LIGHTNING PROXIES

Standard Operating Procedure (SOP) Document

Product Name - [Lightning Proxies](#)
Prepared By - Shivendra Singh
Project Start Date - 01 /04/2025
Project end Date - 02/04/2025

Project Description:

< Mission of project >

● Conducted testing of verify the quality of the website

< Project Output>

● Test Summary Report & Evaluation

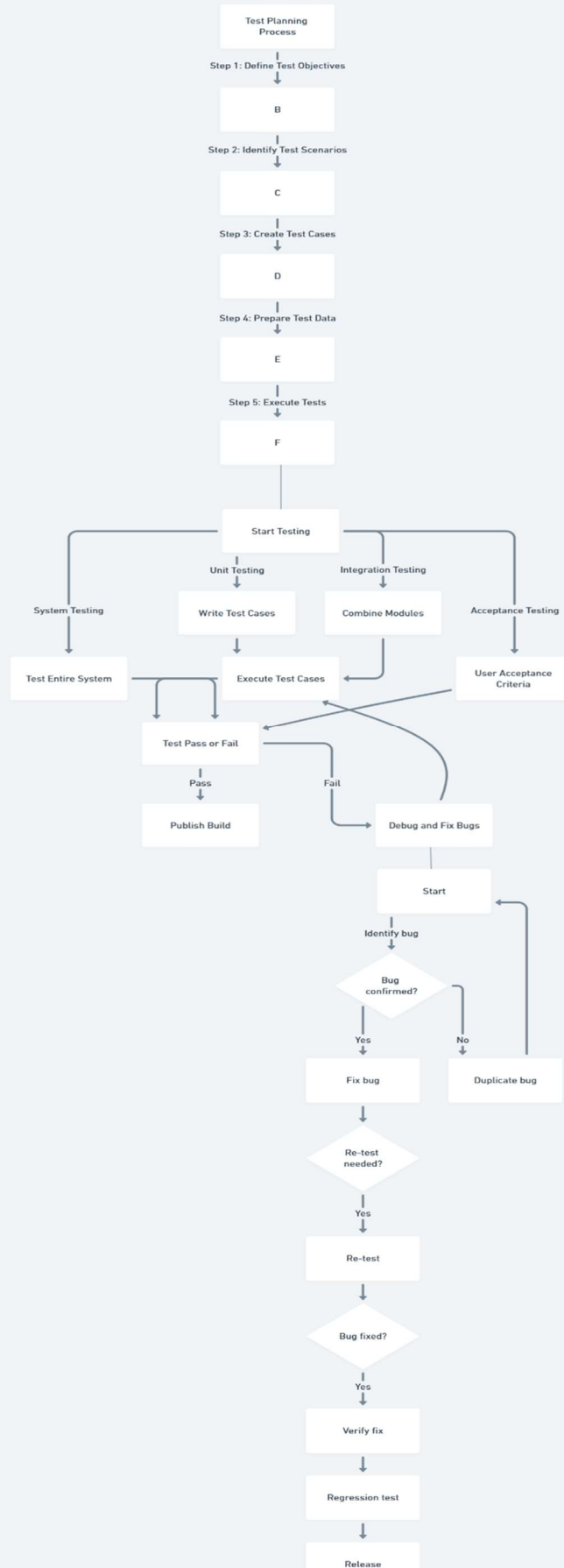● Bug / Defect Report

Table of content

Introduction

This document provides a step-by-step Standard Operating Procedure (SOP) for the QA testing process, detailing each phase from requirement analysis to test closure.

Project Overview

LightningProxies is a comprehensive proxy service provider offering a variety of proxy solutions tailored to diverse web data extraction needs. Their services include Residential, Datacenter, ISP, IPv6, and Mobile Proxies, each designed to cater to specific use cases such as web scraping, market research, and ad verification.

Flowchart

LIGHTNING PROXIES

Test Planning Process

Step 1: Define Test Objectives

B

Step 2: Identify Test Scenarios

C

Step 3: Create Test Cases

D

Step 4: Prepare Test Data

E

Step 5: Execute Tests

F

Start Testing

Unit Testing

Integration Testing

System Testing

Write Test Cases

Combine Modules

Acceptance Testing

Test Entire System

Execute Test Cases

User Acceptance Criteria

Test Pass or Fail

Pass

Fail

Publish Build

Debug and Fix Bugs

Start

Identify bug

Bug confirmed?

Yes

No

Fix bug

Duplicate bug

Re-test needed?

Yes

Re-test

Bug fixed?

Yes

Verify fix

Regression test

Release

Types of Testing Performed

Functional Testing

User Registration & Login

- ✅ Verify users can register with valid details.
- ✅ Test login with valid and invalid credentials.
- ✅ Validate "Forgot Password" functionality.

Dashboard & Proxy Management

- ✅ Ensure correct proxy purchase options are displayed.
- ✅ Verify users can add, modify, and delete proxies.
- ✅ Validate filtering and sorting options for proxies.

Payment Processing

- ✅ Test successful and failed transactions.
- ✅ Validate different payment gateways.
- ✅ Ensure secure handling of payment details.

Subscription & Renewal

- ✅ Verify subscription activation after payment.
- ✅ Ensure renewal notifications are sent.
- ✅ Validate auto-renewal settings.

UI/UX Testing

- ✅ Verify responsiveness across devices (desktop, mobile, tablet).
- ✅ Check consistency in fonts, colors, and alignment.
- ✅ Validate error messages and tooltips.

Security Testing

- ✅ Test login security (SQL injection, brute force).
- ✅ Validate secure storage of sensitive data.
- ✅ Ensure HTTPS and proper session handling.

Performance Testing

☑ Check page load times.
☑ Evaluate system response under high traffic.
☑ Identify bottlenecks and slow queries.

Defect Reporting & Tracking

All defects were logged in [JIRA/Trello/Excel], including:

- **Bug ID**
- **Description**
- **Severity (Critical, Major, Minor)**
- **Steps to Reproduce**
- **Expected vs. Actual Result**
- **Assigned To**
- **Status (Open, In Progress, Fixed, Retested, Closed)**

Regression Testing

After bug fixes, regression testing was conducted to ensure no new defects were introduced. A regression suite was maintained for repeated testing cycles.

Test Summary Report

A test summary document was prepared, detailing:

- **Test Execution Status**
- **Pass/Fail Ratio**
- **Critical Defects Identified**
- **Overall Application Stability**

Test Scope

- The test will cover the following modules:

- **User Authentication** (Login, Create Account)
- **Dashboard Functionality**

- **Deposit Balance**
- **Invoices Management**
- **Proxy Usage Tracking**
- **Purchase Plans**
- **Referral System**

Test Objectives

- Validate core functionalities of the application.
- Ensure the UI elements are properly displayed and functional.
- Detect usability and functionality defects.
- Validate proper workflow between modules.

Test Strategy

Test Types

- **Functional Testing**: Ensure that each module performs as expected.
- **UI/UX Testing**: Verify alignment, responsiveness, and accessibility.
- **Regression Testing**: Ensure changes do not break existing functionality.
- **Boundary Testing**: Check limits and edge cases.

Test Environment

- **Operating Systems**: Windows 10
- **Browsers**: Chrome, Firefox, Edge
- **Devices**: Desktop, Mobile (if applicable)

Test Scenarios

User Authentication

- Verify that users can successfully log in with valid credentials.
- Ensure an error message appears for incorrect credentials.
- Validate the account creation process with required fields.
- Confirm password validation requirements.

Dashboard

- Validate that the dashboard loads correctly with relevant information.
- Check navigation to different sections.
- Verify that real-time data updates are displayed correctly.

Deposit Balance

- Validate deposit transactions reflect correctly in the user's balance.
- Ensure error messages appear for invalid deposit attempts.

Invoices

- Verify the invoices section displays a list of invoices correctly.
- Check the ability to download and view invoices.

Proxy Usage

- Ensure users can track proxy usage history.
- Verify data updates in real-time and reflects accurate information.

Purchase Plan

- Validate the process of selecting and purchasing a plan.
- Ensure the correct amount is deducted after purchase.

Referral System

- Verify referral links are generated correctly.
- Ensure referral bonuses are credited upon successful referral completion.

Referral System

- Verify the referral link generation.
- Check if referred users are tracked properly.
- Validate the referral reward system.

Proxy Usage

- Ensure users can track proxy usage.
- Validate data accuracy in the proxy usage report.

FAQs Section

- Verify the FAQs page loads correctly.
- Check that all FAQ questions and answers are properly displayed.

- Ensure the search/filter function works correctly.
- Validate navigation and usability of the FAQs page.

Proxy Guide

- Ensure the Proxy Guide documentation is accessible.
- Verify the readability and formatting of the guide.
- Check if all proxy setup instructions are correct and up to date.
- Validate links within the guide (if any) to ensure they work.

API Documentation

- Verify that API documentation is accessible.
- Ensure API endpoints, parameters, and responses are correctly documented.
- Validate example requests and responses.
- Check if external links (if any) are functional.

Join Discord Server

- Verify that clicking on "Join Discord" redirects the user to the correct Discord server.
- Ensure users are able to join the server successfully.
- Validate any invitation or access restrictions.

Join Telegram Channel

- Verify that clicking on "Join Telegram" redirects the user to the correct Telegram channel.
- Ensure users are able to join the channel successfully.
- Validate any invitation or access restrictions.

Entry and Exit Criteria

Entry Criteria

- Application features are deployed in a test environment.
- All required test data is available.
- Test cases are prepared and reviewed.

Exit Criteria

- All high-priority defects are resolved.
- Test cases have been executed and documented.

- The application meets the business requirements.

Test Deliverables

- **Test Scenarios and Cases**
- **Bug Reports**
- **Test Execution Summary**
- **Defect Logs**

Test Scenarios

Test Scenario ID	Test Scenario	Priority
TS-01	Verify Signup Module Page	P1
TS-02	Verify Login Page	P1
TS-03	Verify Dashboard Page	P1
TS-04	Verify Purchase Plan	P1
TS-05	Verify Deposit Balance Module	P2
TS-06	Verify Invoices	P2
TS-07	Verify Proxy Usage Log	P2
TS-08	Verify Refferal	P3
TS-09	Verify FAQs Page	P0
TS-10	Verify Proxy Guide Page	P2
TS-11	Verify Join Discord Server & Telegram Channel	P4
TS-12	Verify Profile Page	P1

Test Schedule & Timeline

Activity	Start Date	End Date	Owner

Test Planning	01-04-2025	01-04-2025	Test Lead
Test Case Writing	01-04-2025	02-04-2025	Testers
Test Execution	13-03-2025	16-03-2025	Test Team
Bug Reporting & Fixing	14-03-2025	16-03-2025	Dev & Test Team
Regression Testing	14-03-2025	16-03-2025	Testers
Final Sign-Off	16-03-2025	18-03-2025	Test Manager

Test Summary

Author		Shivendra Singh
Total Test Cases	Functionality	85
	Usability	32
	Accessibility	29
	Cross-Browser & Device	32
	Security	17
	Performance	29
Test Case Status	Passed	225
	Failed	13
Project Duration	Start date	12-Mar-2025
	End date	17-Mar-2025

Conclusion

The manual testing of the **Lightning Proxies** application was conducted systematically to ensure its functionality, usability, performance, and security met the required standards. Through a structured testing approach, various critical aspects of the application—including **user registration, login, proxy management, payment processing, subscription handling, UI/UX design, security, and performance**—were thoroughly verified.

Defects identified during the testing phase were logged, tracked, and retested after resolution, ensuring a stable and reliable application. The **regression testing** process further validated that new updates and bug fixes did not introduce unintended issues.

Overall, the application demonstrated satisfactory performance and met the expected quality benchmarks. However, some areas were identified for improvement, and relevant recommendations were provided for future enhancements. The comprehensive documentation of the testing process ensures reproducibility and consistency in future testing cycles.

This **SOP serves as a reference guide** for QA testers, developers, and stakeholders, ensuring standardized testing procedures and maintaining product quality across future updates.