**Section 1: Manual Testing – Garibook Website**

**Step 1: Test Planning & Requirement Analysis**

**Testing Approach:**  
The testing approach follows the **Software Testing Life Cycle (STLC)**, including:

1. **Requirement Analysis** – Understanding trip booking flow and business logic.
2. **Test Planning** – Defining test scope, approach, and resources.
3. **Test Case Design** – Creating functional and non-functional test cases.
4. **Test Execution** – Manually testing the website.
5. **Defect Reporting** – Logging issues with proper steps and screenshots.
6. **Test Closure** – Making a release decision based on results.

**Key Functional Requirements:**

1. Users should be able to log in with an OTP-based authentication using a mobile number.
2. Users should be able to select a One Way, Round Way, or Hourly trip and book a car.
3. Users should be able to apply a promo code successfully for intercity trips.

**Key Non-Functional Requirements:**

1. **Performance:** The website should load quickly, and the booking process should be responsive.
2. **Usability:** The interface should be intuitive, and the booking process should be easy to understand and complete.

**Step 2: Test Scenario & Test Cases Creation**

**Critical Test Scenarios for Trip Booking:**

1. Verify that a user can successfully log in using OTP authentication.
2. Verify that a user can select a "One Way" trip in car rental and proceed to booking.
3. Verify that a user can select a "Round" trip in car rental and proceed to booking.
4. Verify that a user can select an "Hourly" trip in car rental and proceed to booking.
5. Verify that a valid promo code applies successfully, reducing the fare.

