SUSTAINABLE SMART CITY ASSISTANT

-USING IBM CLOUD

Team Member Name	Roll Number
Janga Sai Sudheshna	22P31A4216
Gunnam Bhavya Sri	22P31A4214
K. Samitha Naga Lakshmi Devi	22P31A4221
K. Alekya Devi	22P31A4219

1.INTRODUCTION

1.1 Project Overview

The **Sustainable Smart City Assistant** is an AI-driven web platform designed to support environmentally sustainable development in rapidly urbanizing areas. Urban populations are increasingly burdened with challenges like inefficient recycling, poor air and water quality, limited access to real-time data, and disorganized infrastructure growth. This assistant acts as a digital guide, empowering both citizens and decision-makers by providing tools to monitor, analyse, and improve city sustainability through smart technology and artificial intelligence.

The system offers **five powerful modules**: ♻️ **Recycle Advisor** gives users personalized recycling tips based on location and material type; 📊 **City Health Dashboard** visualizes real-time health and pollution metrics from urban sensors; 📘 **RAG-Based Problem Solver** simplifies long policy documents or urban challenges into concise summaries; 💠 **AI Image Generator** creates visual concepts for sustainable development ideas; 🏙️ **City Comparison Tool** allows side-by-side analysis of two cities using key indicators such as pollution levels, green space coverage, population density, and smart initiatives..

Together, these modules form an interactive and insightful platform that promotes civic engagement, environmental awareness, and data-driven governance. By enabling users to visualize problems, compare city performance, and explore sustainable solutions, the **Sustainable Smart City Assistant** supports the vision of building smarter, cleaner, and more inclusive urban spaces for future generations.

1.2 Purpose

The primary purpose of the **Sustainable Smart City Assistant** is to **empower citizens, urban planners, and policymakers** with AI-driven tools that promote informed decision-making, enhance public engagement, and accelerate the adoption of sustainable practices in urban environments. As cities grow rapidly, the need for smart solutions to address environmental, health, and infrastructure challenges becomes more urgent.

This project aims to **bridge the gap between complex urban data and actionable insights** by offering intuitive modules such as recycling guidance, real-time city health monitoring, AI-based problem-solving, city comparison, and sustainable design visualization. These features simplify access to important information, encourage responsible behaviour, and foster collaboration between citizens and city administrators.

Ultimately, the Sustainable Smart City Assistant is built to **support the development of smarter, greener, and more liveable cities** by harnessing the power of artificial intelligence, real-time data, and interactive visualization.

2 IDEATION PHASE

2.1 Brainstorming & Idea Prioritization

Date	20 June 2025
Team ID	LTVIP2025TMID29572
Project Name	Sustainable Smart City Assistant
Maximum Marks	4 Marks

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

Before You Collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

- Team Gathering:

Participants:
- Janga Sai Sudheshna

-K. Samitha Devi

-G. Bhavya Sri

- K. Alekya Devi

Purpose:
- Define clear roles: backend, frontend, sustainability logic, image generation, dashboard UI.

- Set the Goal:

Build a unified AI-powered web assistant that helps citizens and city officials in a sustainable smart city.

The assistant should provide:
- Eco-friendly advice for waste management (recycling).
- Real-time city health dashboards (AQI, water, energy, waste).
- Smart city image generation from text.
- Comparison Between Two Cities Using Retrieval Augmented Generation (RAG)

- □ Learn How to Use the Facilitation Tools:

Use collaboration platforms like Mural, GitHub, VS Code Live Share, and Stream lit Cloud for distributed development.

□ Define Your Problem Statement

□ How Might We:
Citizens in growing urban areas face challenges related to sustainability, such as inefficient recycling, rising pollution, and uneven city development. They often lack access to real-time environmental data and clear comparisons across cities. This gap limits their ability to make informed decisions and contribute effectively to sustainable living.

□ Brainstorm

💡 Write down any ideas that come to mind for your team roles or modules.

- **K. Samitha Naga Lakshmi** suggested integrating large language models like IBM Granite for providing intelligent recycle suggestions. Additionally, she recommended comparing two cities using Retrieval-Augmented Generation (RAG) for performance benchmarking and deploying the backend using Fast API for scalability. and to Develop a RAG-based solution generator, where users can input city-related sustainability problems (e.g., air pollution, traffic congestion), and the system suggests practical solutions aligned with sustainable development goals.

- **Janga Sai Sudheshna** proposed creating a real-time city monitoring dashboard that includes air quality index (AQI), energy consumption, and water usage. She emphasized the use of Altair and Stream lit for interactive and visually appealing data charts. Sudheshna also suggested adding a text-to-image generation module that visualizes smart city concepts using Stable Diffusion, served via a Fast API endpoint.

- **Gunnam Bhavya Sri** offered to design and develop an interactive front-end interface for the Smart City Assistant using Stream lit, focusing on user experience and modular navigation.

- **Alekya Devi** took charge of gathering all functional and non-functional requirements related to the project to ensure alignment with sustainable smart city goals and technical feasibility

☐ Key Rules of Brainstorming

- Stay aligned with sustainability and citizen engagement goals.

- Encourage even wild futuristic ideas (drone delivery, solar buses).

- Don't criticize ideas — document everything for evaluation.

☐ Step 3: Idea Prioritization

📊 Criteria: Feasibility, Impact, Innovation, Ease of Integration

✅ Prioritized Ideas:

- ♻ Recycle Advisor: 4.5/5 (High Priority)
- 🏢 Health Dashboard: 4/5 (High Priority)
- 📄 Problems of citizens & Solutions Using RAG: 4.8/5 (High Priority)
- 🎨 Image Generator: 4.5/5 (High Priority)
- 🧑‍🔬 Comparison Between Two Cities: 4.5/5 (High Priority)

☐ Tech Stack

- Frontend: Stream lit

- Backend: Python + Hugging Face Transformers + IBM Granite / FLUX

- Data Visualization: Altair
- Storage/Docs: Using Own Raw Data
- Deployment: Fast API

Project Goals

- Promote sustainability through intelligent, real-time assistance
- Promote Smart Waste Management through Recycle Advisor
- Enable Visual Imagination of Future Cities via AI-Powered Image Generation
- Integrate with external LLMs/APIs to demonstrate extensibility
- Benchmark Urban Development through Smart City Comparison
- Provide Actionable Solutions to City-Level Sustainability Challenges
- Deliver Real-Time Urban Insights with an Interactive City Health Dashboard

2.2 PROBLEM STATEMENT

Customer Problem Statement by Functionality:

Problem Statement (PS-1)	
I am	A citizen living in a rapidly growing urban area
I’m trying to	Live sustainably and contribute positively to my city’s development
But	I face challenges like inefficient recycling, rising pollution, and uneven access to city resources
Because	I lack access to real-time environmental data and city comparison tools
Which makes me feel	Disconnected, uninformed, and unable to make meaningful, eco-conscious decisions

