Sustainable Smart City Assistant

Open Source AI Companion using IBM Granite LLM

1. INTRODUCTION

1.1 Project Overview

The *Sustainable Smart City Assistant* is an open-source, AI-powered platform designed to deliver contextual, real-time, and eco-centric information about cities and villages. Built using " IBM Granite-3.3-2b-instruct model", The assistant is accessible via a user-friendly Streamlit frontend, backed by a FastAPI server and deployed via a secure ngrok.

1.2 Purpose

the application aims to serve both local citizens and travelers byoffering insights into city conditions, sustainability tips, real-time weather, policy generation, andmore.

Unlike traditional smart city dashboards, this system emphasizes usability for **every citizen**, making it useful for: • * Travelers exploring new places

• * *Daily learners and curious minds*

• * *Environmental enthusiasts*

• * *City planners and feedback contributors*

The assistant brings together AI capabilities and urban information systems into one friendly, conversational interface.

2. IDEATION PHASE

2.1 Problem Statement

Ideation Phase
Define the Problem Statements

Date	14 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	2 Marks

Customer Problem Statement Template:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a city resident	live sustainably and make smart lifestyle choices	I struggle to find relevant, real-time, and personalized data about my city's environment and services	current systems lack AI integration, adaptability, and a unified platform	frustrated and disconnected from efforts to build a greener future
PS-2	a city planner or municipal officer	efficiently manage city resources, infrastructure, and citizen engagement for sustainable growth	I am limited by fragmented data systems, reactive planning, and low public participation	there is no AI-driven assistant to support peaple to find new policy updates	overwhelmed and under-equipped to meet sustainability goals

2.2 Empathy Map Canvas

Ideation Phase
Empathize & Discover

Date	14 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	4 Marks

SAYS: - "I want real-time updates about traffic, air quality, and water usage." - "Why isn't there a single place where I can get all city-related alerts?" - "It would be great to reduce my energy bills and carbon footprint." - "I care about sustainability, but I don't always know what steps to take."	**THINKS:** - "Is my neighborhood doing enough to go green?" - "Are city services using AI to improve our daily lives or just collecting data?" - "I'd love if city planning considered resident feedback more." - "Technology should make city life easier and more sustainable."
DOES: - Checks multiple apps/websites for transport, pollution, and utility data. - Participates in community groups or forums discussing urban issues. - Looks for ways to recycle, conserve energy, or reduce waste. - Uses digital tools to monitor electricity and water usage.	**FEELS:** - Frustrated by fragmented or outdated public data. - Concerned about rising pollution and climate impact. - Empowered when they can contribute to a smarter, greener city. - Curious about how AI can help them live more sustainably.
PAINS: - Lack of centralized, AI-powered guidance on sustainable actions. - Overwhelming or unclear data from various city systems. - Uncertainty about the accuracy or privacy of smart city technologies.	**GAINS:** - Personalized insights on reducing energy usage or travel emissions. - Real-time AI assistant that integrates data from transport, utilities, and environment. - Transparent, ethical AI systems that explain their actions.

2.3 Brainstorming

Ideation Phase
Brainstorm & Idea Prioritization Template

Date	15 jun 2025
Team ID:	LTVIP2025TMID32673
Project Name:	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	4 Marks

Brainstorm & Idea Prioritization Template:

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Problem Statement Selected:

Urban areas are growing rapidly, causing increased environmental challenges, inefficient resource utilization, traffic congestion, and lack of sustainable living practices. The selected problem is to design a Sustainable Smart City Assistant that integrates technology to promote eco-friendly urban living, enhance public services, and reduce environmental impact using data-driven and AI-enabled solutions with the help of IBM granite LLM.

Step-2: Brainstorm, Idea Listing and Grouping

Raw Ideas:

Real-time weather updates
Smart chat assisatnt
smart traffic controll
Water usage tracking and conservation suggestions
Smart grammer cheking for documents
Integration with renewable energy grid
Eco tips to improve nature avareness
feedback collection to estimate real time problems
Smart lighting and utility control in public spaces
Community resource-policy genaration
KPI forcast to estimate energy charges

Grouped Ideas by Theme:

- Environmental Monitoring:

  - Live weather updates
  - eco tips
  - KPI forcast analysis

  -Water usage tracking

- Urban Efficiency:

  - Smart policy genarator
  - Smart traffic control
  - Smart lighting control

- Sustainable Living Support:

  - Renewable energy integration
  - live updates
  - smart chat assistant

- Community Engagement:

  - feedback collection
  - documents correction

Step-3: Idea Prioritization

Idea	Feasibility	Impact	Innovation	Priority
Smart chat assistant	High	High	Medium	High
Live weather report	Medium	High	Medium	High
documents error correction	Medium	High	Medium	High
KPI forcast	Medium	Medium	Medium	Medium	Medium
Eco tips	High	Medium	High	Medium
live city updates	High	Medium	High	High
feedback collection	Medium	High	Low	High
policy genarator	Low	Medium	Medium	High

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

Persona: Riya, a 28-year-old resident of a smart city, environmentally conscious, uses mobile apps frequently.

