DEMOBLAZE QA TESTING PROJECT

OVERVIEW

This document summarizes the QA testing activities performed for the Demoblaze demo e-commerce website. It includes both UI and API functional testing designed to validate core features, identify defects, and demonstrate QA documentation best practices.

SCOPE

Testing covered the following key areas: Sign-Up, Login, Cart, and Checkout functionalities (UI & API). Postman was used for API testing, Chrome DevTools for UI analysis, and Excel for test documentation.

DELIVERABLES

Excel File	TestCase_Demoblaze.xlsx with 4 tabs: UI Test Cases, API Test Cases, UI Bug Report, API Bug Report
Screenshots	Organized in separate folders: Screenshots/UI and Screenshots/API
Postman	Includes Demoblaze_API_Collection.json and Environment.json
Exports	PDF versions of all reports stored in /Exports folder

TEST EXECUTION SUMMARY

Category	Total Test Cases	Result Summary
UI	28	18 Passed, 6 Failed, 4 Not Executed
API	24	16 Passed, 2 Failed, 6 Partial (Demo Env)

ENVIRONMENT NOTES

Testing was conducted in Chrome 115 on Windows 10. Demoblaze API has session-based cart and order operations — data does not persist between sessions. Some test results marked as 'Partial Pass' due to demo backend limitations.

KEY FINDINGS

UI Bugs	• Weak password validation (BUG-001, BUG-002) • Username accepts SQL-like input (BUG-003) • Items can be added without login (BUG-004) • Missing quantity selector in Cart (BUG-005) • Mobile layout issues (BUG-006) • Video/Console JS errors (BUG-007, BUG-

	008)
API Bugs	• Empty credentials cause 500 Internal Server Error (BUG_API-001) • Invalid product ID accepted by backend (BUG_API-002)
Environment Limitations	• API lacks persistence; cart/order not stored server-side • /placeorder endpoint not implemented in demo environment

CONCLUSION

The Demoblaze QA Testing Project demonstrates a full QA documentation cycle: from test design and execution to defect tracking and reporting. All files are structured for professional portfolio presentation on GitHub. Test results, screenshots, and reports ensure traceability and QA best practices compliance.

FOLDER STRUCTURE

```
UI-API-Testing-Demoblaze-Project

├── README.md

├── 01_Documentation
│   ├── TestCase_Demoblaze.xlsx
│   └── README_Demoblaze.pdf

├── 02_Screenshots
│   ├── UI
│   └── API

├── 03_Postman
│   ├── Demoblaze_API_Collection.json
│   └── Demoblaze_API_Environment.json

└── 04_Exports
    ├── TestCase_UI.pdf
    ├── BugReport_UI.pdf
    ├── TestCase_API.pdf
    └── BugReport_API.pdf
```