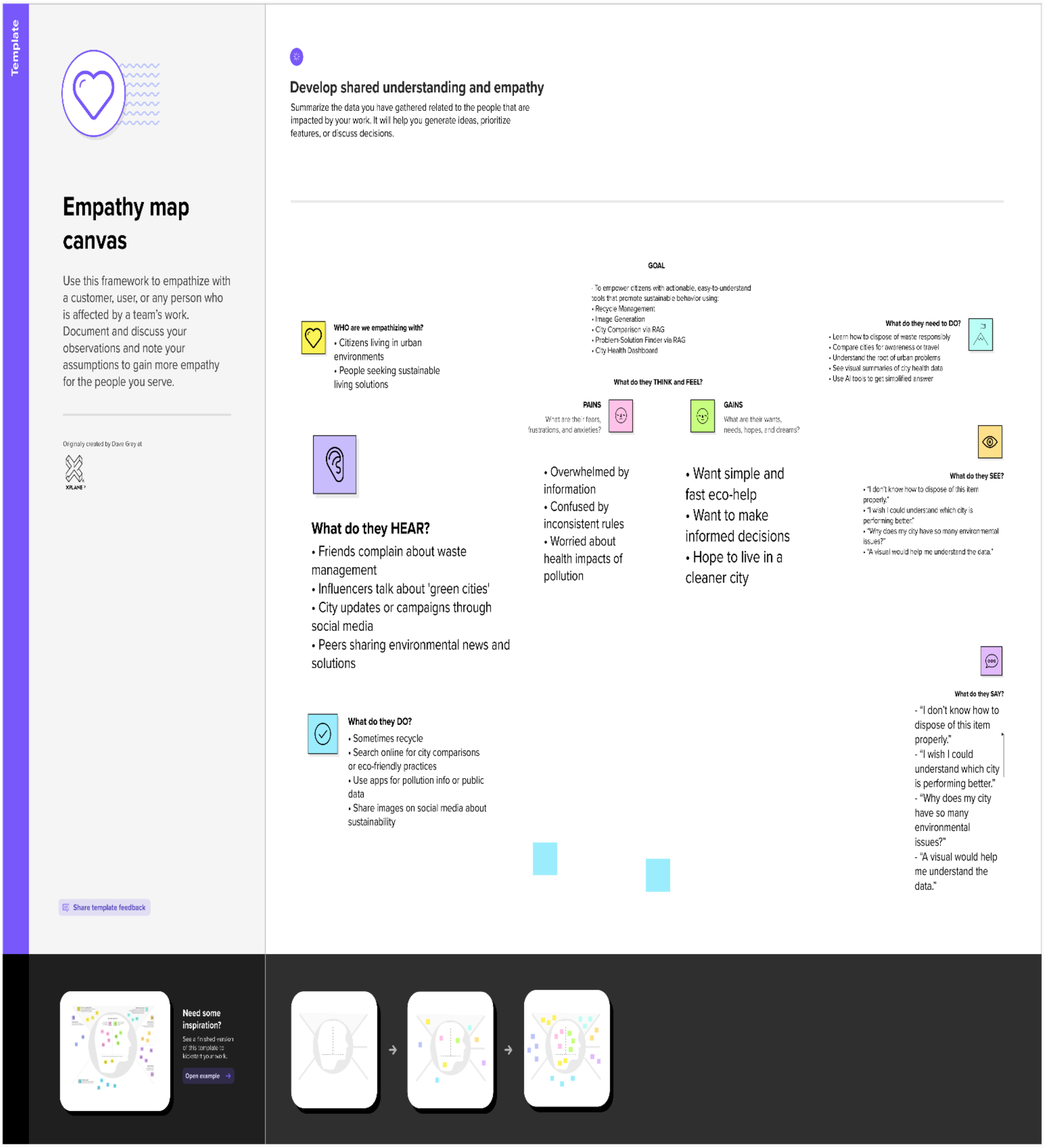
Problem Statement (PS-2)	
I am	A resident trying to dispose of household waste responsibly
I’m trying to	Find the correct way to recycle different types of waste
But	I often don’t know which materials go into which bins.
Because	There is no easily accessible tool that gives personalized recycling suggestions
Which makes me feel	Confused and discouraged from recycling correctly

Problem Statement (PS-3)	
I am	A curious citizen interested in urban innovation
I’m trying to	Visualize what a sustainable future city could look like
But	I have no way to generate visuals or models based on my ideas or text inputs

Because	There are no tools that use AI to convert my imagination into visual representations
Which makes me feel	Limited in creativity and less engaged in future city planning

Problem Statement (PS-4)	
I am	A researcher or city planner
I'm trying to	Understand how my city compares to others in terms of sustainability
But	There's no platform that provides side-by-side comparisons of urban health indicators
Because	Data is scattered and difficult to interpret without technical tools
Which makes me feel	Frustrated and uncertain about prioritizing improvements

2.3 CUSTOMER EMPATY MAPS



3. REQUIREMENT ANALYSIS

3.1 Functional & Technical Requirements Analysis

🌍 Project Overview

The Sustainable Smart City Assistant is an AI-powered platform designed to empower citizens, urban planners, and local authorities with tools that promote environmental sustainability, informed decision-making, and smart urban living. Built as a modular web-based assistant, it integrates generative AI, data analytics, and user-friendly interfaces to address key urban challenges.

? Functional Modules

- ? Recycle Management Advisor – Guides eco-friendly waste disposal and recycling practices.
- ? AI Image Generator – Generates visual representations of sustainable city environments.
- ? Problem & Solution Finder (RAG-based) – Provides AI-generated insights and document-based responses to urban issues.
- ? City Health Dashboard – Allows comparison of environmental health metrics across city
- ? City Comparison Tool – Compares two cities side-by-side based on key sustainability indicators like air quality, traffic levels, waste management, and public health metrics.

3.1 📄 **Customer Journey Map – Sustainable Smart City Assistant**

No.	Module	User Action 1	User Experience Before	System Action 1	User Experience After
1	♻ Recycle Management Advisor	User enters a waste item (e.g., plastic cup)	Curious	System suggests how to dispose or recycle it responsibly	Informed & Motivated

2	🔶 AI Image Generator	User types a prompt describing a sustainable city scene	Imaginative	System generates an image and allows download	Inspired & satisfied
3	📓 Problem & Solution Finder (RAG-based)	User submits a local issue (e.g., water wastage)	Concerned	System provides document-based solutions with summaries	Reassured & empowered
4	📊 City Health Dashboard	User opens dashboard to view metrics like air quality	Alerted	Interacts with charts and insights for awareness	Engaged & Aware
5	🏙 City Comparison Tool	User selects two cities to compare key sustainability data	Curious	Dashboard presents side-by-side comparison with export option	Analytical & Decisive

🔍 Insight: Clear, timely, and readable feedback at each step helps users stay engaged and make informed decisions.

3.2 ⍰ Solution Requirements

- ✅**Functional Requirements:**

  - Accepts user input for waste items via text box.
  - Returns disposal/recycling advice and sustainability tips.
  - Accepts user prompts describing a sustainable city scene.
  - Sends prompt to an image generation model (e.g., Stable Diffusion).
  - Displays generated image and allows download
  - Stores past prompts and images for session reference
  - Accepts user input describing an urban problem.
  - Collects and visualizes metrics (e.g., air quality, energy use, traffic).
  - Displays interactive charts and tables using Altair
  - Allows users to select two cities.

- Fetches real-time/static sustainability data for both.

- **Non-Functional Requirements:**
- **Performance**: Each module is going to return the output within 10 seconds except LLM Interaction
- **Responsiveness**: UI is adaptable for desktop
- **Security**: Handle user input safely and avoid injection or unauthorized access.
- **Usability**: Easy-to-navigate interface with input examples.
- **Reliability**: Modules must respond without crashes, even with incomplete or invalid inputs.
- **Scalability**: Should allow more cities, documents, and user inputs over time.
- **Compatibility**: Should work seamlessly in browsers (preferably via Stream lit)
- **Portability**: Compatible with VS code

User Stories:

♻ Recycle Management Advisor

1. **As a citizen**, I want to enter a household waste item, so that I can know how to dispose or recycle it correctly.
2. **As a user**, I want location-based tips on nearby recycling centres, so that I can contribute to sustainability easily.

◈ AI Image Generator

3. **As a city planner**, I want to generate visual concepts of sustainable environments, so that I can share them with stakeholders.
4. **As a student**, I want to visualize my eco-friendly city ideas using AI, so that I can include them in my school project.