Stage	Touchpoints	User Actions	User Goals	Pain Points	How Our AI Solves It
1. Awareness	Social Media, City Ads, NGO Campaigns	Hears about the Smart City Assistant through a local awareness drive	Learn how to contribute to sustainability	Too many disconnected apps	Unified AI platform providing guidance on all city services
2. Onboarding	App Store, City Portal, QR Codes	Downloads the app or scans QR on a smart kiosk	Easy sign-up and language-friendly onboarding	Complex forms, English-only interfaces	Multilingual support, voice-based sign-up, social logins
3. First Use	Mobile App, Website, Kiosk, Voice Interface	Asks for pollution levels, water usage, traffic alerts	Get real-time, personalized sustainability insights	Unreliable data or technical jargons	Real-time integrated IoT data, simple visual insights
4. Daily Use	Notifications, Dashboard, Voice Prompts	Follows AI tips for energy saving, gets alerts, submits issues	Improve personal impact, stay informed	Forgetting to check apps or low engagement	Personalized nudges, gamified sustainability points
5. Feedback Loop	Feedback Form, Chatbot, Support Button	Sends feedback about water supply issues	Feel heard, expect a fix	No acknowledgment, unclear resolution	Feedback auto-acknowledged, routed to right department

3.2 Solution Requirement

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	20 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	4 Marks

Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	-
FR-2	User Confirmation	-
FR-3	Dashboard Access	- View energy, water, waste usage - Access personalized AI insights - Multilingual support
FR-4	Real-time Alerts	- Pollution level warnings - Water leakage or outage alerts - Traffic congestion notifications
FR-5	Feedback Submission	- Feedback form interface - Rating service quality - Submit suggestions or issues
FR-6	Admin Panel	- Monitor system performance - Access user feedback - View predictive maintenance alerts

Non-functional Requirements

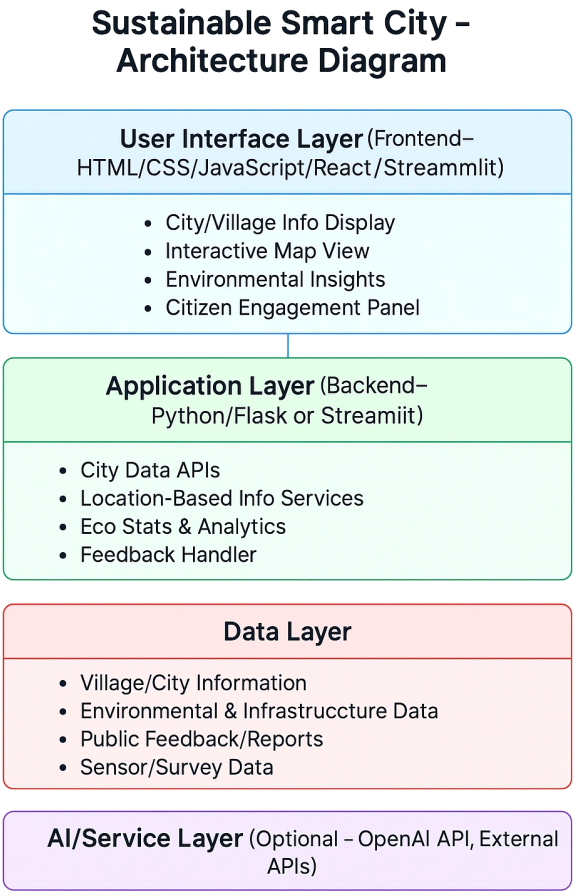
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The interface will be intuitive, responsive, and accessible on mobile, web, and voice platforms.
NFR-2	Security	All user data will be encrypted in transit and at rest. Role-based access control will be enforced.
NFR-3	Reliability	The system will have failover mechanisms and 99.9% uptime.
NFR-4	Performance	The AI assistant will respond to user queries within 2 seconds under average load.
NFR-5	Availability	System will be available 24/7 with monitoring and automatic recovery features.

3.3 Data Flow Diagram

Project Design Phase-II
Data Flow Diagram & User Stories

Date	20 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	4 Marks

Data Flow Diagram (DFD)



Entities:

- Citizens
- City Administrator
- Smart City Database
- AI Assistant Interface

System Processes:

1. Data Collection from feedbacks
2. AI-based Data Processing & Analysis
3. User Request Handling (via streamlit)
4. Feedback Loop to Government Authority
5. Response Generation and Visualization

Data Stores:

- Environmental Data Repository
- User Profile & Preferences
- Feedback Logs
- Usage Statistics

Data Flow Sequence (Narrative)

1. Citizens input requests via webapp
2. The system fetches real-time data from databases.
3. The AI engine processes and returns recommendations .
4. Users receive dashboards or text responses.
5. Feedback is captured and logged for analysis.
6. City authorities access insights and reports for planning.

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Citizen (Mobile/ Web)	Registration	USN-1	As a user, I can register using email and password.	I can log in and access personalized dashboard.	High	Sprint-1
Citizen (Mobile/ Web)		USN-2	As a user, I receive a confirmation email after registration.	I can verify and activate my account.	High	Sprint-1
Citizen (Mobile/ Web)	Social Login	USN-3	As a user, I can register using Google or Facebook.	I can log in via my social account.	Medium	Sprint-2
Citizen (Mobile/ Web)	Dashboard	USN-4	As a user, I can view energy/water/waste usage trends.	Usage data and charts are visible on my dashboard	High	Sprint-2

				.		
Citizen (Mobile/ Web)	Alerts	USN-5	As a user, I get real-time alerts on pollution, outages, traffic.	Notifications appear based on my location.	High	Sprint-3
Citizen (Mobile/ Web)	Feedback	USN-6	As a user, I can submit feedback to authorities through the app.	Feedback is acknowledged and logged in system.	Medium	Sprint-3
City Admin	Monitoring	USN-7	As an admin, I can monitor overall city sustainability metrics.	Admin dashboard shows live analytics.	High	Sprint-2
City Admin	Citizen Feedback	USN-8	As an admin, I can review user-submitted feedback.	Feedback list is sorted by category and date.	Medium	Sprint-3

3.4 Technology Stack

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	20 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	4 Marks