**Test Cases for the Scenarios:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | **Test Case ID** | **Test Case Title** | **Test Steps** | **Test Data** | **Precondition** | **Expected Result** |
| 1 | GB\_TC\_1 | Login Attempt with Valid Mobile Number | 1. Navigate to the login page.  2. Enter a valid mobile number.  3. Click on "Send Code". | 01712345678 | N/A | An OTP will be sent to the entered number. |
| GB\_TC\_2 | Login Attempt with Invalid Mobile Number | 1. Navigate to the login page.  2. Enter an invalid mobile number.  3. Click on "Send Code". | 01389173274 | N/A | A validation error message is displayed, prompting for a valid mobile number. |
| GB\_TC\_3 | Successful Login with Valid OTP | 1. Navigate to the login page.  2. Enter a valid mobile number.  3. Click on "Send Code".  4. Retrieve the OTP.  5. Enter the OTP within the validity period.  6. Click on "Continue". | 01712345678,  OTP | The entered mobile number should be valid | The user is logged in and redirected to the homepage. |
| GB\_TC\_4 | Login Attempt with Incorrect OTP | 1. Navigate to the login page.  2. Enter a valid mobile number.  3. Click on "Send Code".  4. Enter a wrong OTP within the validity period.  5. Click on "Continue". | 01712345678,  OTP | The entered mobile number should be valid | An error message is displayed stating that “The OTP you’ve entered is incorrect. Please try again” |
| GB\_TC\_5 | Login Attempt with Expired OTP | 1. Navigate to the login page.  2. Enter a valid mobile number.  3. Click on "Send Code".  4. Retrieve the OTP.  5. Enter the OTP after the validity period.  6. Click on "Continue". | 01712345678,  OTP | The entered mobile number should be valid | An error message is displayed stating that “The OTP you’ve entered is expired. Please try again” |
| 2 | GB\_TC\_6 | Successfully Select "One Way" and Proceed to Booking | 1. Choose "One Way" and select a car in choosing trip option.  2. Enter pick-up location.  3. Enter drop-off location.  4. Choose a valid date and time.  5. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | Booking will be confirmed and go to the trip details page. |
| GB\_TC\_7 | "One Way" Trip with Missing or invalid Drop-off or Pickup Location | 1. Choose "One Way" and select a car in choosing trip option.  2. Enter blank or invalid pick-up location.  3. Enter blank or invalid drop-off location.  4. Choose a valid date and time.  5. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | Specific error messages will be shown for corresponding errors. |
| GB\_TC\_8 | "One Way" Trip with Invalid Date/Time Selection (e.g., Past Date) | 1. Choose "One Way" and select a car in choosing trip option.  2. Enter pick-up location.  3. Enter drop-off location.  4. Choose a past date and time.  5. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | An error message is displayed indicating that the selected date/time is invalid. |
| 3 | GB\_TC\_9 | Successfully Select "Round Way" Trip and Proceed to Booking | 1. Choose "Round Way" and select a car in choosing trip option.  2. Enter pick-up location.  3. Enter drop-off location.  4. Choose a valid pickup date and time.  5. Choose a valid return date and time  6. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | Booking will be confirmed and go to the trip details page |
| GB\_TC\_10 | "Round Way" Trip with Missing or invalid Drop-off or Pickup Location | 1. Choose "Round Way" and select a car in choosing trip option.  2. Enter blank or invalid pick-up location.  3. Enter blank or invalid drop-off location.  4. Choose a valid pickup date and time.  5. Choose a valid return date and time  6. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | Specific error messages will be shown for corresponding errors. |
| GB\_TC\_11 | "Round Way" Trip with Invalid pickup and return Date/Time Selection (e.g., Past Date) | 1. Choose "Round Way" and select a car in choosing trip option.  2. Enter pick-up location.  3. Enter drop-off location.  4. Choose a past pickup date and time.  5. Choose a return date and time that is before pickup time  6. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | An error message is displayed indicating that the selected date/time is invalid. |
| 4 | GB\_TC\_12 | Successfully Select "Hourly" Trip and Proceed to Booking | 1. Choose "Hourly" and select a car in choosing trip option.  2. Enter pick-up location.  3. Choose hours.  4. Choose a valid pickup date and time.  5. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | Booking will be confirmed and go to the trip details page |
| GB\_TC\_13 | "Hourly" Trip with invalid Duration | 1. Choose "Hourly" and select a car in choosing trip option.  2. Enter pick-up location.  3. Choose less than 2 hours or negative hours.  4. Choose a valid pickup date and time.  5. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | It will not let pick less than 2 hours in the duration. |
| GB\_TC\_14 | "Hourly" Trip with Missing or invalid Pickup Location | 1. Choose "Hourly" and select a car in choosing trip option.  2. Enter blank or invalid pick-up location.  3. Choose hours.  4. Choose a valid pickup date and time.  5. Click "Request Your Trip” |  | User is logged in and on the car rental booking page. | Specific error messages will be shown for corresponding errors. |
| GB\_TC\_15 | "Hourly" Trip with Invalid Date/Time Selection (e.g., Past Date) | 1. Choose "Hourly" and select a car in choosing trip option.  2. Enter blank or invalid pick-up location.  3. Choose hours.  4. Choose a past pickup date and time.  5. Click "Request Your Trip”. |  | User is logged in and on the car rental booking page. | An error message is displayed indicating that the selected date/time is invalid. |
| 5 | GB\_TC\_16 | Apply a valid promo code on an intercity trip | 1. Select any type of trip and car.  2. Enter Pickup and Drop-off location (It should be different cities)  3. Enter a valid promo code.  5. Click "Request Your Trip”. |  | User is logged in and on the car rental booking page. | The Promo code is successfully applied and the fare is reduced according to the promo code discount. |
| GB\_TC\_17 | Apply a valid promo code on a within-city trip | 1. Select any type of trip and car.  2. Enter Pickup and Drop-off location (Both locations should be in the same city)  3. Enter a valid promo code.  5. Click "Request Your Trip”. |  | User is logged in and on the car rental booking page. | It displays an error message that “Promo code is only valid for intercity trips.” |
| GB\_TC\_18 | Apply an invalid promo code on an intercity trip | 1. Select any type of trip and car.  2. Enter Pickup and Drop-off location (It should be different cities)  3. Enter an invalid promo code.  5. Click "Request Your Trip”. |  | User is logged in and on the car rental booking page. | It displays an error message that “Invalid Promo code.” |

