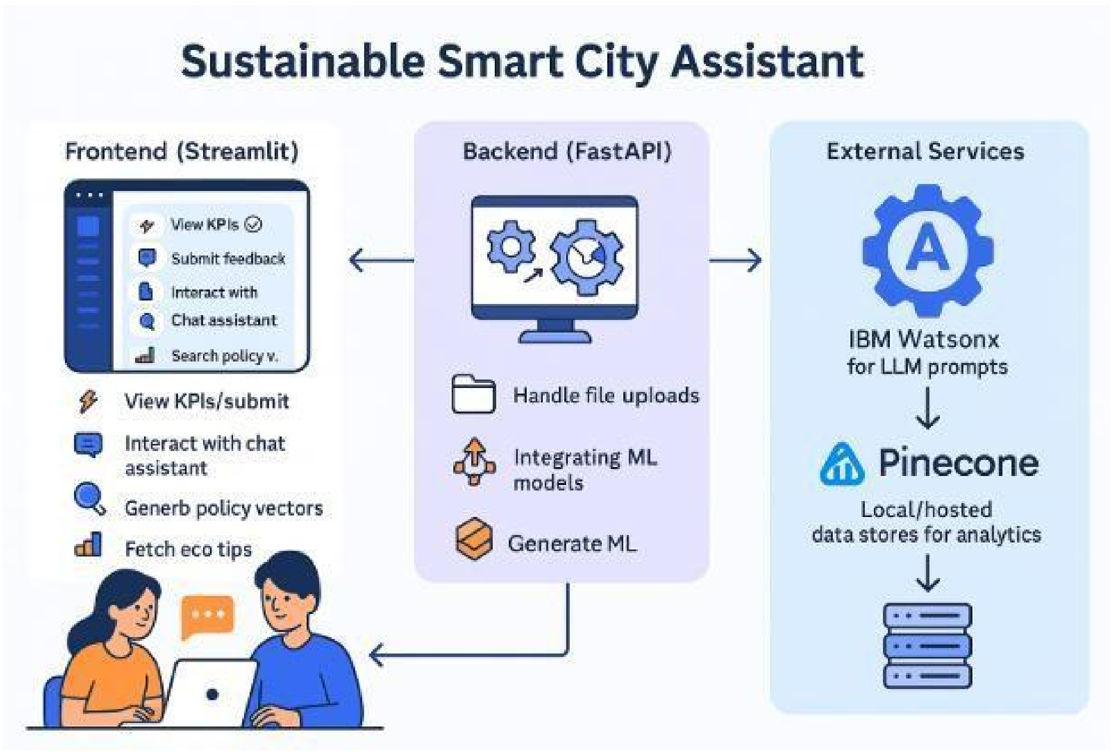
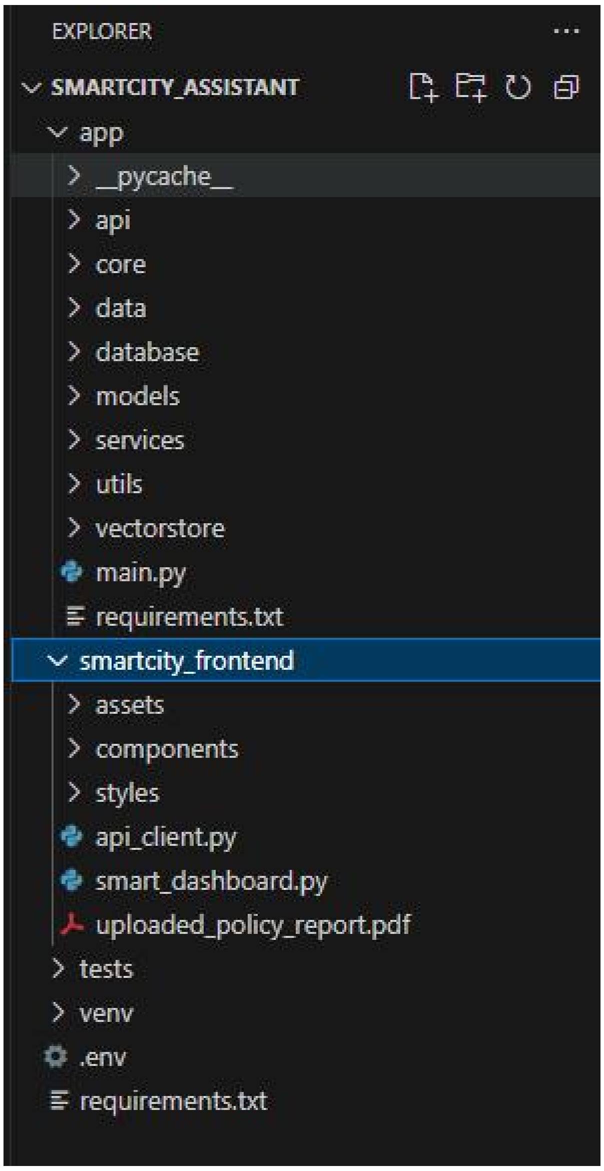
Technical Architecture



