**SMART CITY ASSISTANT – PROJECT REPORT**

**1. INTRODUCTION**

**1.1 Project Overview**

The **Smart City Assistant** is an AI-powered platform designed to help city authorities and citizens monitor sustainability metrics, submit feedback, analyze environmental data, and improve urban management through intelligent insights.

**1.2 Purpose**

The purpose of the project is to create an interactive and dynamic system that offers:

* Real-time environmental insights
* KPI forecasting for water and energy
* Citizen feedback management
* AI-generated policy summaries and eco tips
* Anomaly detection for smart governance

**2. IDEATION PHASE**

**2.1 Problem Statement**

Urban areas struggle to manage resources efficiently, engage with citizens, and gain actionable insights from scattered data. There's a need for a centralized AI-driven solution.

**2.2 Empathy Map Canvas**

| **Says** | **Thinks** | **Does** | **Feels** |
| --- | --- | --- | --- |
| "We need better energy tracking." | "Is this data even reliable?" | Submits complaints or feedback | Frustrated, unheard |
| "Where are green spaces added?" | "I wish I could see city improvements." | Checks weather, city info | Curious, hopeful |

**2.3 Brainstorming**

* AI-powered chat assistant for city queries
* Real-time weather and environmental data
* Forecasting models using temperature/humidity
* Dynamic anomaly detection
* Feedback submission form
* Policy summarization using NLP

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

| **Step** | **User Action** | **System Response** | **Emotion** |
| --- | --- | --- | --- |
| Visit platform | Enters city | Shows metrics & forecasts | Informed |
| Wants eco tip | Inputs topic | Displays AI-generated tip | Motivated |
| Spots an issue | Fills feedback form | Sends confirmation | Valued |
| Needs summary | Pastes policy | Returns summary | Empowered |

**3.2 Solution Requirement**

* Backend APIs with AI/NLP integration
* Frontend dashboard with easy navigation
* Data forecasting and anomaly detection
* City-wise reports and visualization

**3.3 Data Flow Diagram**

**User Input → Streamlit UI → FastAPI Backend → External APIs (Hugging Face, Weather) → Response to UI**

**3.4 Technology Stack**

* **Frontend**: Python, Streamlit
* **Backend**: Python, FastAPI
* **AI Models**: HuggingFace (Mixtral, BART)
* **APIs**: OpenWeatherMap
* **Libraries**: pandas, numpy, requests

**4. PROJECT DESIGN**

**4.1 Problem Solution Fit**

The platform provides centralized access to urban KPIs, citizen engagement, and policy support, aligning directly with smart governance needs.

**4.2 Proposed Solution**

A modular, AI-assisted web application that integrates:

* Real-time weather and forecasting
* Eco-friendly suggestions
* AI summarization
* City-wide anomaly detection
* Feedback collection

**4.3 Solution Architecture**

┌────────────┐

│ Streamlit │

└─────┬──────┘

↓

┌────────────┐

│ FastAPI │

├────────────┤

│ HuggingFace│← AI Chat, Tips, Policy Summarizer

│ Weather API│← Temp, Humidity

└────────────┘

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

| **Week** | **Task** |
| --- | --- |
| 1 | Ideation & problem analysis |
| 2 | Backend setup with FastAPI |
| 3 | HuggingFace integration |
| 4 | Frontend with Streamlit |
| 5 | Testing and Optimization |
| 6 | Documentation and Deployment |

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

* **API response times** tested using local requests
* **Frontend** latency observed under normal usage
* **Load test** for concurrent feedback & summaries

**7. RESULTS**

**7.1 Output Screenshots**

* ✅ Dashboard with KPI metrics
* ✅ Eco Tip output
* ✅ Feedback form and response
* ✅ Summarized policy output
* ✅ Anomaly table and reports

*(Include actual screenshots in final document)*

**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

* AI-enhanced user experience
* Quick policy understanding
* Personalized city reports
* Interactive feedback loop

**Disadvantages**

* Dependent on external APIs (HuggingFace, OpenWeather)
* Model inference latency
* No persistent database for feedback storage

**9. CONCLUSION**

The Smart City Assistant bridges the gap between citizens and urban systems through AI and real-time data. It encourages sustainability, enhances transparency, and empowers both administrators and the public.

**10. FUTURE SCOPE**

* Add user authentication and history
* Implement database for persistent storage
* Introduce multilingual support
* Extend KPIs and visual analytics
* Deploy on cloud with CI/CD pipeline

**11. APPENDIX**

**Source Code**

* Backend: main.py (FastAPI)
* Frontend: app.py (Streamlit)

**Dataset Link**

* Not applicable (API-based)

**GitHub & Project Demo Link**

* GitHub: https://github.com/kalasapatikaveri/Sustainable\_smart\_city
* Demo: