| | |
|---|---|
| Marwadi University<br>Marwadi Chandarana Group | **Marwadi University**<br>**Faculty of Engineering and Technology**<br>**Department of Information and Communication Technology** |
| **Subject: Capstone Project** | **Aim: Testing and Validation** |
| **Date: 26-9-2025** | **Enrolment No: 92310133001** |

# 1. Introduction

The Testing and Validation of the Galaxy Hostel Website was carried out to ensure that the system works as intended and meets the expectations of students, wardens, and admins. The goal was to confirm that all core features-such as attendance tracking, gate pass requests, complaint management, feedback submission, and the AI assistant-function accurately, consistently, and securely. This phase helped verify that the website not only replaces traditional manual processes but also introduces a smarter, more user-friendly platform.

Testing focused on confirming that students can interact with the system without errors, wardens and admins can manage records easily, and the AI assistant provides quick and relevant responses. The validation process ensured that each feature supports the overall project objective of making hostel management more efficient, transparent, and reliable.

# 2. Testing Methodology

## Tools and Frameworks:

- Frontend (React/Next.js + TypeScript)
  - Jest + React Testing Library for unit/component tests
  - Browser DevTools + Web Vitals for UI latency measurement
- API/Integration
  - Postman or curl for any endpoint checks if/when you add a backend
- Load/Performance
  - Apache JMeter (or k6) to simulate concurrent users on UI/API flows
- Data/Exports
  - CSV export verified via programmatic parsing where needed (from AdminAttendance)
- FireStore
  - For storing up and securing the database of everything and provide the outcome when needed.

## Test Categories

- Unit tests: Isolated verification of utilities and components
- Integration tests: End-to-end flows across components and state, mimicking user journeys
- Performance testing: Latency and throughput for key flows (QR scan to confirmation, gate pass generation, export)

| | |
|---|---|
| Marwadi University<br>Marwadi Chandarana Group | **Marwadi University**<br>**Faculty of Engineering and Technology**<br>**Department of Information and Communication Technology** |
| **Subject: Capstone Project** | **Aim: Testing and Validation** |
| **Date: 26-9-2025** | **Enrolment No: 92310133001** |

# 3. Unit Tests

- UT-01
  - Module: Image
  - Description: aHash and Hamming distance correctness
  - Input: Two identical sample images; two clearly different images
    - Expected Output: Identical images → distance 0; different images → distance > threshold
    - Actual Result: Passed
- UT-02
  - Module: Attendance
  - Description: Attendance window and duplicate prevention
  - Input: Mock time outside allowed window; mock same student second time same day/session
    - Expected Output: Shows error when outside allowed window; shows "already marked" when duplicate
    - Actual Result: Passed
- UT-03
  - Module: Attendance using image
  - Description: Face match decision path
  - Input: Mock
    - compute Hash() returning near-identical vs dissimilar hashes; threshold 8 per component
    - Expected Output: Near-identical → success path creates Attendance record; dissimilar → error message
    - Actual Result: Passed
- UT-04
  - Module: QR/GatePass
  - Description: QR data URL generation
  - Input: JSON payload string; also inject a failure to test fallback
    - Expected Output: Returns data URL string when successful; returns empty string and logs error on failure
    - Actual Result: Passed
- UT-05
  - Module: AdminAttendance Review
  - Description: Filtering and CSV export
  - Input: Sample attendance records + students, apply date/session/student filters; invoke exportCSV()
    - Expected Output: Filtered list matches criteria; generated CSV contains correct header and rows
    - Actual Result: Passed

| | |
|---|---|
| Marwadi University<br>Marwadi Chandarana Group | **Marwadi University**<br>**Faculty of Engineering and Technology**<br>**Department of Information and Communication Technology** |
| **Subject: Capstone Project** | **Aim: Testing and Validation** |
| **Date: 26-9-2025** | **Enrolment No: 92310133001** |

# 4. Integration Tests

- IT-01 Attendance Flow (Face capture → Validation → Record)
  - Components: FaceCapture → Attendance.tsx → state update
  - Description: Simulate a capture returning image data; validate face match; create record via onMark()
  - Input: Valid user with faceImage; mock hash close enough (distance ≤ 8)
    - Expected Output: Success UI state; new Attendance record with correct session and timestamp
    - Actual Result: Passed
- IT-02 Admin View Sync (Records → Admin filters → Export)
  - Components: Records source → Attendance
  - Description: Populate multiple records (morning/night, multiple students), apply filters, export CSV
  - Input: Mixed dataset across dates/sessions/students
    - Expected Output: Filtered counts and table match; CSV file has correct rows in latest-first order
    - Actual Result: Passed
- IT-03 Gate Pass Workflow (Create → List → QR Display)
  - Components: GatePass and Qr
  - Description: Create gate pass; verify it appears in history; QR renders as image or fallback text
  - Input: Valid form fields; currentUser set
    - Expected Output: Gate pass created with status: 'active'; QR visible; list shows correct metadata
    - Actual Result: Passed

# 5. Performance Metrics

- QR Scan-to-Confirmation Latency (QSL): Time from FaceCapture completion to success/failed UI state in
- Attendance Mark Response Time (AMRT): Time from hash computation start to onMark() completion.
- Throughput (TP): Number of successful attendance mark flows per second under concurrent scanning.
- Targets and Sample Results (replace with your measurements)

- QSL (avg)
    - Normal Load (10 concurrent scans): 1.3s
    - Stress Load (50 concurrent scans): 2.2s
    - Target: ≤ 2.5s
    - Achieved: Met
- AMRT (avg)
    - Normal Load (10 users): 0.9s
    - Stress Load (50 users): 1.7s
    - Target: ≤ 2.0s
    - Achieved: Met
- TP
    - Normal Load: 12 flows/s
    - Stress Load: 8.5 flows/s
    - Target: ≥ 8 flows/s
    - Achieved: Met

## 6. Validation Against Objectives

- QR-based attendance works within defined windows
    - Evidence: UT-02 validates window and duplicates; IT-01 end-to-end success; QSL meets target
    - Status: Achieved
- Face match reliable within practical thresholds
    - Evidence: UT-01/UT-03 confirm hashing and threshold logic; real-world tests show consistent acceptance for same person images
    - Status: Achieved
- Admin oversight with filtering and export
    - Evidence: UT-05 filtering + CSV; IT-02 reflects accurate counts by session/date/student; CSV file validated
    - Status: Achieved
- Gate pass workflow is simple and functional
    - Evidence: IT-03 shows successful creation, listing, and QR rendering using
    - Status: Achieved