Development Flow

Phase 1-Project Initialization

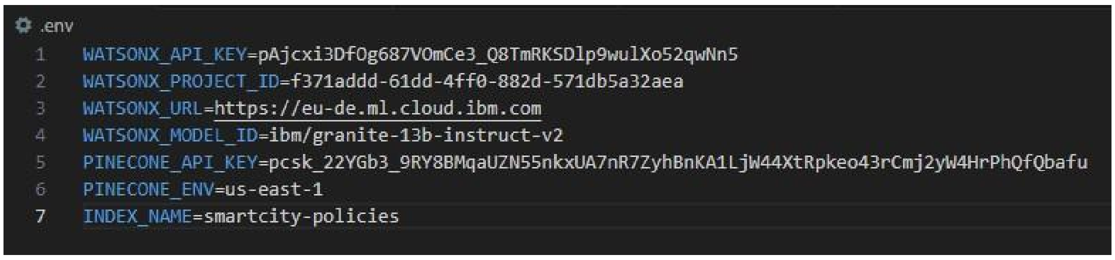
Modular Folder Structure Defined: Created separate folders for app/api, services, vectorstore,core, frontend/components, and utils for organized and scalable development.



```
EXPLORER
SMARTCITY_ASSISTANT
  app
    __pycache__
    api
    core
    data
    database
    models
    services
    utils
    vectorstore
    main.py
    requirements.txt
  smartcity_frontend
    assets
    components
    styles
    api_client.py
    smart_dashboard.py
    uploaded_policy_report.pdf
  tests
  venv
  .env
  requirements.txt
```

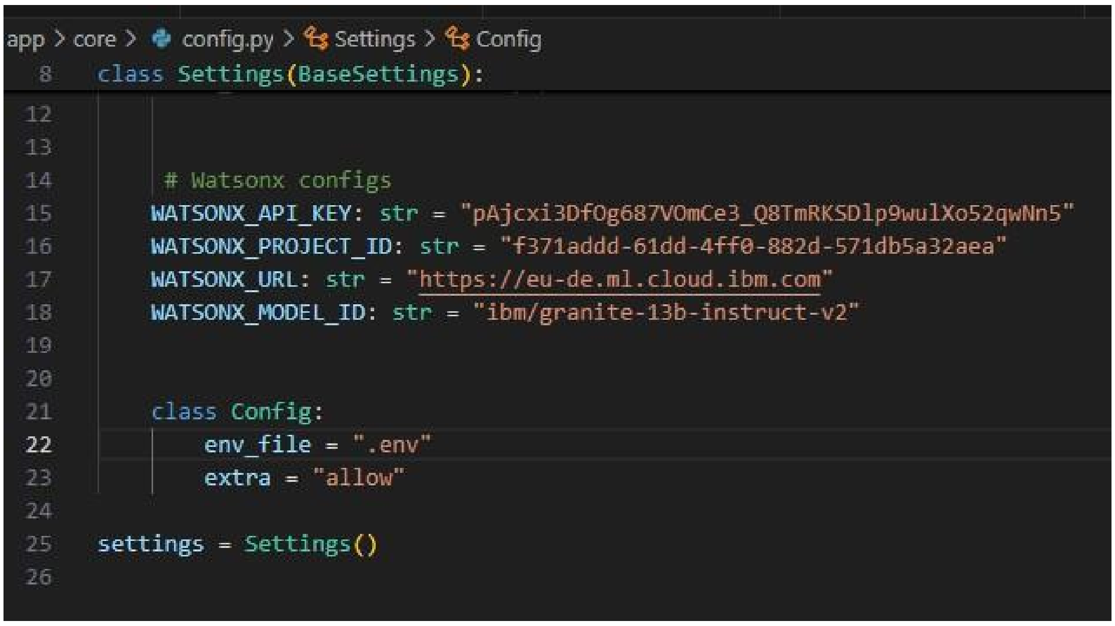
Environment Setup:
.env file created with keys for Pinecone and Watsonx. config.py loads environment variables securely using pydantic.