📕 Problem & Solution Finder (RAG-based) – User Stories

5. **As a concerned resident**, I want to describe a city problem and get AI-generated solutions with policy references, so that I can advocate for change.
6. **As a government employee**, I want to upload a policy document and get a summarized view, so that I can quickly grasp its main points.

📊 City Health Dashboard – User Stories

7. **As a citizen**, I want to monitor my city's air, traffic, and waste metrics, so that I can stay aware of environmental health.
8. **As a researcher**, I want access to time-based city data visualizations, so that I can analyse trends and patterns.

🏙 City Comparison Tool – User Stories

9. **As a policy analyst**, I want to compare two cities based on sustainability indicators, so that I can evaluate which city performs better.

10**. As a journalist**, I want to export city comparison reports, so that can include visuals in my environmental articles.

3.3 Data Flow Diagram

Simplified Flow:

📊 Core System Flow

1. Functionality Selection Gateway

User → Main Dashboard → Functionality Selector → Selected Module

Decision Point: User selects one of five available functionalities:

- ♻ Recycle Management Advisor
- 💠 AI Image Generator
- 📖 Problem & Solution Finder (RAG)
- 📊 City Health Dashboard
- 🏙 City Comparison Tool

🔄 Individual Module Data Flows

♻️ Recycle Management Advisor

Input: Waste Type/Location

↓

Waste Classification Engine

↓

Local Recycling Database Query

↓

Guidelines Generator

↓

Output: Disposal Instructions & Eco-Tips

🎨 AI Image Generator

Input: City Description (Imaginary)

↓

Natural Language Processing

↓

AI Image Generation Model

↓

Image Post-Processing & Optimization

↓

Output: Sustainable City Visualization

📘Problem & Solution Finder (RAG-based)

Input: Urban Problem Query

↓

Query Processing & Intent Recognition

↓

Document Retrieval System

↓

Knowledge Base/Vector Database Search

↓

RAG Processing Engine

↓

Contextual Solution Generation

↓

Output: AI-Generated Solutions with Source

📊 City Health Dashboard

Input: City Selection

↓

Multi-Source Data Aggregation

↓

[Environmental APIs | Health Metrics DB | Real-time Sensors]

↓

Data Processing & Analytics Engine

↓

Visualization Generation

↓

Output: Interactive Health Dashboard

🏙 City Comparison Tool

Input: Two Cities Selection

↓

Parallel Data Retrieval for Both Cities

↓

[Air Quality APIs | Traffic Data | Waste Management | Health Metrics]

↓

Comparative Analysis Engine

↓

Side-by-side Metrics Processing

↓

Output: Comprehensive Sustainability Comparison Report

Flow Description:

Users authenticate and access a dashboard with five key modules: ♻ Recycle Advisor, 🔷 AI Image Generator, 📄 RAG-based Problem Solver, 📊 Health Dashboard, and 🏙 City Comparison Tool.
Each module processes inputs using specialized engines—AI models, document retrievers, or data aggregators—while pulling real-time data from APIs, sensors, and city databases.
The system outputs personalized results like recycling tips, generated images, solution summaries, dashboards, or comparative city reports.

3.4 ▯ Technology Stack

- Frontend: stream lit UI
- Backend: Python
- AI Models: IBM Granite, Stable Diffusion
- Hosting: Vs Code (Model Execution)
- Visualization: Altair

4.Project Design

4.1 Problem Solution Fit

Date	20 June 2025
Team ID	LTVIP2025TMID29572
Project Name	Sustainable smart city Assistant
Maximum Marks	2 Marks

In our Sustainable Smart City project, problem–solution fit means identifying real, everyday challenges faced by citizens, planners, and researchers — such as confusion about recycling, lack of sustainability data, and limited tools for future planning — and developing an AI-powered assistant hub that directly solves those problems through smart, personalized, and accessible features.

Purpose:

❑Solve complex problems in a way that fits the state of your customers: Our project addresses challenges such as waste mismanagement, lack of sustainability awareness, and difficulty in comparing urban indicators — all of which are common in growing cities and underserved villages.

❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channel of behaviour By using intuitive tools like Stream lit dashboards and chat-based Fast API services, we make sustainability data and advice accessible in the formats users are already familiar with.

❑ Sharpen your communication and marketing strategy with the right triggers and messaging:
Each module (e.g., waste recycling assistant, city comparison, dream city visualizer) is based on common user problems and is designed to deliver helpful information in a personalized and timely manner.

❑ Increase touch-points with your company by finding the right problem-behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems: Our assistant helps users solve day-to-day issues like waste confusion or lack of data, building trust through repeated interactions and reliability.

❑ Understand the existing situation in order to improve it for your target group: We gather, process, and visualize sustainability indicators for multiple regions, helping users identify gaps and opportunities for improvement in a data-driven way.

1. Customer Segment(s):   Citizens in growing urban areas, village planners, researchers, and environmentally conscious individuals.

2. Jobs-to-be-Done / Problems:

- Dispose of waste correctly
- Make eco-conscious daily decisions
- Compare sustainability across regions
- Visualize a future smart city
- Access real-time sustainability data

3. Triggers:

- Seeing others recycle or follow eco-habits
- Local news about pollution or government green policies
- Frustration from confusing waste bins or lack of information

4.Emotions (Before/After):

Before: Confused, uninformed, discouraged

After: Empowered, confident, engaged, proud of contributing to sustainability

5. Available Solutions:

- Government awareness websites
- Recycle info printed on bins
- Environmental blogs/articles
- Urban data portals (not integrated)

6. Customer Constraints:

- Limited time and awareness
- No access to centralized data
- Low technical knowledge
- Language or digital barriers

7. Behaviour:

- Search online for eco-practices
- Ask neighbours or community
- Rely on trial-and-error for recycling
- Rarely engage with smart city data

8.Channels of Behaviour: -

Online: Google, YouTube, Instagram, WhatsApp, civic websites
Offline: Word-of-mouth, flyers, local community meetings

9. Problem Root Cause:

- Scattered or inaccessible data
- No integrated smart city sustainability tool
- Citizens want to act but lack clear, accessible guidance

10. Our Solution: An AI-powered Smart City Assistant Hub that includes:

- A sustainability dashboard
- Recycling/upcycling suggestions
- Village/city comparison tool

- Dream city visualizer
- Smart City RAG Solver
- Empowers users to act sustainably through a unified, easy-to-use digital platform.

4.2 Proposed Solution

Date	20 June 2025
Team ID	LTVIP2025TMID29572
Project Name	Sustainable Smart City
Maximum Marks	2 Marks

Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Citizens in growing urban and rural areas face challenges related to sustainability, including inefficient waste management, rising pollution, and lack of development insights. There is limited access to real-time environmental data and clear comparisons between cities and villages, making it difficult for individuals and planners to make informed decisions for sustainable living.
2.	Idea / Solution description	1. Recycling Assistant – Uses a generative model to provide eco-friendly disposal suggestions, recycling techniques, and upcycling DIY ideas based on user inputs. 2. Village Comparator –

		Allows comparison of key sustainability indicators between villages, supporting Rural development and identifying gaps in resources or policy. 3. Smart City RAG Solver – A smart city query-solving feature that uses Retrieval-Augmented-Generation (RAG) to answer sustainability-related citizen questions using relevant knowledge bases. 4. AI Dashboard – A Stream lit-based visual interface that integrates all modules, provides comparative analytics, and enables user interaction with visual data and models. 5. Dream City Builder – A feature that allows users to simulate and design their own ideal sustainable city by selecting the best parameters from different real cities or villages, educating them on what makes a truly sustainable environment
3.	Novelty / Uniqueness	This solution uniquely blends advanced AI (LLMs, RAG) with user interaction, covering both urban and rural areas. Unlike traditional dashboards or comparison tools, it

		includes a Dream City Builder, allowing users to experiment and learn by virtually designing a sustainable city. It also bridges the gap between awareness and action through personalized recycling advice and smart query resolution.
4.	Social Impact / Customer Satisfaction	The system fosters a culture of sustainability by helping individuals understand and take part in solving environmental challenges. Citizens learn how to recycle, compare their village or city, ask questions, and even design a better city — all in one place. Planners, students, and policymakers gain powerful data-driven insights. This improves engagement, awareness, and overall satisfaction.
5.	Business Model (Revenue Model)	Revenue can be generated through: • Subscription plans for smart city departments, educational institutions, And NGOs. • Freemium access for citizens with premium tools (e.g., Dream City export, detailed analytics). • White-labelling to sustainability-focused startups and government agencies. • Sponsored collaborations with environmental brands and green campaigns.

6.	Scalability of the Solution	All modules are modular and cloud-deployable, allowing seamless expansion across new cities and villages. Language support, region-specific datasets, and customizable dashboards make the solution adaptable for different demographics. It can be used nationally or globally, across education, governance, and community platforms.

4.3 Solution Architecture

Date	20 June 2025
Team ID	LTVIP2025TMID29572
Project Name	Sustainble Smart City
Maximum Marks	4 Marks

Solution Architecture:

The Sustainable Smart City system is designed as a modular, scalable AI-powered platform that enables citizens and authorities to engage with sustainability initiatives through an intuitive user interface and intelligent services. At a high level, the system is organized into five interconnected layers: the User Dashboard, API Gateway, Microservices Layer, AI/ML Layer, and the Data Layer. Each layer plays a critical role in delivering responsive, insightful, and real-time sustainability solutions.

Frontend Layer:

The system's user-facing interface is built using modern frontend technologies like React or Vue.js, ensuring a responsive and user-friendly experience across devices. It includes a dynamic dashboard that enables users to interact with different features such as the Recycle DIY tool, Village Comparator, Problem-Solution Finder, and Dream City Generator. JWT-based authentication ensures secure login and user management. For live updates and dynamic content, WebSocket connections are integrated. Visual elements like sustainability graphs and comparison metrics are displayed using visualization libraries such as Chart.js or D3.js.

API Gateway and Load Balancer:

To manage traffic efficiently and route requests, the platform uses an API Gateway such as Kong or AWS API Gateway. This layer handles rate limiting, authentication, and logging, making the APIs robust and secure. A load balancer like NGINX distributes incoming requests to different services, maintaining availability and preventing overload on any single service.

Microservices Architecture:

The platform is powered by a suite of independent microservices, each handling a specific feature of the system.

- The Recycle DIY Service processes user inputs describing waste materials. It utilizes IBM Granite models for material classification and recommends upcycled DIY solutions. Environmental impacts, such as carbon footprint reduction, are calculated based on a database of material properties and templates.
- The Village Comparator Service enables users to compare sustainability metrics across two regions. It gathers data from external APIs and internal sources, then leverages a Retrieval-Augmented Generation (RAG) pipeline backed by a vector database. The Google Flan model is used to generate insights and recommendations for sustainable development.
- The Problem-Solution Service functions as an AI assistant for sustainability-related challenges. It uses natural language understanding to process user queries and performs a semantic search using RAG techniques to retrieve the most relevant solutions from a curated knowledge base.
- The Dream City Generator allows users to envision their ideal sustainable city. By extracting parameters from user input, the system uses generative AI models like DALL·E or Stable Diffusion to create city images. It also calculates city health metrics (e.g., pollution levels, renewable energy usage) and offers recommendations for improvement.

AI/ML Layer:

This layer supports all AI-driven functionalities. Models such as IBM Granite and Google Flan-T5 are hosted either on cloud platforms like IBM Watson or containerized within the infrastructure. An image generation model is integrated for the Dream City feature. For managing multiple model versions and tracking performance, ML flow is used. The RAG engine uses vector databases like Pinecone to store and search through embeddings generated by models like Sentence-BERT. Retrieval is based on cosine similarity, ensuring accurate and context-aware results.

Data Layer:

Structured data such as user profiles, city metrics, and templates are stored in PostgreSQL, while unstructured content including AI-generated responses is managed in MongoDB. Redis serves as a caching layer to handle session data and frequently accessed information. The system draws data from a combination of sources including government APIs, environmental datasets, and user-contributed content. These sources keep the platform up-to-date with real-world sustainability metrics.

Workflow and Feature Flow:

Each feature has a clearly defined workflow. For example, in the Recycle DIY flow, a user submits a material description, which is classified by AI. Templates and impact data are fetched from the database, and a personalized suggestion is returned. The Village Comparator involves comparing two regions by retrieving data, running a RAG process, and generating insights through AI analysis. The Problem-Solution flow starts with user queries, which are understood and matched against a semantic knowledge base, returning actionable solutions. For the Dream City feature, user preferences are translated into parameters, visualized using AI-generated imagery, and analysed to produce city health scores and recommendations.

Technology Stack:

On the backend, services are built with Node.js (Express) or Python (Fast API) and deployed using Docker containers orchestrated by Kubernetes. Message queues like Kafka manage asynchronous processing, while Prometheus and Grafana provide real-time monitoring. AI model serving is handled by TensorFlow Serving or ML flow, with GPU support from NVIDIA CUDA. Large-scale data is processed in batches using Apache Spark, and Kubeflow automates the ML pipeline.

Security and Compliance:

Security is enforced through OAuth 2.0 authentication, RBAC-based authorization, and strong data encryption protocols such as TLS 1.3 and AES-256. The system includes protections against common web threats via input validation and vulnerability scanning. GDPR compliance is ensured by implementing user consent flows, anonymizing data, and logging access for audit purposes. Regular backups and penetration testing protect against data loss and breaches.

Data Flow Architecture:

Data flows through the system in three main pipelines. In the real-time pipeline, external APIs feed data into Kafka, processed in streams, and cached for instant dashboard updates. The AI processing pipeline takes user inputs through preprocessing, inference, post-processing, and caching for efficient response delivery. In the batch processing pipeline, large datasets undergo ETL operations and are stored in a data warehouse, with insights visualized in the analytics dashboard.

Deployment Strategy:

The system is deployed on cloud platforms like AWS, Azure, or GCP. It uses Kubernetes clusters for elastic compute, with cloud-based storage for data and media. CDNs like CloudFront or Azure CDN deliver static content efficiently. The CI/CD pipeline automates code integration, testing, containerization, and deployment, ensuring rapid iteration and continuous delivery.

Scalability and Performance:

The architecture supports horizontal scaling by independently scaling services and sharding databases. Caching at multiple levels (Redis, CDN, browser) improves response times, and load balancing distributes user requests efficiently. Optimization strategies include indexing database queries, GPU acceleration for models, and minimizing API latency below 200ms.

Monitoring, Analytics, and Alerts:

Operational health is tracked using metrics dashboards that monitor system performance, AI model accuracy, and user engagement. Alerts are triggered on performance drops, resource overuse, or AI model drift. Frontend performance is also monitored to ensure a seamless user experience.

Solution Architecture Diagram:

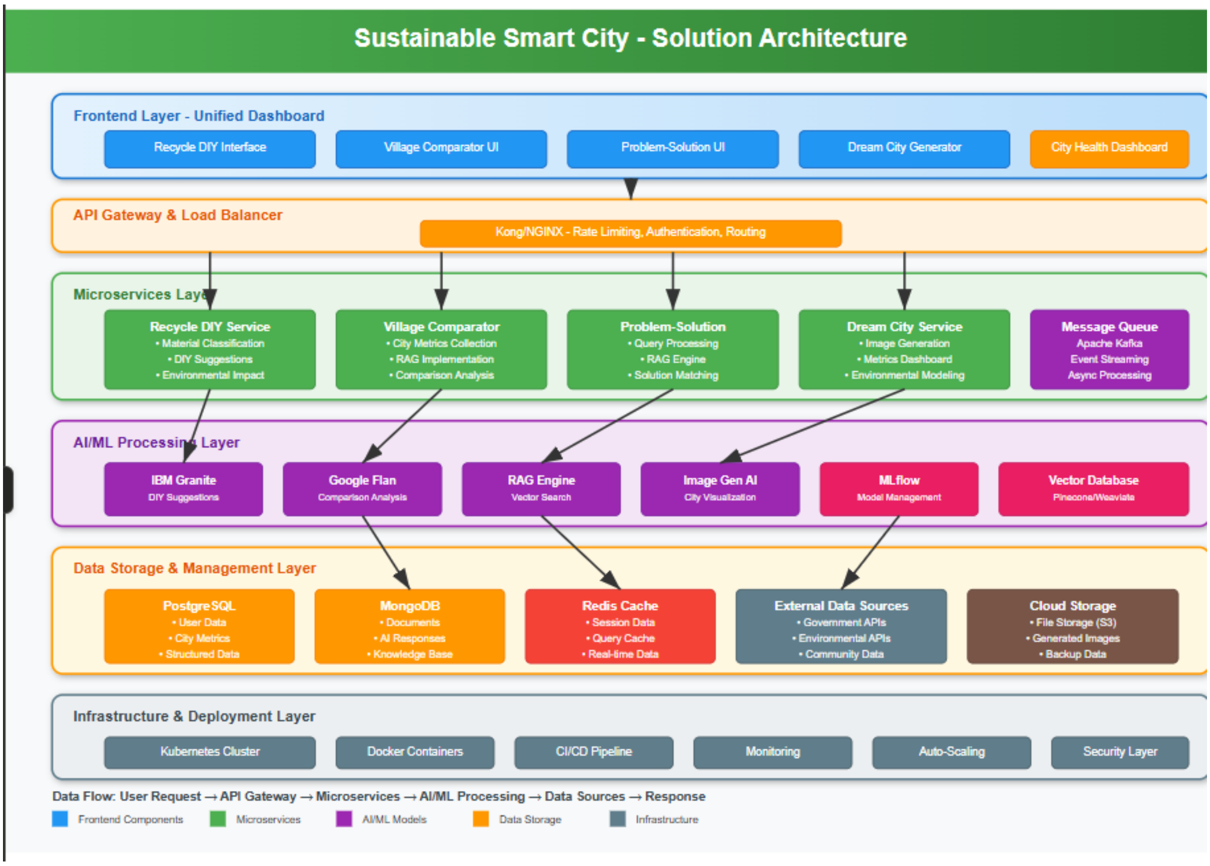


Figure 1: Architecture and data flow of sustainable

5.PROJECT PLANNING AND SCHEDULING

5.1 Project Planning Logic

📅 Project Start Date: 20th June 2025

🛠 Agile Framework: 2 Sprints (2 working days each)

- **Sprint 1:** June 20–June 21, 2025
- **Sprint 2:** June 22–July 23, 2025

🚀 Epic 1: Data Preparation and Analysis

Sprint 1: June 20–June 21, 2025

Story	Story Points	Difficulty
Collect sustainability datasets	2	Easy
Load datasets into environment	1	Very Easy
Handle missing values	3	Moderate
Encode categorical values	2	Easy

✅ **Total Story Points Sprint 1:** 8

▯ Epic 2: Model Development and Deployment

Sprint 2: June 22–July 23, 2025

Story	Story Points	Difficulty
Build ML model (e.g., RAG, Recycle Manager)	5	Difficult
Test model with scenarios	3	Moderate
Design working HTML UI pages	3	Moderate

Story	Story Points	Difficulty
Deploy app with FAST API	5	Difficult

✅ **Total Story Points Sprint 2:** 16

📊 Velocity Calculation

Metric	Value
Sprint 1 Points	8
Sprint 2 Points	16
Total Points	**24**
Number of Sprints	**2**
Velocity	**12 story points/sprint**

📌 Summary

- 🗂 **Epics:**
  - Epic 1: Data Pipeline Setup
  - Epic 2: Model Deployment & UI Integration
- ⏳ **Duration:** 2 Weeks (10 working days)
- ⚙️ **Velocity:** 12 points/sprint (Team capacity baseline)

- 📈 **Next Planning:** Use this velocity to estimate remaining features (like Image Gen, Problem and Solutions Using RAG, Dashboards) for further sprints.

Sustainable Smart City Assistant Project: Agile Report

? Sprint Duration

• Sprint Duration: 2 Days Each

• Sprint Timeline: 20 June 2025 to 28 June 2025

? Product Backlog & Sprint Schedule

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**
Sprint-1	Recycle Management	USN-1	As a user, I can enter a recyclable item and receive advice on how to dispose or reuse it properly	3	High
Sprint-1	Recycle Management	USN-2	As a user, I can get eco-tips for my neighborhood based on my pin code	2	Medium
Sprint-2	Image Generation	USN-4	As a user, I can input a text prompt to generate a sustainable city image using an AI model	3	High

Sprint-2	Image Generation	USN-5	As a user, I can download and share the generated image	2	Medium
Sprint-3	Problem & Solution (RAG)	USN-6	As a user, I can enter a sustainability issue and get AI-generated solutions with relevant documents	4	High
Sprint-3	Problem & Solution (RAG)	USN-7	As a user, I can get real-time summaries of city issues and resolutions	2	Medium
Sprint-4	City Comparison & Health Dashboard	USN-8	As a user, I can compare two cities based on air quality, traffic, and waste data	4	High
Sprint-4	City Comparison & Health Dashboard	USN-9	As a user, I can access a real-time city health dashboard visualized with charts	3	High

? Sprint Tracker

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Completed	Sprint Release Date
Sprint-1	8	2 Days	20 June 2025	21 June 2025	8	21 June 2025
Sprint-2	5	2 Days	22 June 2025	23 June 2025	5	23 June 2025
Sprint-3	6	2 Days	24 June 2025	25 June 2025	6	25 June 2025
Sprint-4	7	2 Days	26 June 2025	27 June 2025	7	27 June 2025

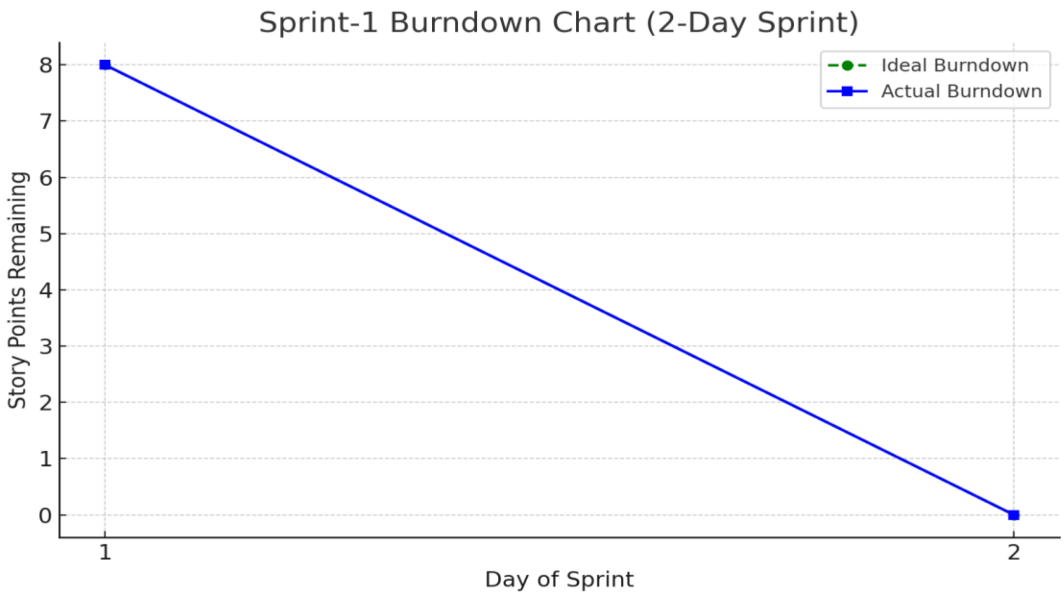
? Velocity

- Sprint Duration: 2 Days Each
- Sprint-1 Velocity: 4 Story Points per Day

? Sprint-1 Burndown Chart

Ideal Burndown vs Actual Burndown (2-Day Sprint):

Day	Ideal Story Points Remaining	Actual Story Points Remaining
1	8	8
2	0	0



? Summary

- The team successfully completed all planned tasks for Sprint-1 within 2 days.
- Future sprints are scheduled every 2 days, ending by 28 June 2025.
- All five modules of the assistant will be completed across 4 sprints.

6.FUNCTIONAL AND PERFORMANCE TESTING

6.1 Functional & Performance Testing Template

Model Performance Test

Date	26 June 2025
Team ID	LTVIP2025TMID29572
Project Name	Sustainability Smart City
Maximum Marks	

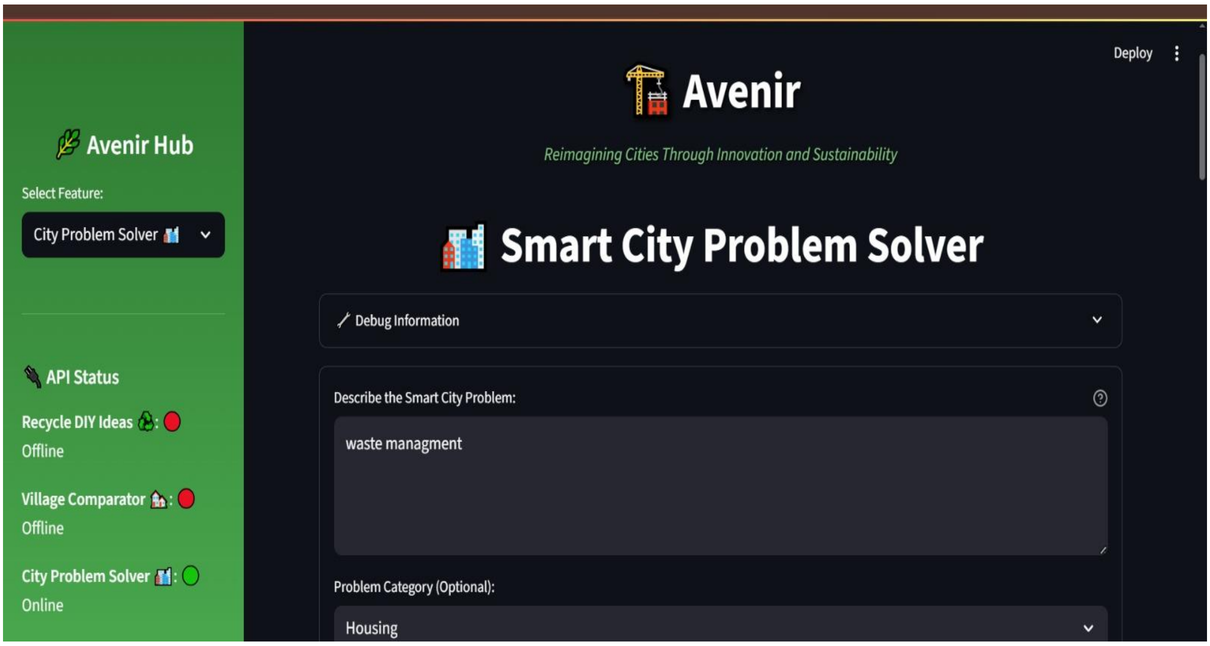
Test Scenarios & Results

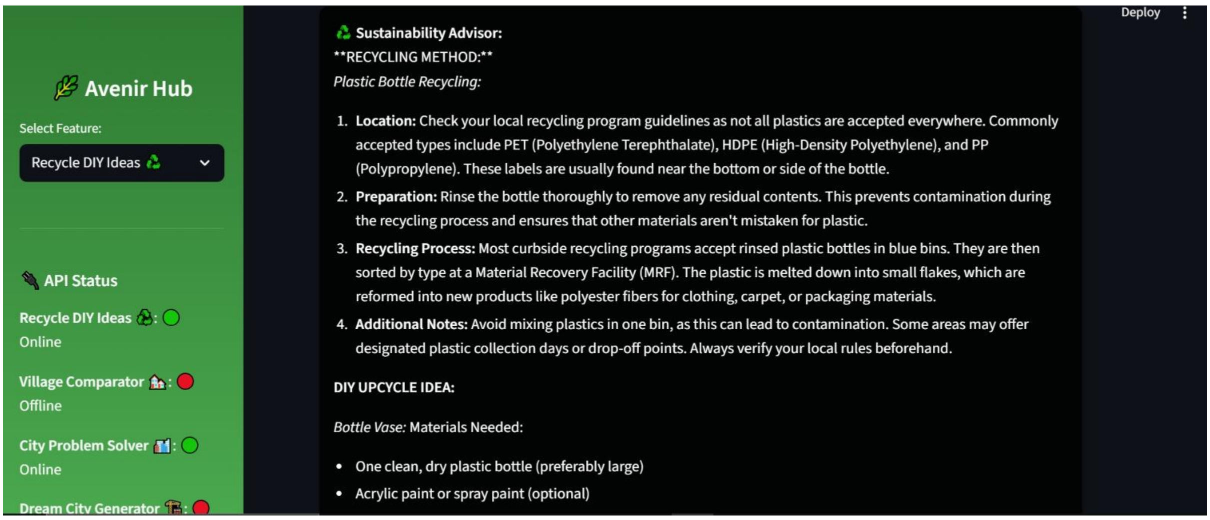
Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	Valid and invalid text handled correctly	Pass
FT-02	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	All number inputs validated properly	Pass
FT-03	Content Generation (e.g., recycle idea, Image Generation	Provide complete inputs and click "Generate"	Correct content is generated based on input	Content generated for recycle, image, city compare, solution	Pass