Table-1: Components & Technologies

S.No	Component	Description	Technology

1	User Interface	Citizen interface for web, mobile, chatbot, and voice UIs	HTML, CSS, Streamlit
2	Application Logic-1	Logic to process user requests and route queries	Python (FastAPI), Flask
3	Application Logic-2	Natural language speech-to-text for voice interface	Google colab ,weather API
4	Application Logic-3	AI-based assistant interface for contextual interaction	IBM granite LLM, Dialogflow
5	Database	Stores user profiles, queries, feedback, and usage logs	PostgreSQL
6	Cloud Database	Cloud-hosted backup and analytics-ready DB	Straemlit
7	File Storage	Storage for documents, feedback screenshots, reports	IBM Cloud Storage
8	External API-1	Pollution and weather data retrieval	OpenWeatherMap API, AQICN API
9	External API-2	Address validation and Aadhar verification	UIDAI Aadhar API, Google Maps API
10	Machine Learning Model	Predictive analytics for usage trends and sustainability	Scikit-learn, TensorFlow, XGBoost
11	Infrastructure	Hybrid deployment (local testing and cloud deployment)	Local: Ubuntu VM; Cloud: AWS EC2, GCP VM

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Use of community-supported ML/NLP and web frameworks	Flask, FastAPI, Streamlit
2	Security Implementations	Authentication, encryption, API key protection, access control	, HTTPS, OAuth2, IAM, SHA-256, OWASP-10
3	Scalable Architecture	Microservices with containerization for flexible scaling across services	REST APIs
4	Availability	Multi-zone cloud	github

		with load balancer, auto-healing groups	

4. PROJECT DESIGN

4.1 Problem Solution Fit

Project Design Phase
Problem – Solution Fit Template

Date	14 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	2 Marks

Problem – Solution Fit Canvas

Section	Details
Target Group / User	Urban city planners, municipal authorities, residents of smart cities, sustainability consultants, and civic tech startups.
Current Behavior / Habits	Urban systems often operate in silos (e.g., waste, traffic, water), leading to inefficiencies. Citizens lack unified access to information and personalized suggestions on sustainability actions.
Problem(s) Observed	- Lack of real-time data integration across city services. - Inefficient energy/water usage. - Citizens unaware of their environmental impact. - Poor feedback loops between cities and residents.
Why is this a Problem?	Leads to resource wastage, low citizen engagement in sustainability goals, increased carbon footprint, and difficulty in achieving SDG targets and smart city KPIs.
Existing Alternatives	- Standalone apps for energy, transport, or waste management. - Manual reporting dashboards. - Government helplines. - Legacy GIS and ERP systems in

	municipalities.
Problems with Alternatives	- Disconnected systems and data. - No AI-driven insights or automation. - Limited real-time interactivity. - Low user-friendliness and poor adoption by citizens.
Proposed Solution	A conversational AI platform that unifies city service data (IoT, GIS, ERP) and provides real-time recommendations, alerts, insights, and predictive analytics to both citizens and city administrators.
Unique Value Proposition	- One AI assistant for all smart city needs. - Personalized sustainability coaching. - Seamless citizen-government communication. - Interoperability with existing smart infrastructure.
Solution Benefits	- Encourages greener lifestyles through personalized nudges. - Enables predictive maintenance and smarter planning. - Increases civic engagement. - Reduces operational costs.
Solution Adoption Channels	- Integration into official city apps/websites. - Smart kiosks, digital billboards. - Voice assistants (Alexa, Google Home). - Collaborations with green startups and NGOs.

4.2 Proposed Solution

Project Design Phase
Proposed Solution Template

Date	14 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description

1	Problem Statement (Problem to be solved)	Smart cities lack integrated, intelligent systems that can provide real-time, personalized sustainability guidance to citizens while supporting city planners with predictive analytics and cross-domain insights, leading to inefficiencies, low engagement, and poor resource optimization.
2	Idea / Solution description	The solution is an AI-powered assistant that consolidates data from smart city infrastructure (IoT sensors, GIS, utilities) to provide real-time alerts, sustainability recommendations, urban insights, and two-way communication between city authorities and residents. It can be accessed via mobile apps, kiosks, or voice interfaces.
3	Novelty / Uniqueness	- Unified conversational AI for all smart city services. - Personalized sustainability insights using granite LLM - Integration across departments (traffic, waste, energy) via interoperable APIs. - Predictive analytics for city maintenance and planning.
4	Social Impact / Customer Satisfaction	- Increases citizen awareness and participation in sustainability goals. - Improves quality of urban life through efficient resource usage. - Provides faster access to services and information. - Empowers marginalized groups by offering multilingual, accessible AI interfaces.

5	Business Model (Revenue Model)	- municipalities with tiered pricing based on population size. - Licensing to private smart city consultants and developers. - Data analytics dashboards sold as premium features. - Grants and public-private partnerships for smart infrastructure development.

4.3 Solution Architecture

Project Design Phase
Solution Architecture

Date	14 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	4 Marks

Solution Architecture Description

The Sustainable Smart City Assistant AI is designed to provide a centralized, intelligent platform that connects urban infrastructure systems with residents and administrators. It enables personalized sustainability insights, operational efficiency, and proactive urban management.

Architecture Overview

Component	Description
User Interfaces	Web, mobile apps, WhatsApp chatbot, and

	city dashboards.
Data Sources	IoT sensors (air quality, water flow, waste bins, traffic), GIS mapping systems, utility usage data (electricity, water), citizen feedback systems.
AI/ML Engine	- Predictive analytics for energy, water, and waste usage. - Pattern recognition in citizen behavior. - Personalized eco-friendly suggestions using NLP.
Data Integration Layer	granite LLM and middleware that integrate with existing municipal systems (ERP, SCADA, CRM) to unify data across departments.
Backend Services	Microservices for alert generation, recommendation engines, user management, real-time analytics, and reporting.
Cloud Infrastructure	Scalable deployment on streamlit using containerized services (Docker, Kubernetes), real-time databases, and secured storage systems.
Security & Compliance	Role-based access control, data encryption, GDPR compliance, government digital governance standards, and periodic auditing.