.env file



```
.env
WATSONX_API_KEY=pAjcxi3DfOg687VOmCe3_Q8TmRKSDlp9wulXo52qwNn5
WATSONX_PROJECT_ID=f371addd-61dd-4ff0-882d-571db5a32aea
WATSONX_URL=https://eu-de.ml.cloud.ibm.com
WATSONX_MODEL_ID=ibm/granite-13b-instruct-v2
PINECONE_API_KEY=pcsk_22YGb3_9RY8BMqaUZN55nkxUA7nR7ZyhBnKA1LjW44XtRpkeo43rCmj2yW4HrPhQfQbafu
PINECONE_ENV=us-east-1
INDEX_NAME=smartcity-policies
```

Config.py file



app > core > config.py > Settings > Config

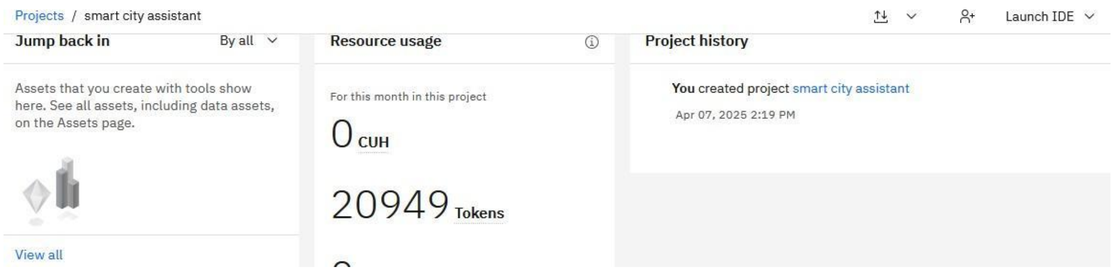
```python
class Settings(BaseSettings):


        # Watsonx configs
    WATSONX_API_KEY: str = "pAjcxi3Df0g687VOmCe3_Q8TmRKSDlp9wulXo52qwNn5"
    WATSONX_PROJECT_ID: str = "f371addd-61dd-4ff0-882d-571db5a32aea"
    WATSONX_URL: str = "https://eu-de.ml.cloud.ibm.com"
    WATSONX_MODEL_ID: str = "ibm/granite-13b-instruct-v2"


    class Config:
        env_file = ".env"
        extra = "allow"

settings = Settings()
```

Phase 2 – IBM Watsonx Integration



Assets that you create with tools show here. See all assets, including data assets, on the Assets page.

For this month in this project

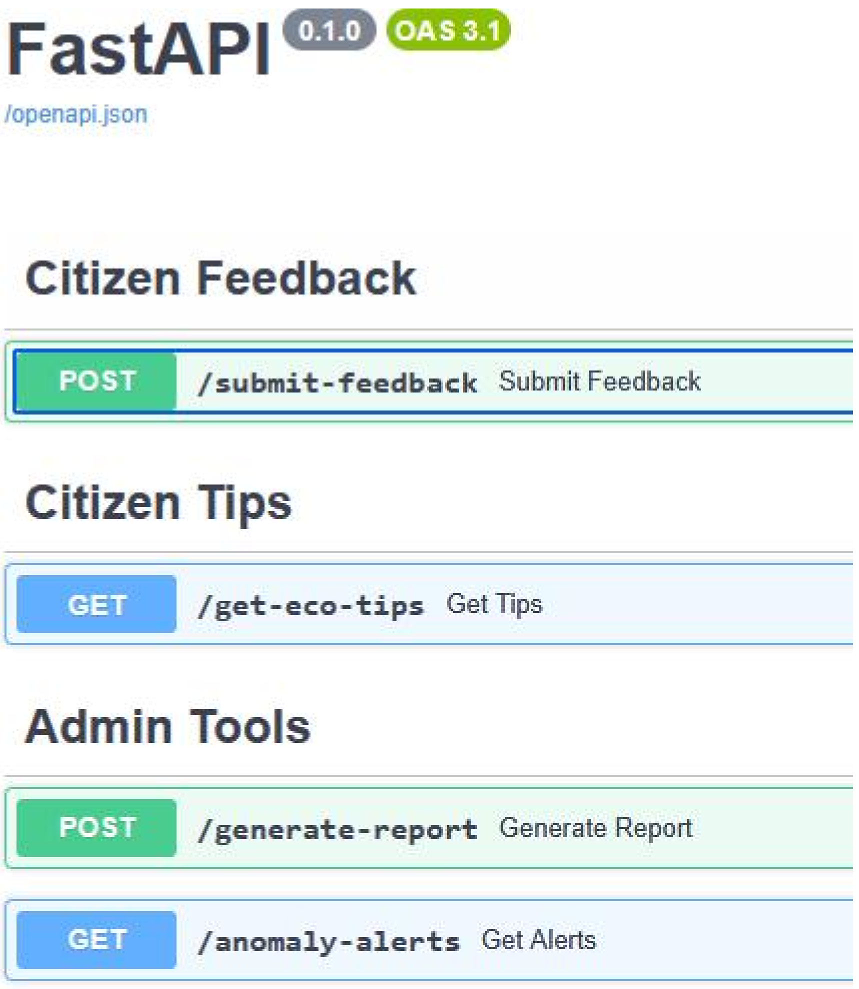
0 CUH

20949 Tokens

You created project smart city assistant

Apr 07, 2025 2:19 PM

Endpoint Testing:
Validated /chat, /policy/summarize, and /get-eco-tips FastAPI routes using Swagger UI


