# Functional & Performance Testing – Ensuring Quality and Efficiency

## 🧪 6.1 Performance Testing

To validate the responsiveness, reliability, and efficiency of the Smart City AI Assistant, performance testing was conducted across all modules within a Google Colab environment. The testing focused on both quantitative performance metrics and qualitative user experience.

### ✅ Test Environment

* **Platform:** Google Colab Pro with GPU (T4 or A100)
* **UI Framework:** Gradio
* **Test Dataset:** Custom CSVs, PDFs, and prompt inputs
* **Models Used:** Mistral-7B-Instruct, IBM Granite-2B-Instruct

### 

### 🔍 Key Performance Metrics

| **Module** | **Test Performed** | **Result** |
| --- | --- | --- |
| Chat Assistant | Prompt response time, relevance, history tracking | Avg. ~6.5s per response, highly relevant |
| PDF Summarizer | Summarization of 5–10 page documents | Output within 8–12s, accurate summary |
| KPI Forecasting | Linear regression on 100–500 rows CSV | Result in ~4s, accurate predictions |
| Anomaly Detection | Flagging spikes in CSV data | Detected outliers in < 3s |
| Eco Tips Generator | Randomized prompt and concise output | < 5s, consistent 3-point output |
| Report Generator | PDF generation with formatted paragraphs | File generated in < 3s, no formatting issues |
| Feedback Storage | Session memory and display update | Instant feedback reflection |

### 

### ⚖️ Performance Observations

* **Inference Speed:** IBM Granite is generally faster, while Mistral-7B provides better contextual understanding.
* **Memory Usage:** Optimized by limiting max\_new\_tokens and disabling unnecessary torch gradients.
* **Load Handling:** Gradio UI handled multiple inputs across tabs without crashing, suitable for 3–5 concurrent users.
* **Input Validation:** Handled improper file formats, empty inputs, and corrupted data gracefully.

### 🛠️ Performance Optimization Techniques Used

* Disabled gradient calculations using torch.inference\_mode()
* Controlled token length and output to avoid unnecessary lag
* Session-only data storage to eliminate backend overhead
* Used device\_map="auto" to maximize GPU utilization in Colab

📌 *All modules passed expected performance benchmarks and delivered real-time feedback under typical usage scenarios—ensuring a seamless user experience across devices and sessions.*