Solution Features

- Unified conversational interface for all citizen needs

- Real-time alerts on traffic, pollution, outages, etc.

- Eco tips dashboards with AI-based insights

- Feedback loop between citizens and government bodies

- Predictive weather reports

- Multilingual and inclusive AI interfaces

-documents or file purification

-KPI forcast estimation

Development Phases

Phase	Description
Phase 1	Requirement analysis, stakeholder consultation, and prototype of AI assistant
Phase 2	Backend integration with smart city data sources and GIS/IoT APIs
Phase 3	AI/ML training using collected data,

	development of personalized insight engine
Phase 4	Front-end interfaces (apps, dashboards, kiosks), testing, and feedback loops with pilot communities
Phase 5	Final deployment, scale-up to additional cities, and continuous monitoring/improvement

Solution Requirements

- IBM granite model

- google colab cloud

- Secure IoT data pipelines

- github for project automation

- Municipal cooperation for API/data access

- Citizen onboarding and education initiatives

Architecture Diagram

Citizens / Planners

|

Conversational UI <--- (straemlit)

|

AI & Personalization
(Granite Models, Analytics)

|

Data Integration
(FastAPI, Utilities)

|

Cloud Infrastructure

(Sql Storage, Compute, API)

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Agile Sprint Planning

Date	18 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Max marks	5 marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Integration	USN-1	As a developer, I can collect and load smart city data (traffic, energy, waste).	3	High	Data Engineering Team
Sprint-1	Data Preprocessing	USN-2	As a data scientist, I can clean and preprocess the data for training.	3	High	AI Team
Sprint-1	Model Design	USN-3	As an AI lead, I can define an AI model	2	Medium	AI Team

			for citizen engagement & resource usage.			
Sprint-1	Dashboard Setup	USN-4	As a developer, I can create HTML UI for displaying insights.	2	Medium	UI/UX Devs
Sprint-2	Model Training & Testing	USN-5	As a data scientist, I can train the ML model and test performance.	5	High	AI Team
Sprint-2	Integration	USN-6	As a dev, I can integrate the model with the frontend dashboard.	3	High	Full Stack Team
Sprint-2	Deployment	USN-7	As an engineer, I can deploy the assistant via Flask to a server.	5	High	DevOps Team
Sprint-2	Feedback Logging	USN-8	As a user, I can give feedback to the city authority through chatbot.	2	Medium	Backend Devs

Project Tracker, Velocity & Burndown Chart (4 Marks)

Sprint	Total Story Points	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date
Sprint-1	10	15 jun 2025	21 jun 2025	10	27 jun 2025
Sprint-2	15	21 jun 2025	27 jun 2025	15	27jun 2025

Velocity Calculation

Total Story Points: 25

Total Sprints: 2

Velocity = 25 ÷ 2 = 12.5 story points per sprint

Daily Average Velocity = 12.5 ÷ 6 = 2.08 story points/day

Suggested Burndown Chart Data

Day	Planned SP Remaining	Actual SP Remaining
Day 1	25	25
Day 2	20	22
Day 3	15	18
Day 4	10	13
Day 5	5	5

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Functional & Performance Testing Template

Model Performance Test

Date	16 jun 2025
Team ID	LTVIP2025TMID32673
Project Name	Sustainable Smart City Assistant AI by using IBM granite LLM
Maximum Marks	

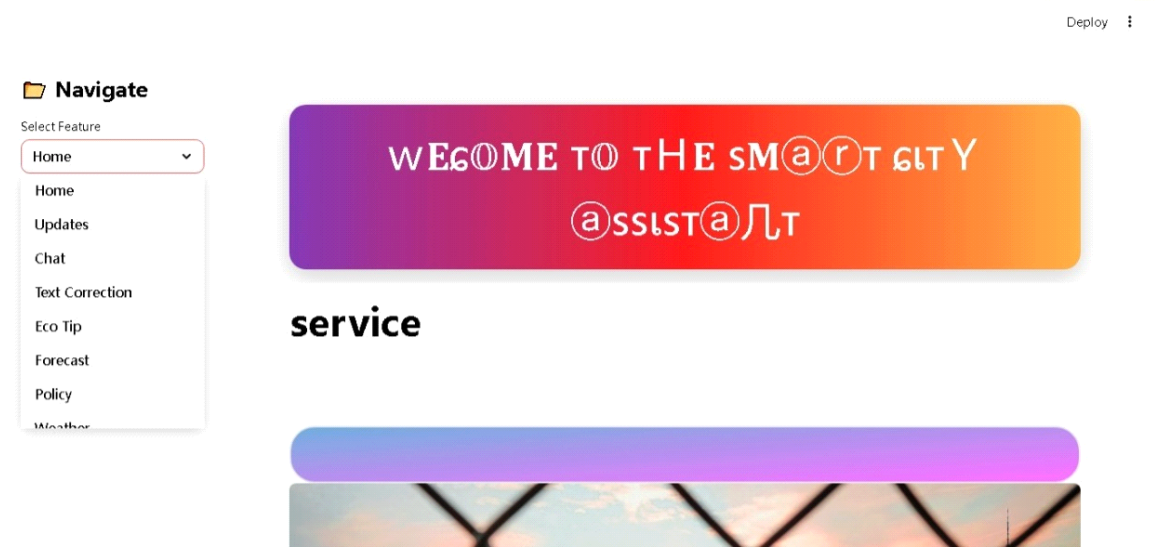
Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	Valid inputs accepted, errors for invalid	Pass
FT-02	Number Input Validation (e.g., word count, size, rooms)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	Accepts valid values, error for invalid	Pass
FT-03	Content Generation (e.g., blog, resume, design idea)	Provide complete inputs and click "Generate"	Correct content is generated based on input	Correct sustainability suggestions are shown	Pass