FastAPI 0.1.0 OAS 3.1

/openapi.json

Citizen Feedback

POST /submit-feedback Submit Feedback

Citizen Tips

GET /get-eco-tips Get Tips

Admin Tools

POST /generate-report Generate Report

GET /anomaly-alerts Get Alerts

Phase 3 – Backend API Routers API Routes Implemented:

Developed modular routers:

chat_router.py

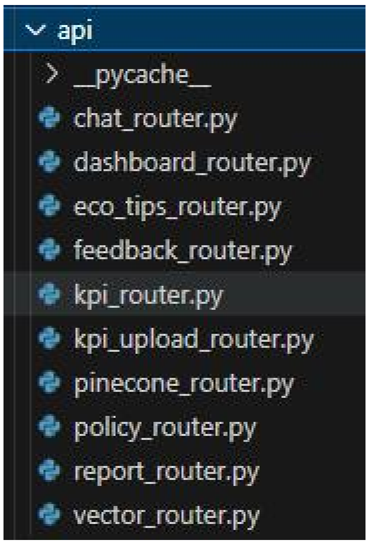
feedback_router.py

eco_tips_router.py

kpi_upload_router.py

anomaly_checker.py

vector_router.py, etc



```
api
  __pycache__
  chat_router.py
  dashboard_router.py
  eco_tips_router.py
  feedback_router.py
  kpi_router.py
  kpi_upload_router.py
  pinecone_router.py
  policy_router.py
  report_router.py
  vector_router.py
```

Testing & Validation:

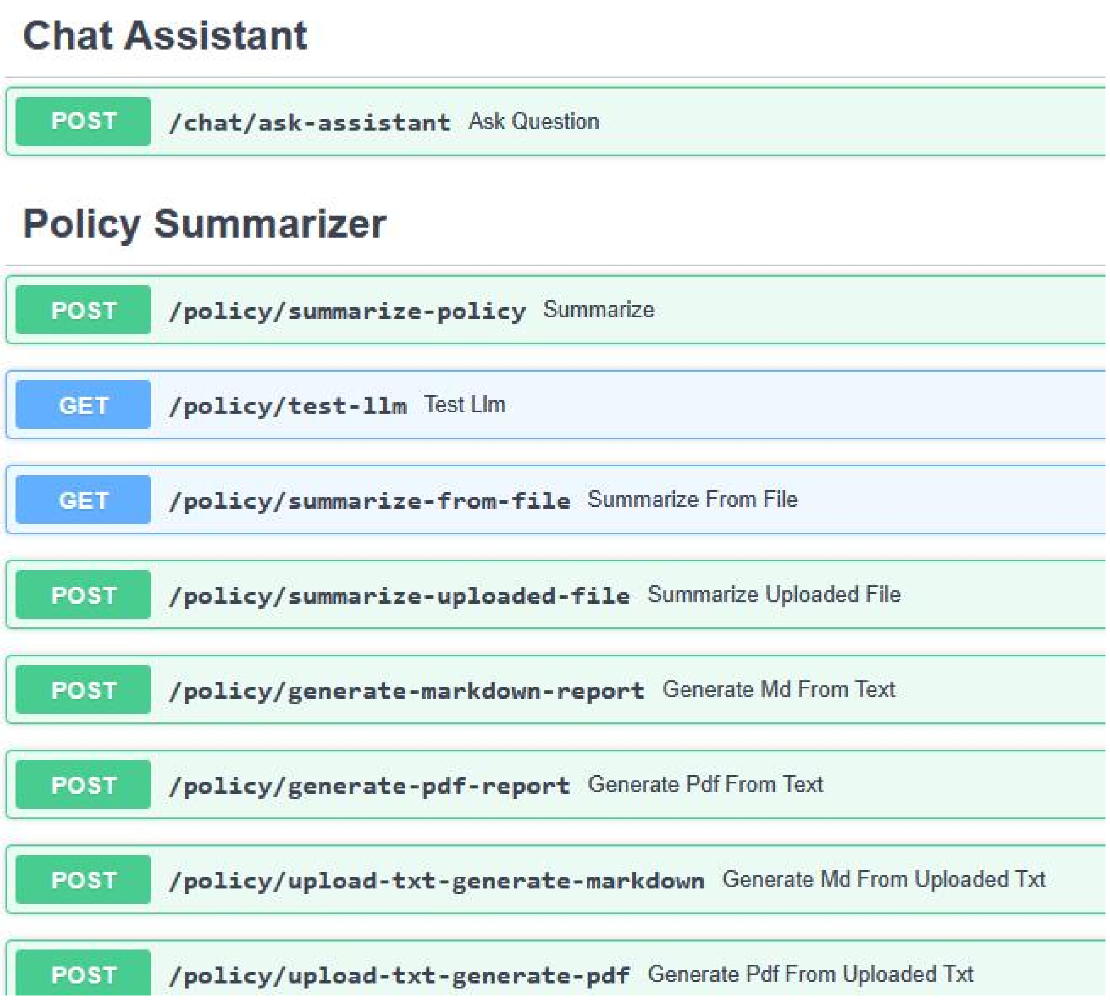
Each route tested for:

JSON payload correctness

File upload parsing

Error handling & logging

Swagger auto-documentation generation


Chat Assistant

POST /chat/ask-assistant Ask Question

Policy Summarizer

POST /policy/summarize-policy Summarize

GET /policy/test-llm Test Llm

GET /policy/summarize-from-file Summarize From File

POST /policy/summarize-uploaded-file Summarize Uploaded File

POST /policy/generate-markdown-report Generate Md From Text

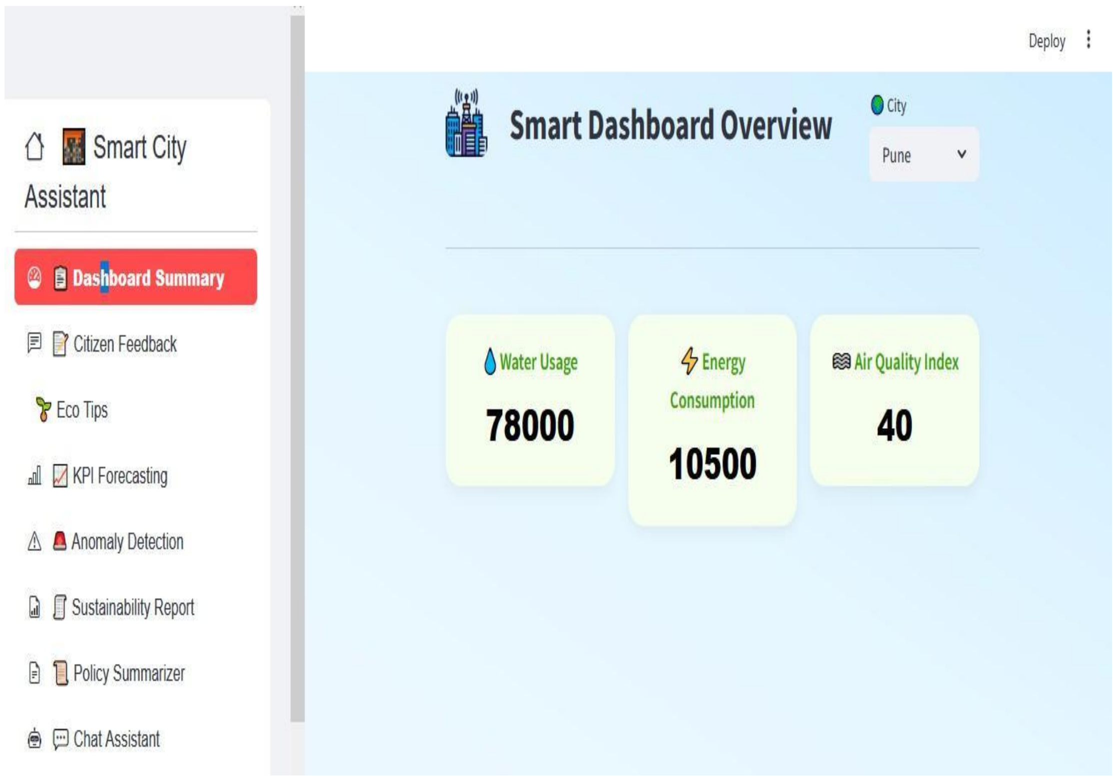
POST /policy/generate-pdf-report Generate Pdf From Text

POST /policy/upload-txt-generate-markdown Generate Md From Uploaded Txt

POST /policy/upload-txt-generate-pdf Generate Pdf From Uploaded Txt

Phase 4 – Frontend UI Design

Streamlit UI Structure Implemented:
Created central file smart_dashboard.py with conditional rendering for each module using sidebar navigation.