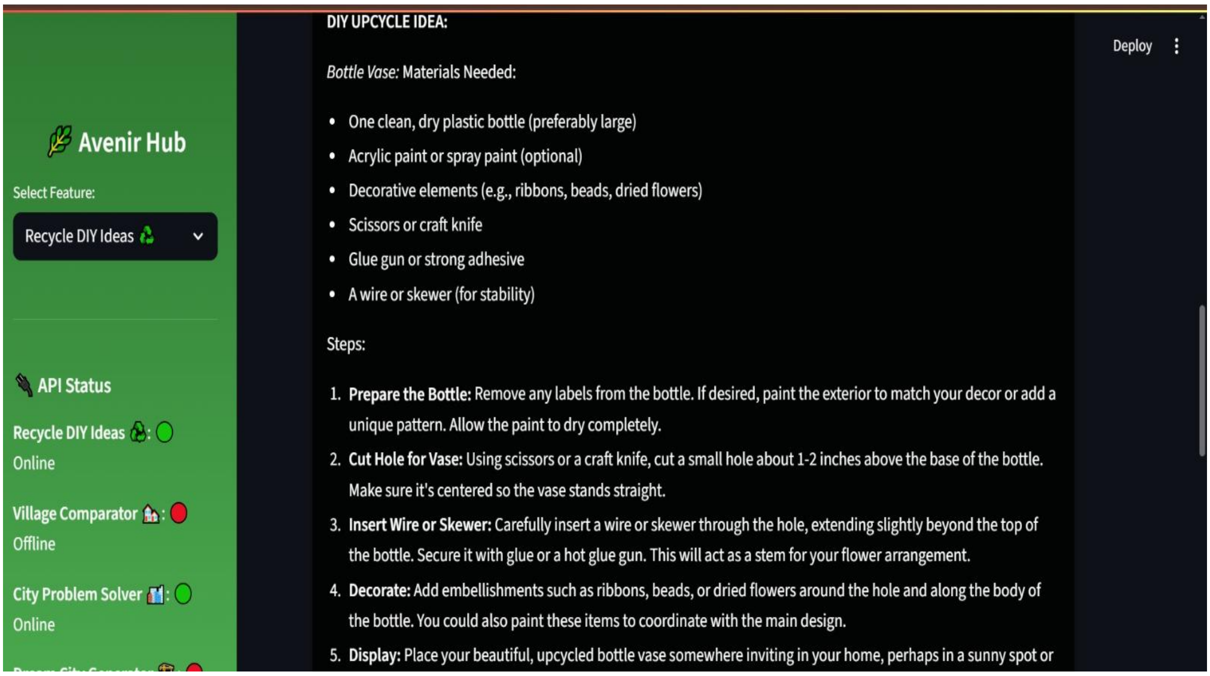
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	API connected and functioning for all features	Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	All functionalities respond under 3 seconds	Pass
PT-02	API Speed Test	Send multiple API calls at the same time	API should not slow down	API speed maintained under load	Pass
PT-03	File Upload Load Test (e.g., PDFs)	Upload multiple PDFs and check processing	Should work smoothly without crashing	Multiple file uploads tested successfully	Pass

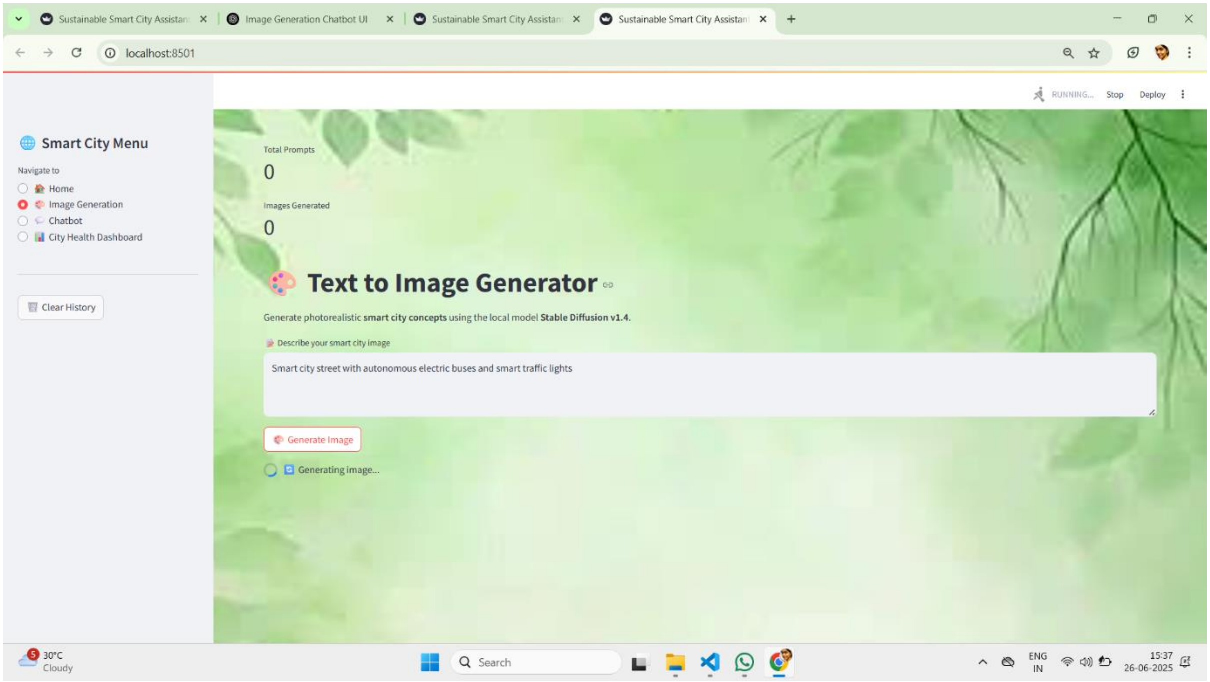
7.RESULTS

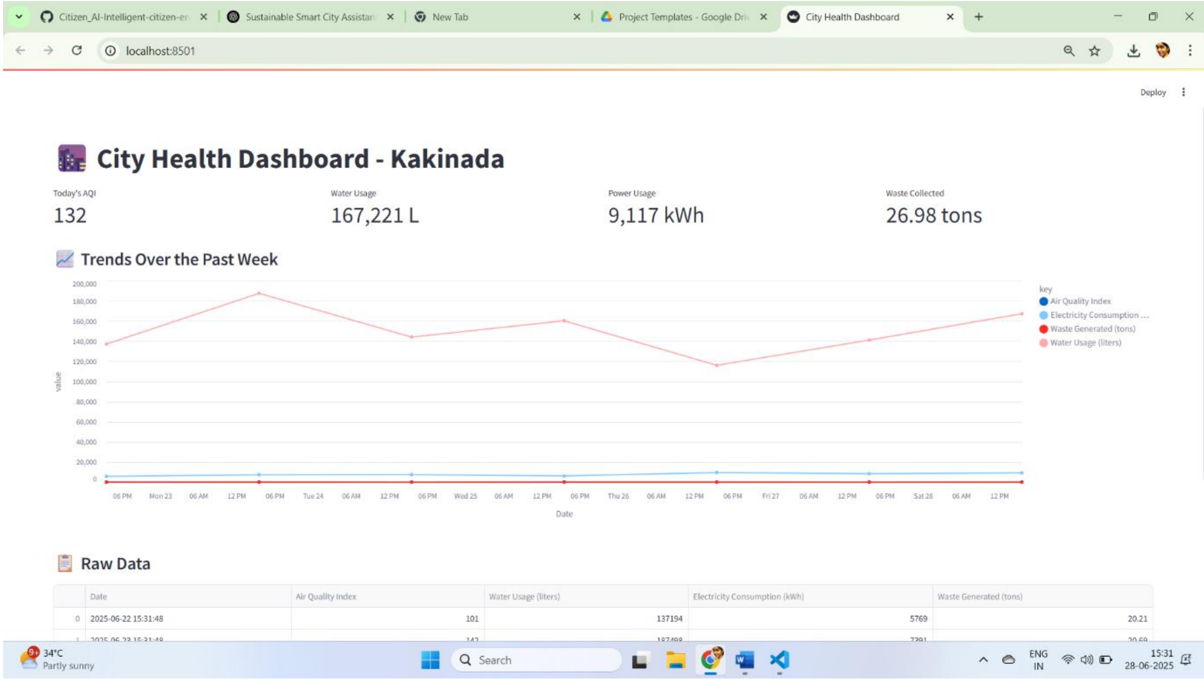
7.1 OUTPUT SCREENSHOTS

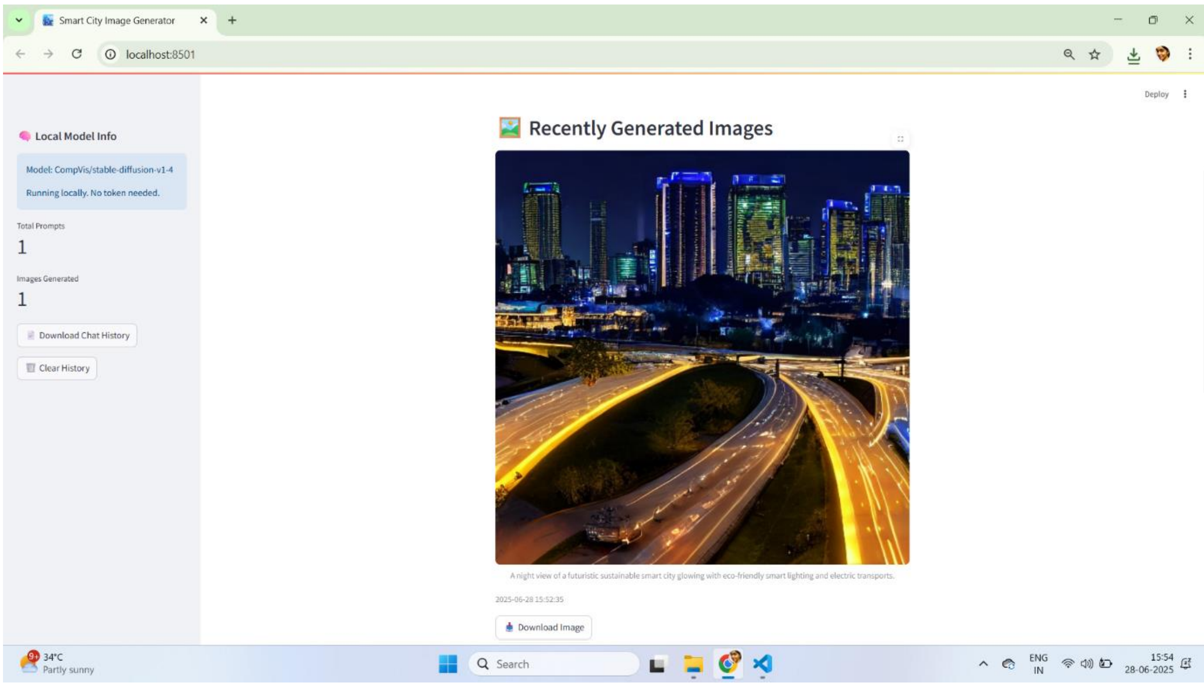





DIY UPCYCLE IDEA:
Deploy
Avenir Hub
Select Feature:
Recycle DIY Ideas
API Status
Recycle DIY Ideas: Online
Village Comparator: Offline
City Problem Solver: Online
Bottle Vase: Materials Needed:
One clean, dry plastic bottle (preferably large)
Acrylic paint or spray paint (optional)
Decorative elements (e.g., ribbons, beads, dried flowers)
Scissors or craft knife
Glue gun or strong adhesive
A wire or skewer (for stability)
Steps:
1. Prepare the Bottle: Remove any labels from the bottle. If desired, paint the exterior to match your decor or add a unique pattern. Allow the paint to dry completely.
2. Cut Hole for Vase: Using scissors or a craft knife, cut a small hole about 1-2 inches above the base of the bottle. Make sure it's centered so the vase stands straight.
3. Insert Wire or Skewer: Carefully insert a wire or skewer through the hole, extending slightly beyond the top of the bottle. Secure it with glue or a hot glue gun. This will act as a stem for your flower arrangement.
4. Decorate: Add embellishments such as ribbons, beads, or dried flowers around the hole and along the body of the bottle. You could also paint these items to coordinate with the main design.
5. Display: Place your beautiful, upcycled bottle vase somewhere inviting in your home, perhaps in a sunny spot or


localhost:8501
RUNNING... Stop Deploy
Smart City Menu
Navigate to
Home
Image Generation
Chatbot
City Health Dashboard
Clear History
Total Prompts
0
Images Generated
0
Text to Image Generator
Generate photorealistic smart city concepts using the local model Stable Diffusion v1.4.
Describe your smart city image
Smart city street with autonomous electric buses and smart traffic lights
Generate Image
Generating image...


Citizen_AI-Intelligent-citizen-en
Sustainable Smart City Assistant
New Tab
Project Templates - Google Dri
City Health Dashboard
localhost:8501
Deploy
City Health Dashboard - Kakinada
Today's AQI
132
Water Usage
167,221 L
Power Usage
9,117 kWh
Waste Collected
26.98 tons
Trends Over the Past Week
key
Air Quality Index
Electricity Consumption ...
Waste Generated (tons)
Water Usage (liters)
Date
Raw Data
Date
Air Quality Index
Water Usage (liters)
Electricity Consumption (kWh)
Waste Generated (tons)
0 2025-06-22 15:31:48 101 137194 5769 20.21
34°C
Partly sunny
Search
ENG
IN
15:31
28-06-2025


Smart City Image Generator
localhost:8501
Deploy
Local Model Info
Model: CompVis/stable-diffusion-v1-4
Running locally. No token needed.
Total Prompts
1
Images Generated
1
Download Chat History
Clear History
Recently Generated Images
A night view of a futuristic sustainable smart city glowing with eco-friendly smart lighting and electric transports.
2025-06-28 15:52:35
Download Image
34°C
Partly sunny
Search
ENG
IN
15:54
28-06-2025

8.ADVANTAGES AND DISADVANTAGES

✅ Advantages

1. **Modular Functionality**: Offers a wide range of services—from recycling guidance to comparative analytics—within a unified dashboard.
2. **AI-Powered Intelligence**: Uses generative AI, RAG, and machine learning for smart, contextual outputs.
3. **User-Friendly Interface**: Interactive and accessible via web platforms (Stream lit), suitable for both experts and citizens.
4. **Personalized Recommendations**: Outputs are tailored based on user input, location, or city data.

⚠ Disadvantages

- **Dependency on External APIs**: Real-time modules may fail if API sources are slow or unavailable.
- **Data Quality Issues**: Accuracy depends heavily on the quality and completeness of input datasets.
- **High Resource Requirement**: AI models (especially image generation or RAG) need GPUs and consume significant memory.

9.CONCLUSION

The **Sustainable Smart City Assistant** represents a forward-thinking solution to the complex challenges faced by modern urban environments. By integrating artificial intelligence with real-time data and user-friendly interfaces, this project offers a comprehensive platform that empowers both citizens and city planners to engage actively in sustainable development. The assistant bridges the information gap between people and policies, encouraging environmentally responsible behaviour through accessible tools and intelligent automation.

Each module in the system is purposefully designed to address a specific urban need—be it promoting eco-friendly waste disposal through the Recycle Advisor, enhancing environmental awareness via the City Health Dashboard, or generating creative ideas using the AI Image Generator. The inclusion of advanced features like the City Comparison Tool and RAG-based Problem Solver helps individuals evaluate city performance, understand policies, and suggest data-driven improvements.

Furthermore, the assistant's scalability allows it to be adapted and integrated into different city infrastructures across the globe. With ongoing advancements in AI, IoT, and cloud computing, the platform can evolve to incorporate new features like citizen feedback sentiment analysis, smart transportation planning, and energy consumption tracking.

In conclusion, the Sustainable Smart City Assistant is not just a technological innovation, but a vision for smarter governance and inclusive urban development. It supports environmental sustainability, improves quality of life, and lays the foundation for smarter, greener cities.

10.FUTURE SCOPE

10.1 Future Scope

1. **Voice-based Assistant**: Integrate voice input/output for accessibility and multilingual support.
2. **Mobile App Deployment**: Create a lightweight mobile version for field use by municipal staff or citizens.
3. **Advanced Analytics**: Incorporate predictive models for forecasting urban growth, pollution, or resource strain.
4. **Community Feedback Loop**: Add crowdsourcing or citizen reporting features for grassroots data.