**Step 3: Test Execution & Defect Reporting**

After a manual test on the Garibook website the Bug report for the found bugs below-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Bug Title** | **Bug Description** | **Steps to Reproduce** | **Expected Result** | **Actual Result** | **Priority** | **Severity** | **Status** | **Attachment** |
| B\_01 | OTP Expiry Not Being Enforced During Login | When a user enters an expired OTP for the login, the system incorrectly accepts the expired OTP and successfully logs the user in. | 1. Go to the Garibook login page.  2. Enter a valid mobile number in the phone number field.  3. Click on the 'Send Code' button to request the OTP.  4. Wait for the OTP to expire.  5. Enter the expired OTP in the OTP input field.  6. Click the 'Continue' button. | The system should reject the expired OTP and display an error message, instructing the user to request a new OTP. | The system accepts the expired OTP and successfully logs the user in. | High | Critical | Open |  |
| B\_02 | Missing Validation and Error Messages for Passenger Details Input | When entering passenger details (email, date of birth), the system does not display clear error messages or validation prompts for incorrect input formats, such as an invalid email. | 1. Login for the first time.  2. Request the first trip  3. Enter an invalid date of birth (Future date) and invalid email in the prompt passenger information form.  4. Click on continue. | The system should display clear validation error messages for each invalid input. | No error messages or validation prompts appear, and the form is submitted with invalid data. | Low | Low | Open |  |
| B\_03 | Date Picker Allows Selecting Past time | The date picker allows selecting past time for trips, but when the "Request Your Trip" button is clicked, an error message appears on the results page stating, "Please select a valid date." This inconsistency can confuse users. | 1. Go to the trip request page.  2. Enter Pickup and Drop-off location.  3. Pick a time that already passed.  4. Click on “Request Your Trip” | The date picker should prevent the selection of past time. | The date picker allows selecting past time, but after clicking "Search," an error message is shown stating "Please select a valid date." | Low | Trivial | Open |  |
| B\_04 | Invalid Stoppage Location Allowed During Trip Booking | The trip booking process allows the user to enter an invalid stoppage location and proceed to confirm the booking without displaying an error or validation prompt. | 1. Go to the trip request page.  2. Enter Pickup and Drop-off location.  3. Enter an invalid stoppage location.  3. Pick a pickup date and time.  4. Click on “Request Your Trip” | The system should display an error or validation prompt indicating that the stoppage location is invalid and prevent the user from confirming the booking. | The system allows the invalid stoppage location and confirms the booking without any error or validation message. | High | Medium | Open |  |
| B\_05 | Unhandled Runtime Error When Entering Invalid Pickup/Drop-off Location During Trip Booking | When a user initially enters invalid pickup or drop-off location multiple times, then corrects the location and clicks the "Request Your Trip" button, an unhandled runtime error is thrown. | 1. Go to the trip request page.  2. Enter invalid Pickup and Drop-off location multiple times then put valid locations.  3. Pick a pickup date and time.  4. Click on “Request Your Trip” | The system should handle the validation of locations properly and allow the user to successfully proceed with the booking after entering a valid location without throwing any errors | After correcting the invalid location and clicking the "Request Your Trip" button, an unhandled runtime error is thrown. | High | High | Open |  |
| B\_06 | Server Error When Directly Accessing "My Trips" Page URL | When a user directly  enters the URL https://  garibook.com/  trip/my-trips in the browser, a server error occurs with the following error message:  TypeError: Cannot set property 'ResponsiveMasonry' of #<Object> which has only a getter.  The error happens while generating the page, and any console logs related to this will be displayed in the terminal window. | Open a browser and directly enter the URL https://garibook.com/trip/my-trips.  Observe the server error with the message TypeError: Cannot set property 'ResponsiveMasonry' of #<Object> which has only a getter. | The page should load successfully without any errors when the URL is directly accessed. | A server error occurs with the message TypeError: Cannot set property 'ResponsiveMasonry' of #<Object> which has only a getter, preventing the page from loading. | High | Critical | Open |  |

