# Test Plan for Lyft Website

## Introduction

This test plan outlines the testing approach, scope, objectives and resources required to thoroughly test the ‘Rent a Car’ functionality on Lyft’s website (https://www.lyft.com/rider/rentals’

## Objectives

The primary objectives of testing the ‘Rent a Car’ functionality are to ensure it’s reliability, functionality, useability and performance. This test plan aims to identify any defects, inconsistencies or issues that may arise during the rental process.

## Scope

The testing scope encompasses all aspects of the ‘Rent a Car’ functionality available on Lyft’s website. This includes but not limited to

- User registration and login
- Car selection and availability
- Booking and reservation process
- Payment processing
- Cancellation and refund process
- Booking management

## Test Strategy

The testing strategy for the ‘Rent a Car’ functionality involves a combination of both manual and automated testing. Both functional and non-functional aspects will be assessed too.

# Test Environment

The testing environment will consist of various browsers (Chrome, Firefox, Safari, Edge etc) and devices (tablet, mobile) to ensure cross-browser and cross-device compatibility.

# Test Cases

## Registration and Login

User can create an account using valid information
User can log in using registered credentials
Validate behaviour when incorrect login credentials are entered

## Car Selection and Availability

Check the list of available cars is correctly displayed
Verify users can sort and filter cars based on specific criteria
Ensure selected car details match the displayed information

## Booking and Reservation

Test booking process for different car types, dates and durations
Verify users receive confirmation of their booking
Check booked cars are reserved and not available for other users – is this possible?

## Payment Processing

Test various payment methods (Credit card, Debit card, Paypal?)
Ensure accurate calculations of rentals
Verify payment processing is secure and user information is protected

## Cancellation and Refund

Test cancellation process for booked rentals
Verify users receive information on cancellation and are refunded appropriately
Check cancelled cars become available again for others to book (is this possible)

## User Dashboard and Booking Management

Verify user can see both active and previous bookings
Test update user profile and preferences

## Test Execution

Testing will be conducted in multiple phases:

- Unit Testing – individual components and function will be tested in isolation
- Integration Testing – Interactions between different components will be tested
- System Testing – The entire ‘Rent a Car’ system will be tested end-to-end
- Regression Testing – After each change, previously working functionalities will be retested to ensure they haven’t been adversely affected using an Automation Test pack.

## Performance Testing

Load testing will be performed to determine system behaviour under peak load conditions. This will ensure that the system can handle multiple users booking cars simultaneously without performance degradation

## Security Testing

The security of user information and payment processing will be assessed to identify vulnerabilities and ensure compliance with data protection standards

## Defect Management

Defects will be tracked using a suitable tool i.e. Jira. Each defect will be categorised, prioritised, assigned and tracked to resolution.

## Test Reporting

Test status reports will be generated, including metrics on passed, failed and pending test cases. A final test summary report will be created to provide an overview of the testing process, results and any recommendations for further improvement.