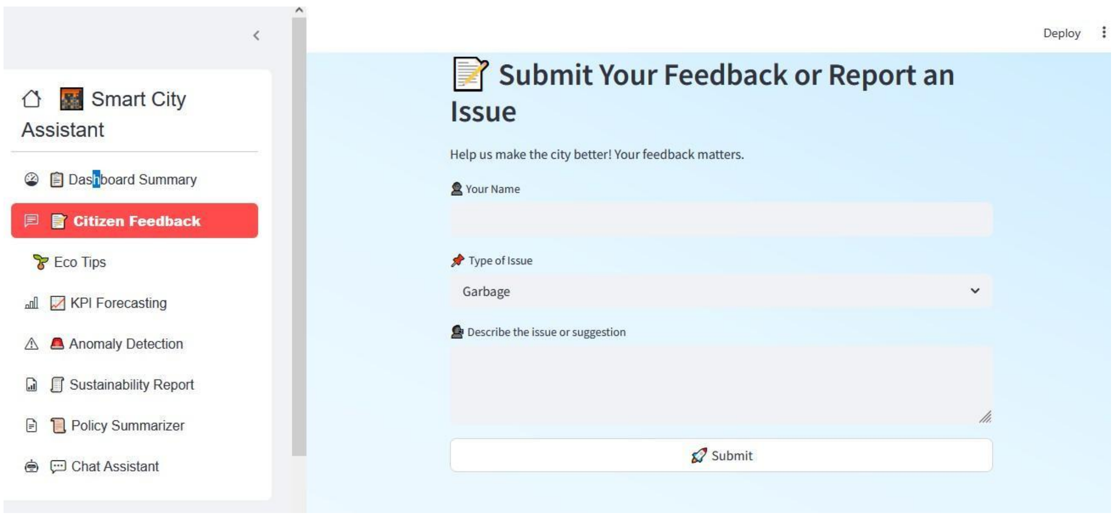
Smart Dashboard Overview

City: Pune

Water Usage	Energy Consumption	Air Quality Index
78000	10500	40

Component Development:

Developed reusable Streamlit components:
summary_card.py – Beautiful KPI cards chat_assistant.py – Text prompt and AI reply
feedback_form.py, eco_tips.py, report_generator.py, etc.



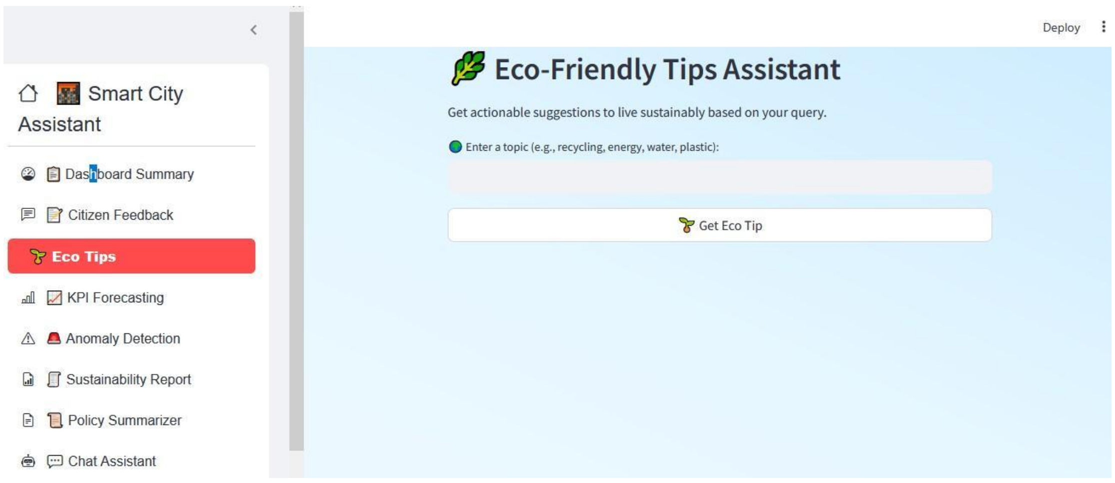
Submit Your Feedback or Report an Issue

Help us make the city better! Your feedback matters.

Your Name

Type of Issue: Garbage

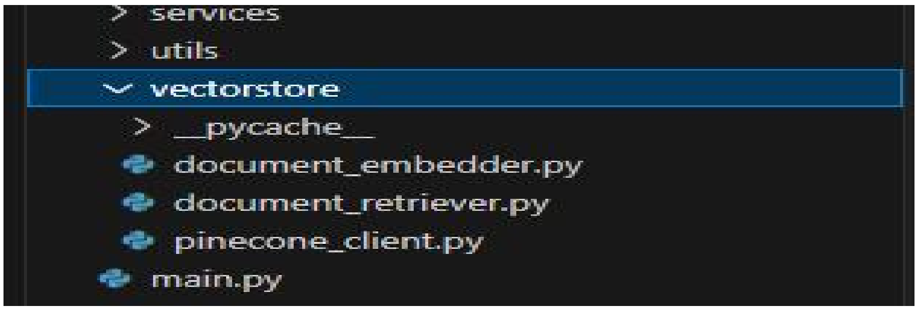
Describe the issue or suggestion



UI Enhancements Done:
Gradient backgrounds
Icon-rich sideba rusing streamlit-option-menu
Rounded buttons, font styles, padding fixes

Phase 5 – Pinecone & Document Embedding

Embedding Logic Built:
Created document_embedder.py and document_retriever.py using sentence-transformers.



