6. Functional and Performance Testing – TrafficTelligence

# A. Functional Testing

Functional testing ensures that each feature of the application operates in conformance with the requirement specification.  
For TrafficTelligence, functional testing covers the end-to-end flow including form validation, prediction logic, and output rendering.

The following test cases were executed:

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Status |
| Load homepage | Form with all input fields should load | Pass |
| Submit valid input | Prediction result is shown on the same page | Pass |
| Submit empty fields | Shows validation error or does not process | Pass |
| Input string instead of number | Displays error or prevents submission | Pass |
| Remove model file and refresh | Displays server error | Fail (Handled manually) |
| Reload after prediction | Form resets and works again | Pass |
| Use back button after submit | No session crash, safe reload | Pass |
| Cross-browser testing (Chrome/Edge) | Same prediction behavior | Pass |

# B. Performance Testing

Performance testing evaluates the speed, responsiveness, and stability of the system under a particular workload.  
Tests were conducted in a local development environment with realistic traffic volume input cases.

Key performance metrics are as follows:

|  |  |  |
| --- | --- | --- |
| Performance Metric | Target | Result |
| Model prediction response time | Less than 1 second | 0.4 sec |
| Form submission delay | Under 2 seconds | Instant |
| UI loading time | Under 2 seconds | 1.2 sec |
| Concurrent users handled | At least 5 users | 5 users (Pass) |
| CPU Usage during prediction | < 50% | ~35% (Local test) |
| Memory usage | < 200MB | ~120MB |

**Testing ensures the system performs as expected and handles user inputs and predictions accurately and efficiently.**

**🧪 A. Functional Testing**

**Goal:** Validate that each component of the application performs its intended function correctly.

**🔹 Test Scenarios:**

| **Test Case** | **Expected Result** | **Status** |
| --- | --- | --- |
| Form loads correctly on homepage | Input fields and submit button are visible | ✅ |
| Submit valid input | Predicted traffic volume is displayed | ✅ |
| Submit empty input | Proper error message or form validation | ✅ |
| Model file is missing | App shows error or handles it gracefully | ⚠️ |
| Wrong input type (e.g., text instead of number) | Shows input error or fails silently | ⚠️ |

**⚙️ B. Performance Testing**

**Goal:** Measure the speed and responsiveness of the application under expected usage.

**🔹 Metrics Evaluated:**

| **Metric** | **Description** | **Result** |
| --- | --- | --- |
| **Model Prediction Time** | Time taken to return a prediction | < 1 second ✅ |
| **Form Submission Delay** | Time between submit and output shown | Instant ✅ |
| **App Load Time** | Time to load the homepage | Fast ✅ |
| **Concurrent Users** | Can handle 3–5 test users simultaneously | Acceptable ✅ |

**📌 Notes:**

* **Testing done manually** in a local development environment (browser).
* For advanced testing, tools like **Postman**, **Selenium**, or **JMeter** can be used.