "Your road complaint has been noted. Local municipality will be informed.",

"electricity": "Electricity issue reported. Power board will check it soon.",

"water": "Water supply issue reported to the local corporation.",}

for key, value in issues.items():

if key in query.lower():

return value

return "Sorry, I could not identify the issue. Please try again."

def awareness(self):

tips = [

"Every citizen has the right to equality before law.",

"You should keep your city clean by avoiding littering.",

"Paying taxes honestly is a citizen’s responsibility."]

return random.choice(tips)

def random\_tip(self):

tips = [

"Plant a tree in your neighborhood.",

"Help senior citizens with digital services.",

"Promote eco-friendly transport.",

"Participate in local community services."]

return random.choice(tips)

def ask(self, query):

if any(word in query.lower() for word in ["aadhaar", "passport", "voter", "pan"]):

return self.govt\_services(query)

return self.report\_issue(query)

elif "awareness" in query.lower():

return self.awareness()

elif "tip" in query.lower():

return self.random\_tip()

else:

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

ai = CitizenAI(name)

while True:

query = input(f"\n{name}, enter your query (or type 'exit' to quit): ")

if query.lower() == "exit":

print("Thank you for using Citizen AI. Goodbye!")

**MODULES**

**Real-Time Conversational AI Assistant:**

The Real-Time Conversational AI Assistant in Citizen AI serves as the primary interface for citizen interaction. It allows users to engage with public services naturally by typing questions or requests. The system captures user input in real- time and immediately sends it to a powerful underlying AI model, such as IBM Granite. This model processes the query and generates a relevant, human-like response on the fly. It aims to provide a seamless and efficient conversational experience for civic engagement.