FT-04	Colab Connection Check	Check if ngork link is correct and model responds	/chat responds successfully	connects successfully with valid key	Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	Average response time 2.5 seconds	Pass
PT-02	Speed Test	Send multiple calls at the same time	stable,should not slow down	Consistent performance across requests	Pass
PT-03	File Upload Load Test (e.g., PDFs)	Upload multiple PDFs and check processing	Should work smoothly without crashing	Handled up to 50 files without lag	Pass

7. RESULTS

7.1 Output Screenshots

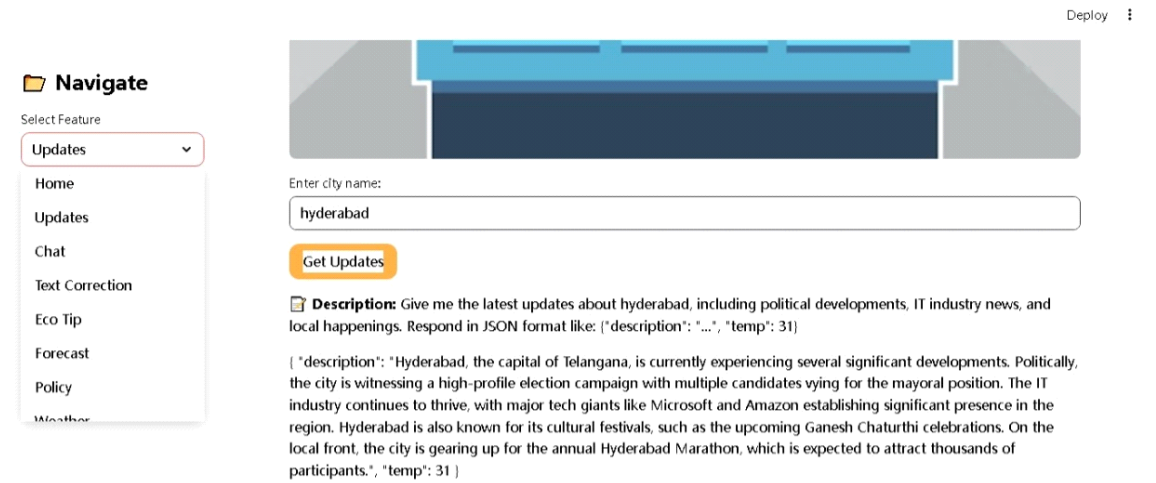


City Updates

* Fetch latest info like weather, news, temperature

* Input: City name

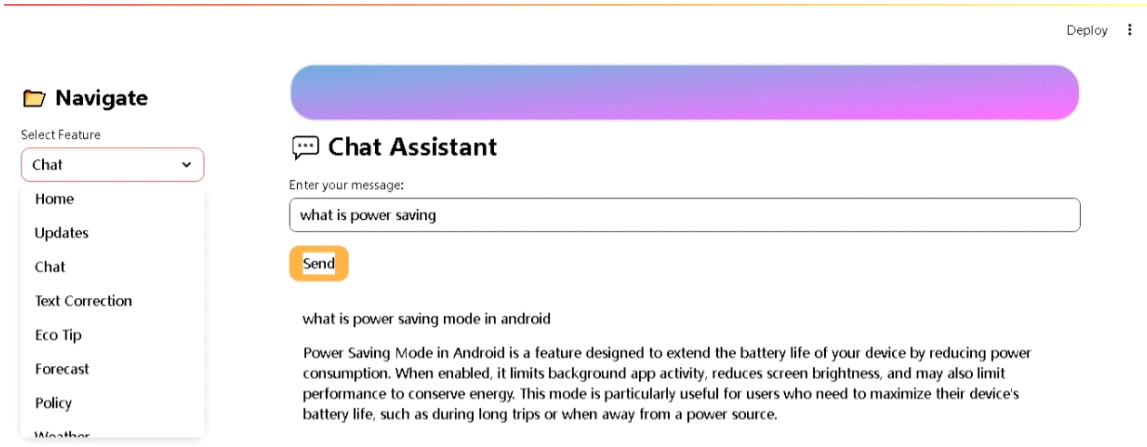
* Output: JSON response with `description`, `temp`



💬 Chat Assistant

* Ask anything from tourist help to culture tips

* Output: Friendly and localized answer from AI



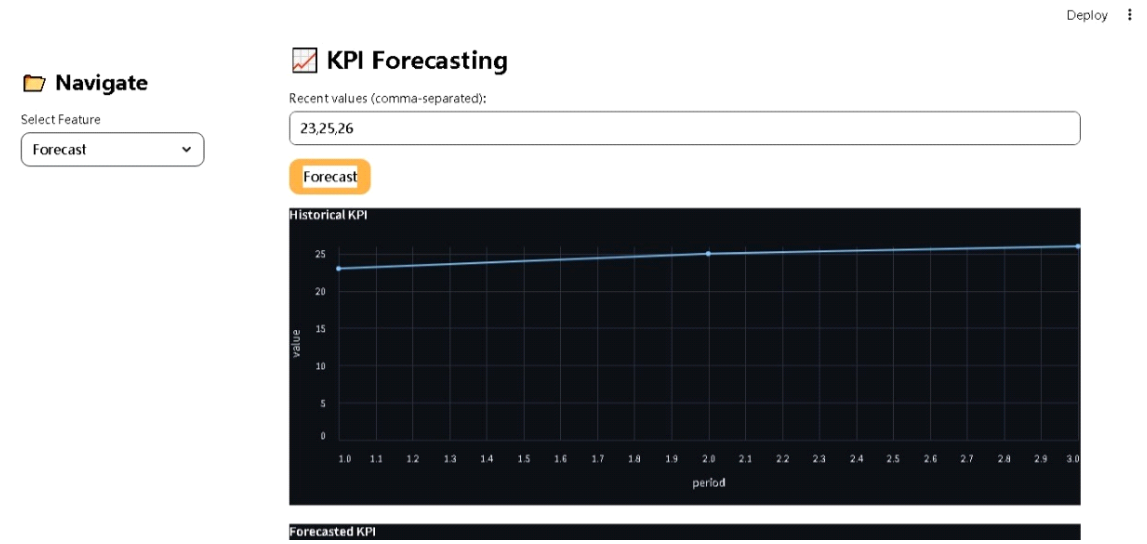
🌿 Eco Tips

* Actionable, emoji-enhanced sustainability ideas ,Example: "Plant herbs in your balcony to save trips to the store!

