Test Plan for Steadfast

Test Plan ID: TP-Steadfast-001

Introduction:

This test plan describes the approach for testing **Steadfast.com.bd**, a delivery-based web and mobile application. The goal is to validate core features, performance, and security while ensuring a seamless and safe user experience. Given the nature of logistics and real-time order management, the platform must function reliably and efficiently across all modules.

•**Application overview**

Steadfast is an on-demand delivery platform available via both web and mobile applications. It allows users to schedule pickups and deliveries, track their parcels in real-time, and manage delivery addresses. Businesses can also integrate with the platform for logistics support.

• **Purpose**

The purpose of this testing effort is to:

- Verify functional correctness of all modules (order creation, tracking, etc.)
- Ensure usability across devices and browsers
- Evaluate system performance under load
- Validate security for user data and transactions

Test Items:

List of the specific modules of the Steadfast application that will be tested:

- **User Authentication**: Login, Signup, Password Reset
- **Add Percel**: Regular, Express, Pick N Drop
- **Pickup Request**: Regular, Express, Multiple pickup, Pick N Drop
- **Pick N Drop:** add new
- **Latest Entries:** List by date
- **Payment Request:** Add payment method
- **Support:** Call, Visit Page, Open a support ticket, View Map
- **Order Management**: View orders, cancel orders, order history
  Notifications: SMS/Email updates, app alerts
- **Global UI Components**: Navbar, Footer, Responsive UI

References:

- IEEE 829 Test Documentation Standard
- Business Requirement Documents (BRD)
- UX/UI Design Mockups
- API Documentation

Features to be Tested:

This section lists the application features to be validated during testing:

- Login, Signup, Password reset
- Order scheduling and cancellation
- Delivery tracking and map updates
- Admin Support, order management
- Notifications (email/SMS/app)
- UI responsiveness across devices
- Price calculation based on delivery parameters
- Transaction Management functionalities

Features Not to be Tested:

Specifies what will be excluded from testing:

- External map APIs (Google Maps rendering)
- Third-party SMS/Email service logic
- Non-core pages (e.g., Careers, Blog)

Test Approach:

- **Phase 1**: Requirement Analysis & Test Planning
- **Phase 2**: Test Design (test cases, data sets)
- **Phase 3**: Functional Testing (manual)
- **Phase 4**: Performance Testing (basic load test)
- **Phase 5**: Compatibility Testing (different browsers/devices)
- **Phase 6**: Regression Testing after fixes

Defects will be logged, prioritized, and tracked to closure. No automation is included at this stage.

Entry Criteria:

- Conditions that must be met before testing can begin:
- Finalized business and functional requirements

- Approved UI designs and wireframes
- Test environment is stable and configured
- Test cases and data are reviewed

Exit Criteria:

Conditions to be fulfilled before the testing phase can be closed:

• All critical test cases executed.
• No high-severity defects remaining.
• Test summary report completed.

Suspension Criteria:

Situations when testing may need to be paused:

- Major blocker defects preventing further testing.
- Test environment becomes unavailable
- Major changes are introduced mid-cycle

Roles and Responsibilities:

Outlines the responsibilities of each team member:

- • **Test Lead:** Responsible for planning, monitoring, and reporting the testing process.
- **QA Engineers:** Execute test cases and log defects.
- • **Developers:** Fix bugs, assist with test data/environment setup

Schedule:

A timeline for each phase of testing:

• **Test Planning:** Week 1
• **Test Case Design:** Week 2
• **Test Execution:** Weeks 3-4
• **Defect Fixing & Regression Testing:** Week 5

Training:

Training required for effective testing:

• **Domain Training:** Delivery workflows, user flows (courier, admin, customer)
• **Process Training:** Manual testing techniques and reporting standards

Test Environment:

Details the hardware and software required for testing:

- • **Hardware:** Windows/Linux machines with 8GB RAM, SSD storage.

- • **Software:** Chrome, Firefox, Edge
- **Phone:** Android and iOS devices

Test Deliverables:

Lists the documents and reports generated during testing:

- Test Plan Document
- Mind Map
- Test Scenarios
- Test Cases and Test Data
- Test Summary Report
- Bug Report

Approvals:

Specifies who has the authority to approve the test plan and test results:

- **QA Lead, PM, Development Team**.

Glossary:

Defines key terms used in the document for clarity:

- **QA:** Quality Assurance
- **PM:** Project Manager