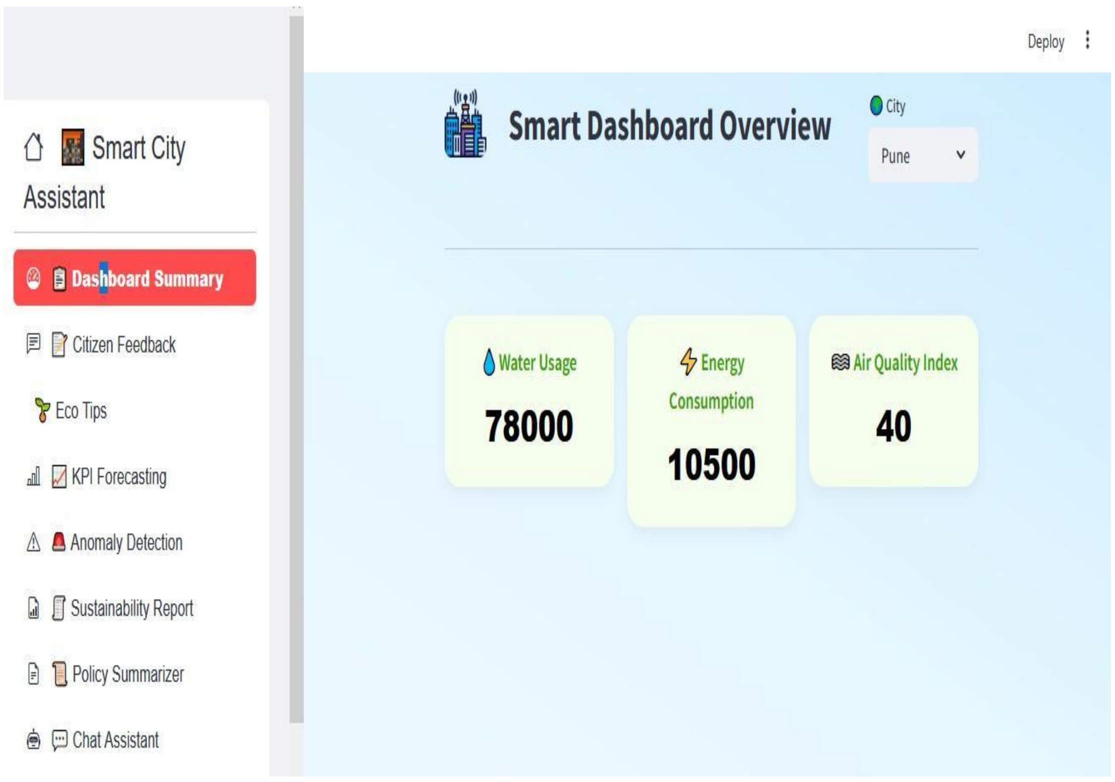
```
> services
> utils
v vectorstore
  > __pycache__
  document_embedder.py
  document_retriever.py
  pinecone_client.py
main.py
```

Phase 6 – Report Generation & Deployment

Granite LLM Report Generator:
report_generator.py takes city name and KPI data, generates detailed city sustainability report using Granite LLM prompts.

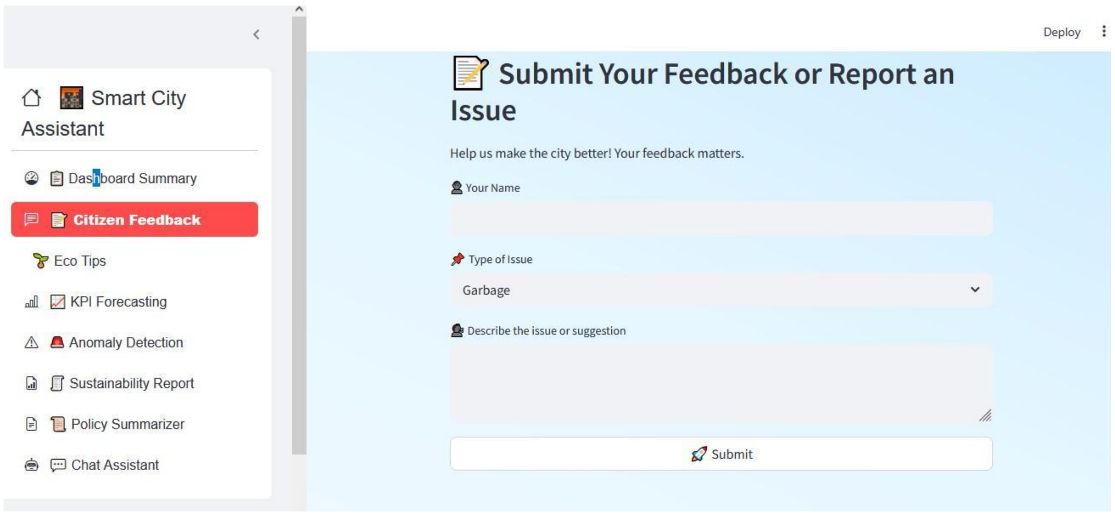
Markdown & PDF Support:
Output formatted to text block for copy/paste or PDF download (optional).
End-to-End Integration Testing:
Final dashboard tested on all 8 features: KPI dashboard, feedback form, policy summarization, eco tips, chat, anomaly check, vector search, report generation.
Phase 4 – Frontend UI Design

Streamlit UI Structure Implemented:
Created central file smart_dashboard.py with conditional rendering for each module using sidebar navigation.



Component Development:

Developed reusable Streamlit components: summary_card.py – Beautiful KPI cards chat_assistant.py – Text prompt and AI reply feedback_form.py, eco_tips.py, report_generator.py, etc.



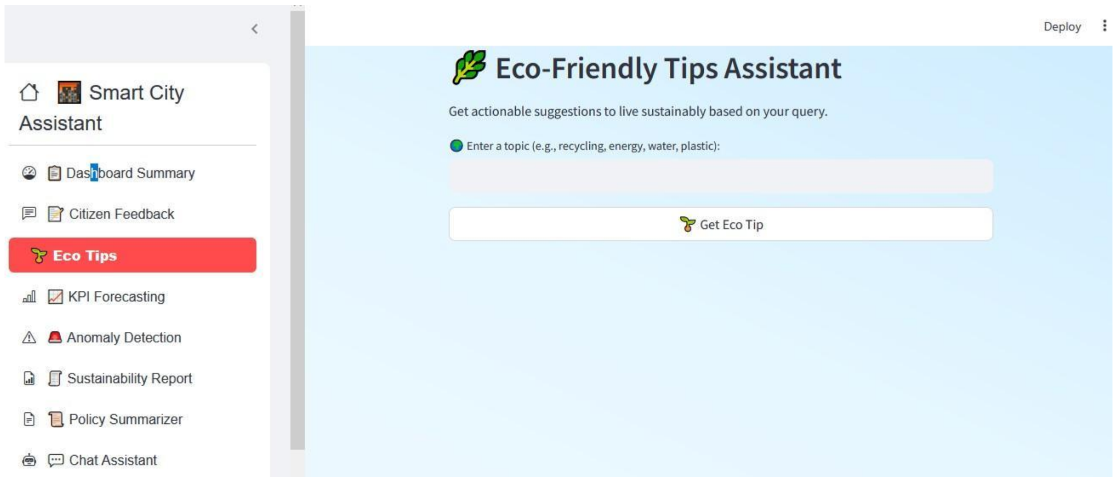
Submit Your Feedback or Report an Issue

Help us make the city better! Your feedback matters.

Your Name

Type of Issue: Garbage

Describe the issue or suggestion



Eco-Friendly Tips Assistant

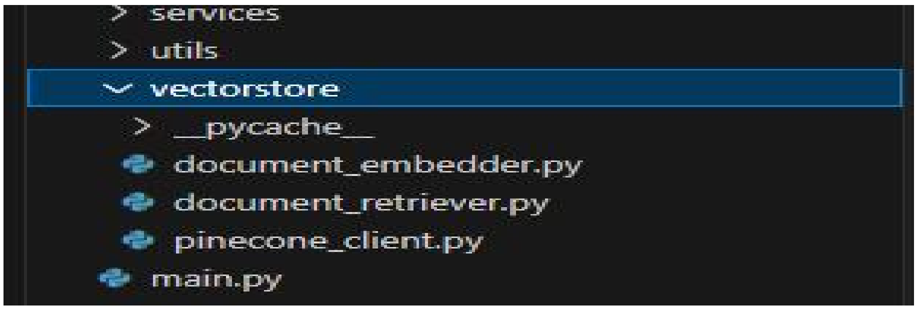
Get actionable suggestions to live sustainably based on your query.

Enter a topic (e.g., recycling, energy, water, plastic):

UI Enhancements Done:
Gradient backgrounds
Icon-rich sidebar using streamlit-option-menu
Rounded buttons, font styles, padding fixes

Phase 5 – Pinecone & Document Embedding

Embedding Logic Built:
Created document_embedder.py and document_retriever.py using sentence-transformers.



```
  > services
  > utils
  ⌄ vectorstore
    > __pycache__
      document_embedder.py
      document_retriever.py
      pinecone_client.py
    main.py
```

Phase 6 – Report Generation & Deployment

Granite LLM Report Generator:
report_generator.py takes city name and KPI data, generates detailed city sustainability report using Granite LLM prompts.
Markdown & PDF Support:
Output formatted to text block for copy/paste or PDF download (optional).
End-to-End Integration Testing:
Final dashboard tested on all 8 features: KPI dashboard, feedback form, policy summarization, eco tips, chat, anomaly check, vector search, report generation.