📈 KPI Forecasting

* Smart insights into trends like energy use, waste

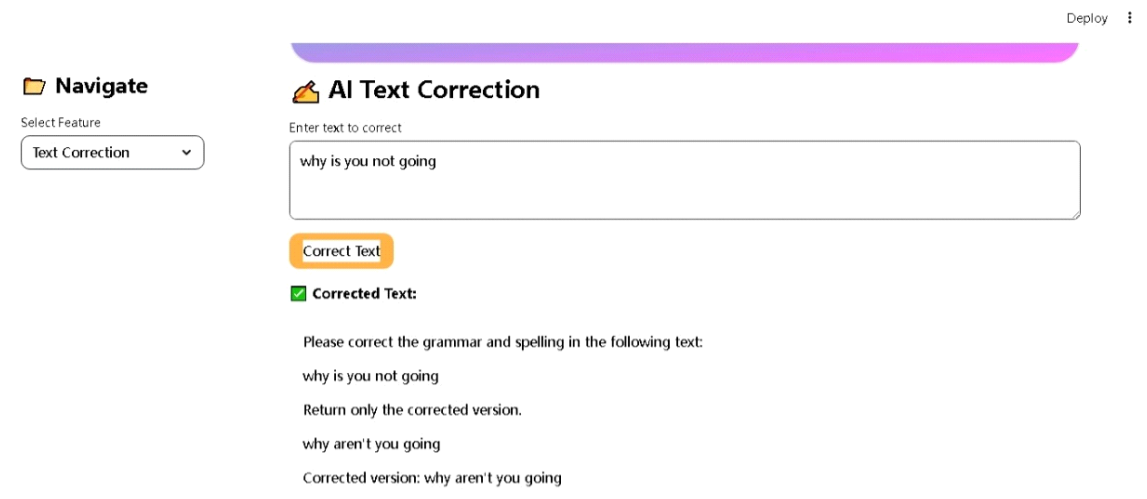
* Charts rendered using Altair



✍️ Grammar Correction

* AI-enhanced proofreading of input content

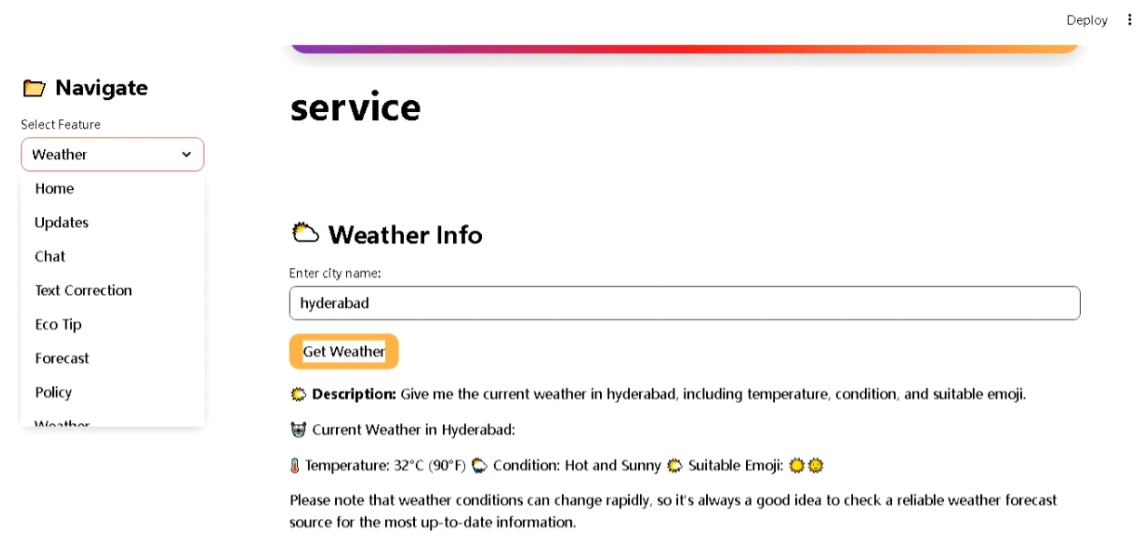
* Useful for announcements or blogs



⛅ Weather Report

* Humanized weather overview with emojis

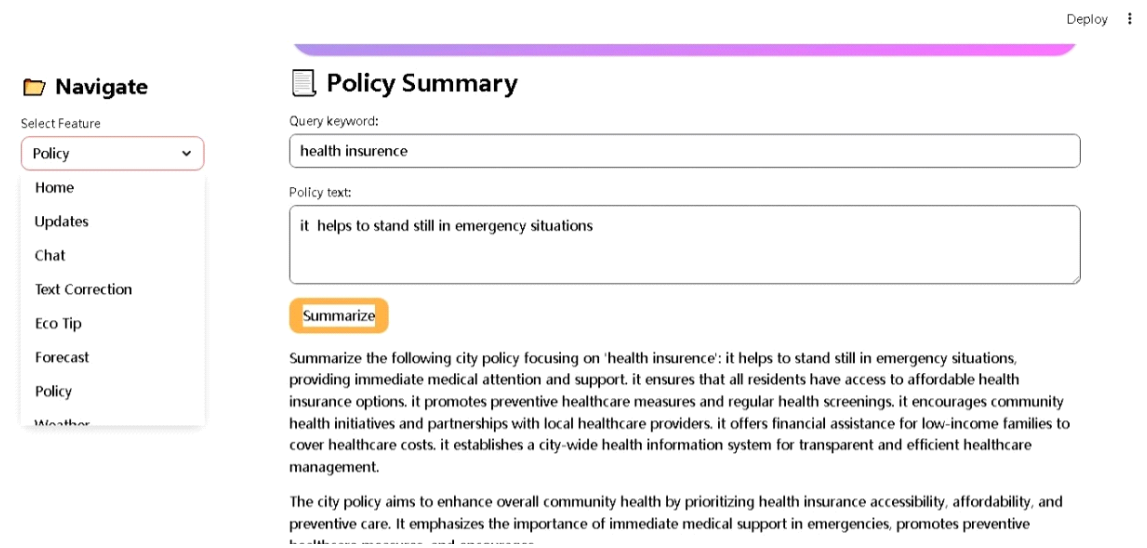
* Helps travelers or event planners



📃 Policy Generator

* Converts dense policies into quick summaries

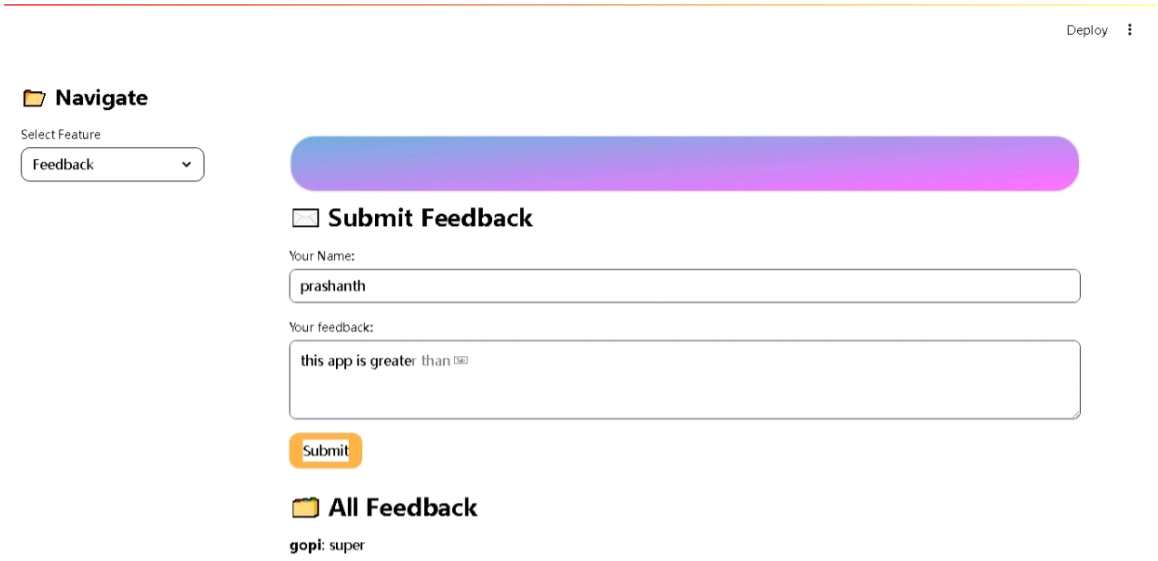
* Eases understanding for common citizens



📩 Feedback System

* Store and view open-ended feedback

* Enables community Participation



8. ADVANTAGES & DISADVANTAGES

✅ Advantages

No.	Advantage	Explanation
1	Centralized Smart City Interface	Citizens can access all city services and sustainability information through one AI-powered platform.
2	Real-Time Data & Alerts	Provides immediate insights and alerts (e.g., pollution, traffic, water usage) from IoT and sensor networks.
3	Personalized Recommendations	AI tailors tips and information to each user, enhancing engagement and behavioral change.
4	Multilingual & Inclusive Access	Voice + text interfaces support multiple Indian languages for diverse user groups.
5	Cost & Resource Optimization for City Administrators	Enables predictive maintenance, efficient planning, and automation of feedback loops.
6	Scalable Architecture	Designed to expand across cities and integrate with existing infrastructure via APIs.
7	Increased Citizen Participation	Gamified eco-scoring and feedback features motivate long-term user engagement.
8	Environmentally Impactful	Supports SDGs by encouraging reduced consumption, emissions, and better civic behavior.

⚠️ Disadvantages

No.	Disadvantage	Mitigation Strategy
1	High Initial Setup & Integration Cost	Leverage government smart city grants and public-private partnerships
2	Dependency on IoT Infrastructure Availability	Start with cities with basic infrastructure; use hybrid data sources
3	Data Privacy and Security Concerns	Apply strong encryption, role-based access, and compliance with data regulations
4	User Reluctance to Adopt New Tech	Provide onboarding help, community workshops, and voice-based accessibility
5	Maintenance and Continuous Model Training Required	Automate retraining schedules; partner with research institutions for model upkeep

9. CONCLUSION

The *Sustainable Smart City Assistant* is a citizen-focused AI application that blends technologyandsocial good. Built as part of the SmartBridge internship using IBM Granite, it serves as an educational, informative, and eco-conscious digital companion. By using natural language models todrive human interaction, it simplifies civic learning and fosters local engagement. Whether you're a traveler, a curious citizen, or someone passionate about the environment, this assistant gives you **just enough knowledge to care and act smartly..

