Functional And Performance Testing

Date	
Team ID	**LTVIP2025TMID31460**
Project Name	**To Supply Leftover Food to Poor**
College Name	**Ideal Institute Of Technology**

Functional Testing

Functional testing ensures that each part of the system works according to the requirements.

- **Functional Test Cases**

Module	Test Scenario	Expected Result
User Registration	Register as a restaurant, volunteer, or NGO	User is successfully registered and receives confirmation
Login	Login with correct/incorrect credentials	Success/failure message shown accordingly
Food Donation	Submit leftover food with time, quantity, and location	Donation is recorded and available for volunteers
Task Assignment	Assign pickup task to available volunteer	Volunteer is notified and task is marked “Assigned”
Collection	Volunteer updates pickup status	Status changes to “Picked Up”
Delivery	Volunteer/NGO marks delivery as completed	Status changes to “Delivered”
Feedback	Submit and view feedback	Feedback is saved and visible to relevant users
Admin Functions	View users, reports, donations	Admin dashboard displays accurate, real-time data

Performance Testing

Performance testing ensures that the application performs well under expected or unexpected load conditions.

🔹 Types of Performance Testing

Type	Purpose
Load Testing	Check system performance under expected load (e.g., 100 users simultaneously)
Stress Testing	Determine the upper limit the system can handle before crashing
Spike Testing	Evaluate behavior during sudden increases in traffic
Endurance Testing	Assess system stability over an extended period
Scalability Testing	Measure ability to scale up with increased user/data volume

---

🔹 Performance Testing Scenarios

Test Case	Expected Result
100 volunteers login at the same time	Login response time < 2 seconds
500 food donations submitted in 1 hour	No timeouts or crashes; each donation recorded
Assigning 200 tasks in bulk	Tasks are created and visible to volunteers in under 5 seconds
1,000 users use the app continuously for 24 hours	No memory leaks or database crashes
App receives a spike of 1,000 pickup updates	App responds within acceptable range (< 3 seconds per request)

🔹 Performance Testing Tools (Suggested)

Tool	Use
Apache JMeter	Simulate concurrent users, API requests
Locust	Python-based load testing
K6 (by Grafana)	Developer-centric performance testing tool
Postman + Newman	Functional and load testing of APIs
Google Lighthouse	Performance testing of web app UI
Firebase Performance Monitoring	For mobile app performance metrics

---

📊 Key Performance Metrics

Metric	Target Value
Response Time	< 2 seconds for 95% of requests
Throughput	100+ transactions/sec under normal load
Error Rate	< 1% during high load
CPU/Memory Usage	< 80% during load
Uptime	99.5% or higher