

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

Date	28 June 2025
Team ID	LTVIP2025TMID35907
Project Name	Sustainable Smart City Assistant
Maximum Marks	4 Marks

**Functional Requirements:**

A Sustainable Smart City Assistant must be capable of understanding and responding to citizen queries through a conversational AI interface, enabling intuitive and inclusive interaction. It should forecast key performance indicators such as energy, water, and traffic usage using time-series models, while also detecting anomalies in environmental or utility data to support proactive decision-making. The assistant must summarize lengthy government policies into concise, accessible formats for public understanding. It should generate personalized eco-friendly tips based on user behavior or city data, and provide a feedback system for citizens to report issues or suggestions. Additionally, it must support the generation of downloadable sustainability reports and integrate with real-time data sources like IoT sensors and public APIs to ensure up-to-date insights and alerts. These capabilities together empower city administrators and residents to collaborate toward a greener, smarter urban future.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3		
FR-4		

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Intuitive UI/UX (e.g., Streamlit or Gradio interfaces) Multilingual and accessible design

		(voice/text, screen readers)Simple onboarding and clear feedback mechanisms
NFR-2	<b>Security</b>	End-to-end encryption (TLS/SSL)Role-based access control (RBAC)Compliance with GDPR, India's Data Protection Bill, etc.Secure APIs and audit logging
NFR-3	<b>Reliability</b>	Fault-tolerant architecture with retry logic Redundant data pipelines and backup systems Continuous monitoring and alerting
NFR-4	<b>Performance</b>	Optimized AI models (quantized/distilled for faster inference) Asynchronous processing for tasks like summarization Caching and CDN for static content
NFR-5	<b>Availability</b>	Multi-zone cloud deployment (AWS, Azure, GCP) Load balancers and failover clusters Uptime monitoring and auto-scaling
NFR-6	<b>Scalability</b>	Microservices architecture (FastAPI, Docker, Kubernetes) Horizontal scaling of compute and storage Modular APIs for easy feature expansion