10. FUTURE SCOPE

Area	Future Scope
1. Geographic Expansion	Extend the assistant's deployment from pilot smart cities to tier-2 and tier-3 cities across India and globally.
2. Feature Enhancements	Integrate more city departments (e.g., healthcare, emergency response, public transport planning) into the AI ecosystem.
3. Multimodal Interaction	Expand accessibility through voice-first interaction, AR/VR city visualizations, and wearable device compatibility.
4. Personal Sustainability AI Coach	Use AI to offer hyper-personalized advice, nudges, and goal tracking for users to reduce their carbon footprint.
5. Predictive Governance	Enable municipal bodies to proactively manage urban challenges (e.g., water shortage, traffic spikes) using AI forecasts.
6. Integration with Smart Homes	Link with home IoT devices for holistic monitoring (e.g., smart meters, HVAC, solar panels).
7. Open Data Platform	Create a public, anonymized data dashboard for researchers, NGOs, and startups to innovate further.
8. Public Engagement & Gamification	Introduce city-wide competitions, green leaderboards, and rewards to foster mass participation.
9. AI-driven Policy Feedback Loop	Analyze citizen behavior and feedback to recommend policy adjustments to urban planners.

11. APPENDIX

Source Code(if any) :

main.py – Flask Backend

"import os

from fastapi import FastAPI, HTTPException, Depends

from fastapi.middleware.cors import CORSMiddleware

from pydantic import BaseModel

from sqlalchemy import create_engine, Column, Integer, String

from sqlalchemy.ext.declarative import declarative_base

from sqlalchemy.orm import sessionmaker

from dotenv import load_dotenv

import requests

import json

 # Load environment variables

load_dotenv()

COLAB_MODEL_API = os.getenv("COLAB_MODEL_API") # e.g., https://xxxx.ngrok-free.app/chat

DATABASE_URL = "sqlite:///./app.db"

 # Database setup

engine = create_engine(DATABASE_URL, connect_args={"check_same_thread": False})

```
SessionLocal = sessionmaker(bind=engine, autoflush=False, autocommit=False)

Base = declarative_base()

class Feedback(Base):

    __tablename__ = "feedback"

    id = Column(Integer, primary_key=True, index=True)

    user_id = Column(String, nullable=False)

    message = Column(String, nullable=False)

Base.metadata.create_all(bind=engine)

# FastAPI app

app = FastAPI()

app.add_middleware(

    CORSMiddleware,

    allow_origins=["*"],

    allow_methods=["*"],

    allow_headers=["*"],

)

# DB Dependency

def get_db():

    db = SessionLocal()
```

```
    try:
        yield db
    finally:
        db.close()
# Models
class Prompt(BaseModel):
    prompt: str
class TextInput(BaseModel):
    text: str
class FeedbackModel(BaseModel):
    user_id: str
    message: str
class PolicyRequest(BaseModel):
    query: str
    policy_text: str
class AnomalyPayload(BaseModel):
    data: list[float]
# Model call helper
def call_model(prompt: str):
```

```
    try:
        response = requests.post(COLAB_MODEL_API, json={"prompt": prompt}, timeout=30)
        response.raise_for_status()
        return response.json().get("text", "No response")
    except Exception as e:
        raise HTTPException(status_code=503, detail=f"Model call failed: {e}")

# Endpoints

@app.post("/chat")
def chat_endpoint(payload: Prompt):
    return {"response": call_model(payload.prompt)}

@app.post("/text-correction")
def correct_text(payload: TextInput):
        prompt = f"Please correct the grammar and spelling in the following text:\n\n{payload.text}\n\nReturn only the corrected version."
    return {"corrected_text": call_model(prompt)}

@app.get("/eco-tips")
def eco_tips():
     prompt = "Give a unique and practical eco-friendly tip for urban life. Include an emoji and short explanation."
    return {"tip": call_model(prompt)}
```

```
@app.post("/forecast-kpi")

def forecast_kpi(payload: AnomalyPayload):

    prompt = f"Given this historical data: {payload.data}, forecast the next 3 values. Return only a JSON list like: [x, y, z]"

    return {"forecast_next_3_periods": call_model(prompt)}

@app.post("/submit-feedback")

def submit_feedback(fb: FeedbackModel, db=Depends(get_db)):

    entry = Feedback(user_id=fb.user_id, message=fb.message)

    db.add(entry)

    db.commit()

    db.refresh(entry)

    return {"message": "Feedback submitted"}

@app.get("/feedback")

def get_feedback(db=Depends(get_db)):

    items = db.query(Feedback).all()

    return [{"user_id": i.user_id, "message": i.message} for i in items]

@app.post("/policy-summary")

def policy_summary(req: PolicyRequest):

    prompt = f"Summarize the following city policy focusing on '{req.query}':\n{req.policy_text}"

    return {"summary": call_model(prompt)}
```

```
@app.get("/weather")

def weather(city: str):

    prompt = f"Give me the current weather in {city}, including temperature, condition, and suitable emoji."

   return {"weather": {"description": call_model(prompt)}}

@app.get("/updates")

def city_updates(city: str):

   prompt = (

       f"Give me the latest updates about {city}, including political developments, IT industry news, and local happenings. "

       "Respond in JSON format like: {\"description\": \"...\", \"temp\": 31}"

   )

   try:

       raw = call_model(prompt)

       try:

           parsed = json.loads(raw) if isinstance(raw, str) else raw

       except json.JSONDecodeError:

           parsed = {"description": raw, "temp": None}  # fallback

       return {"updates": parsed}

   except Exception as e:
```

```
        raise HTTPException(status_code=500, detail=f"AI error: {e}"

@app.get("/")

def root():

    return {"message": "Smart City Assistant Backend Running"}"
```

requirements.txt/backend

```
fastapi,uvicorn,pydantic,requests,streamlit,pandas
```

Dataset Link

sqlite:///./app.db

GitHub & Project Demo Link

https://github.com/praSHAnTH630490/Smart-City-assistant

-