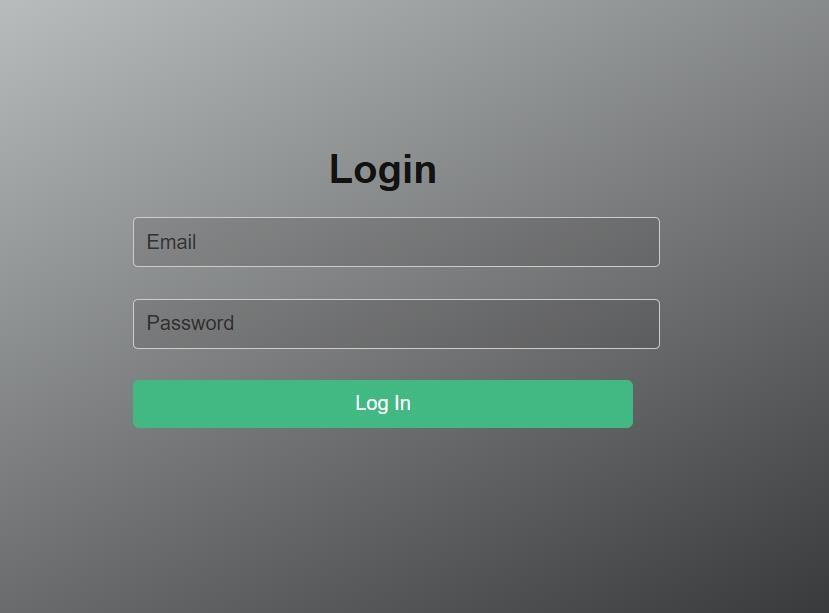
**Citizen Sentiment Analysis:**

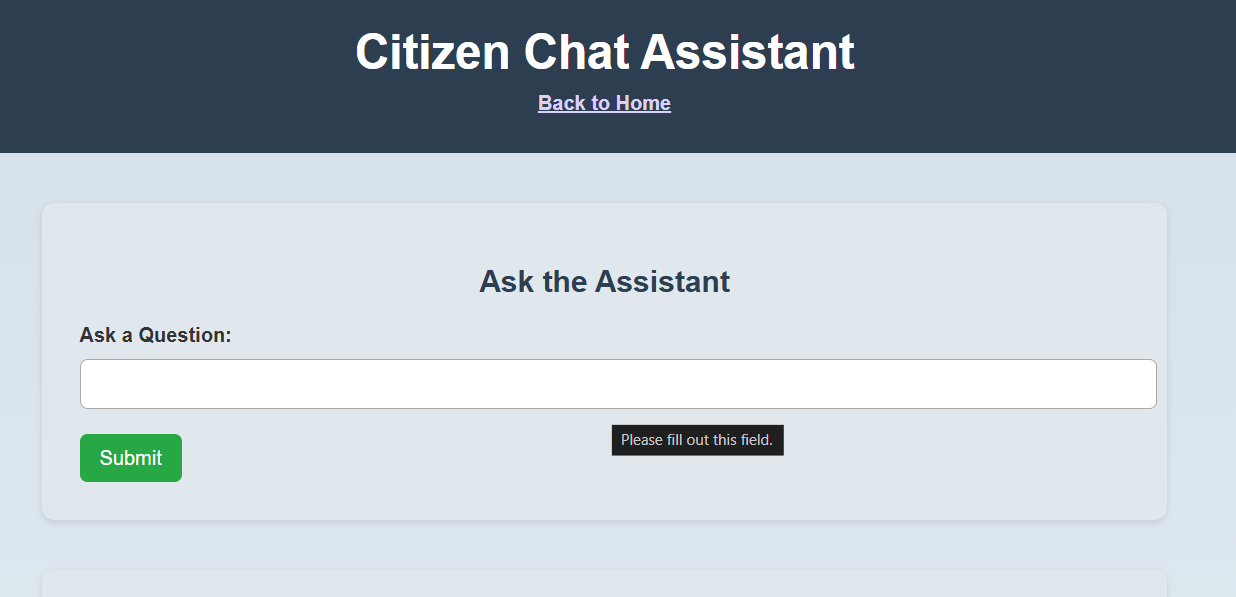
Citizen Sentiment Analysis in Citizen AI is a core feature designed to understand the public's feelings about government services and related topics.It works by analysing text input, whether from direct citizen feedback submitted through the platform or potentially from other digital interactions (though the current implementation focuses on submitted text).Using AI (like the simple analyse\_sentiment function in app.py), the system classifies the sentiment of the text as Positive, Neutral, or Negative. The results are presented on the dashboard for easy monitoring.

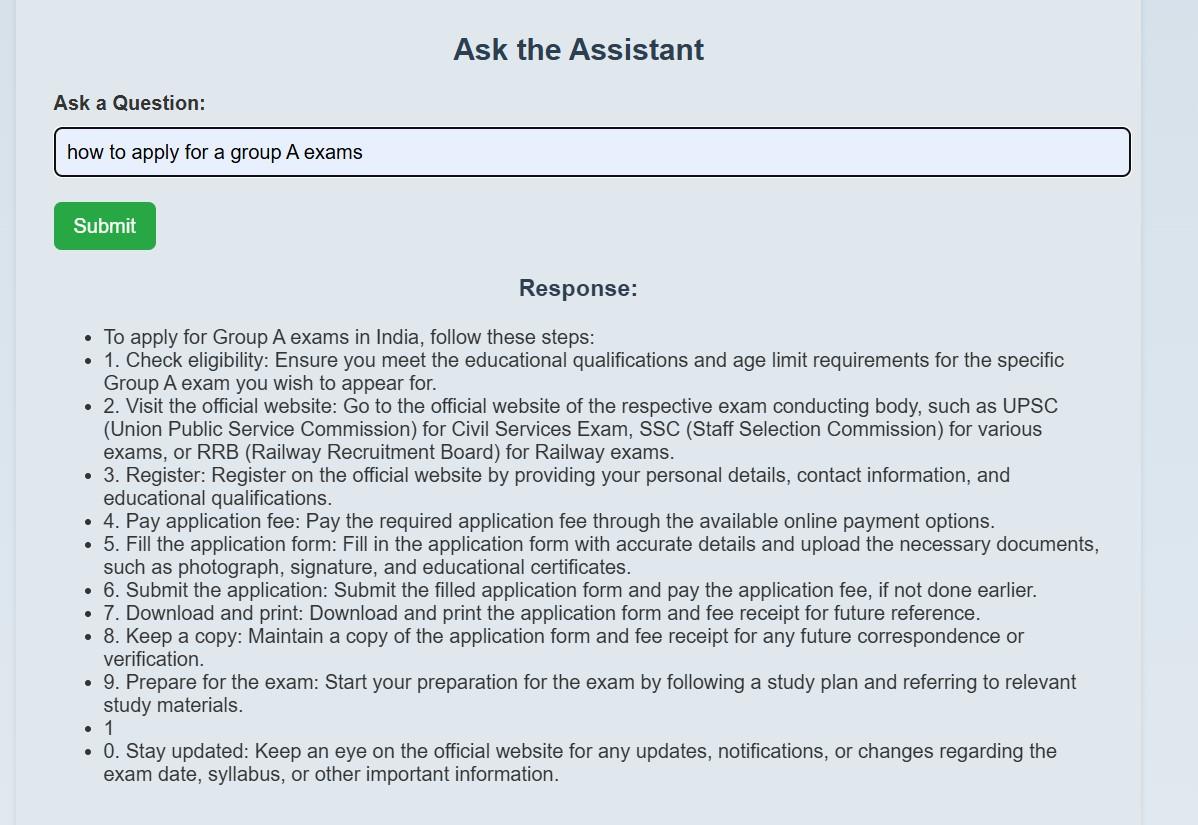
**Dynamic Dashboard:**

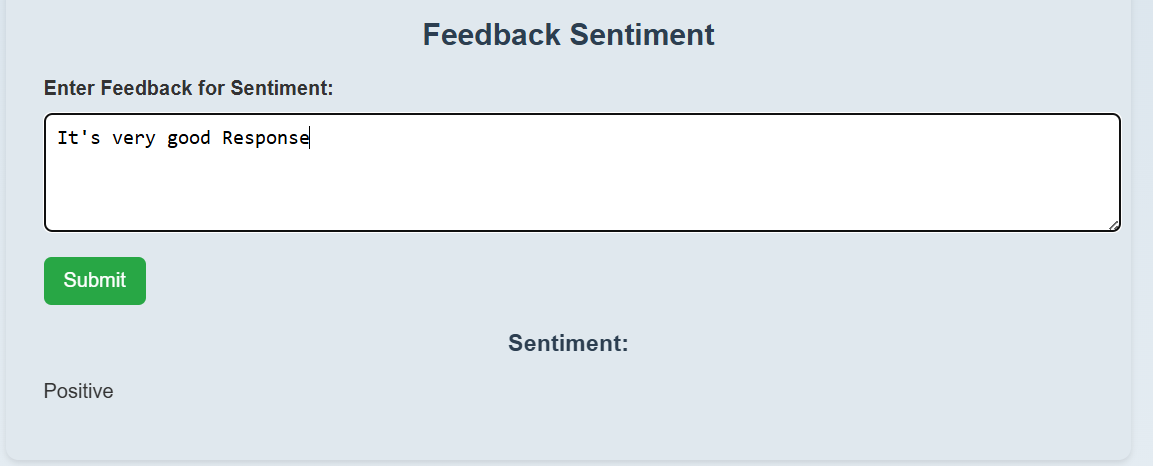
The Dynamic Dashboard in Citizen AI serves as a central hub for government officials to gain real-time insights into citizen feedback and interactions. It visualizes key data points, including the overall citizen sentiment (positive, neutral, negative) derived from submitted feedback. The dashboard also tracks interaction trends over time, showing peak periods of activity. Furthermore, it can display aggregated government service ratings or issues reported by citizens. It transforms raw interaction data into actionable intelligence for a more responsive government.

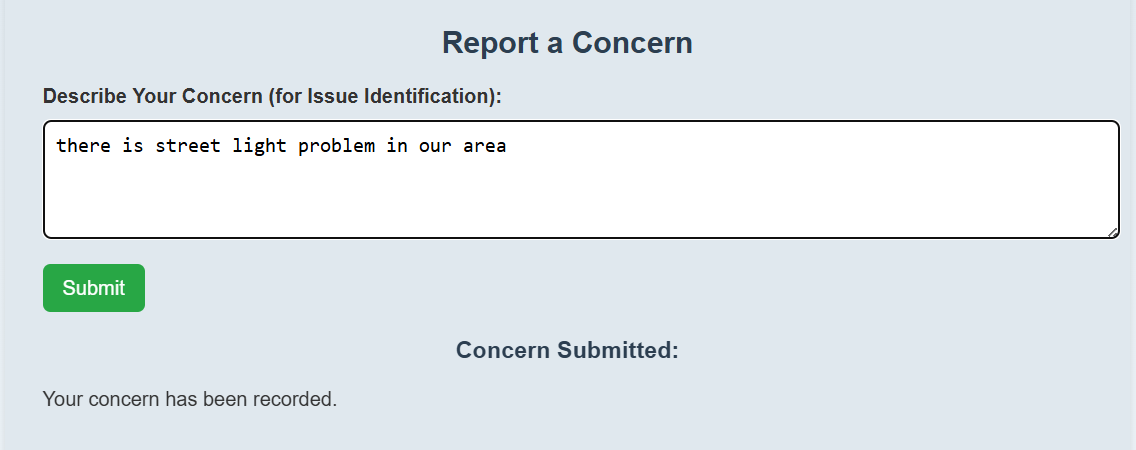
OUTPUT SCREENSHOT

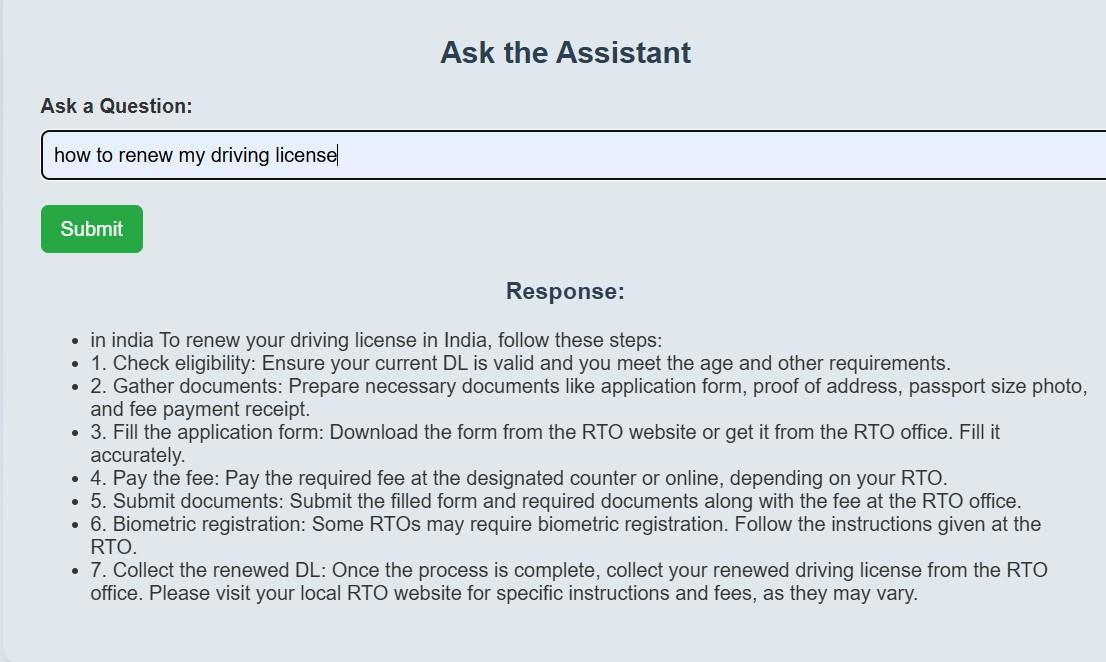












**CONCLUSION**

Your AI-powered CitizenAI platform is designed to enhance interaction, accessibility, and transparency between citizens and government services. By integrating an AI chat assistant, sentiment analysis, concern reporting, and dashboard insights, the platform empowers users to easily access information, provide feedback, and report issues. With a Flask backend and an interactive HTML/CSS frontend, powered by the IBM Granite AI model, your project ensures a user-friendly experience while providing smart and responsive civic engagement tools. This innovative solution simplifies communication and fosters trust, making civic participation more convenient and efficient for all.