Step 4: Test Closure & Release Decision

Release Decision:

1. Test Execution Review:
   * After manually testing the Garibook website trip booking process, all test scenarios were executed successfully with a few defects found during testing.
   * Critical test cases, such as invalid pickup or drop-off location handling and form validation, were executed and found to have issues in some scenarios. The test cases for correct user flow, including OTP verification and trip booking confirmation, passed successfully.
2. Defects Identified:
   * Several defects were found during testing, such as:
     + Invalid stoppage location was allowed to proceed to the booking confirmation without error prompts.
     + An unhandled runtime error occurred when the user corrected a previously invalid pickup/drop-off location and clicked "Request Your Trip."
     + A server error was encountered when directly accessing the URL https://garibook.com/trip/my-trips, with the error message indicating a problem with the ResponsiveMasonry property.
3. Defect Severity & Priority:
   * The defects related to invalid location handling and the unhandled runtime error are considered high priority since they impact user experience directly and may prevent users from completing the booking process.
   * The server error is critical and should be resolved before the feature is released, as it prevents page access entirely.
4. Bug Fixing Status:
5. Test Coverage and Risk Assessment:

Despite the defects, the website's main features, such as trip booking, OTP verification, and form submissions, function correctly.

Users may encounter some issues, but they should still be able to complete bookings in most cases.

Final Decision:

Approval Status: Approved for Release with the understanding that the identified defects will be fixed in future updates.

Justification: The trip booking flow is functional, and major features work as expected. While the identified defects are concerning, they do not completely block the user from completing the booking process. The risks are manageable, and the fixes can be scheduled for the next release cycle. Moving forward with the release will allow users to access the website and use the core features while the development team works on resolving these issues.

Release Decision: Pass (Approved for release with known issues to be addressed in a future update).

**Section 2: Garibook Website Automation Testing**

For automation, I'll write a **Selenium** script in **Python** to:

* Open Garibook website
* Enter a phone number
* Extract OTP from the screen
* Enter OTP & complete login
* Select "One Way" trip
* Pick a Sedan
* Enter Pickup & Drop-off locations
* Select Date & Time
* Apply Promo Code
* Submit trip request

The script is given below-

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

import re

# Initialize the WebDriver

driver = webdriver.Chrome()

try:

# Open Garibook website

driver.get("http://fe.garibook.com/")

# Click on the 'Log in' button

login\_button = WebDriverWait(driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Log in"))

)

login\_button.click()

# Enter phone number

phone\_input = WebDriverWait(driver, 10).until(

EC.presence\_of\_element\_located((By.NAME, "mobile"))

)

phone\_number = "01905302401" # Replace with your phone number

phone\_input.send\_keys(phone\_number)

# Submit phone number to receive OTP

send\_code\_button = driver.find\_element(By.XPATH, "//button[text()='Send Code']")

send\_code\_button.click()

# Wait for OTP to appear on the screen

otp\_element = WebDriverWait(driver, 30).until(

EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "div.alert.alert-info"))

)

otp\_text = otp\_element.text # This will return "Your OTP is: 7680"

otp\_match = re.search(r"(\d+)", otp\_text)

if otp\_match:

otp\_code = otp\_match.group(1)

print("Extracted OTP:", otp\_code)

else:

raise Exception("OTP code not found in the alert text.")

if len(otp\_code) != 4:

raise Exception("Expected OTP code with 4 digits, got: " + otp\_code)

# Enter each digit into its corresponding OTP input field

driver.find\_element(By.ID, "otp1").send\_keys(otp\_code[0])

driver.find\_element(By.ID, "otp2").send\_keys(otp\_code[1])

driver.find\_element(By.ID, "otp3").send\_keys(otp\_code[2])

driver.find\_element(By.ID, "otp4").send\_keys(otp\_code[3])

# Now click the Continue button (assuming it's always clickable)

continue\_button = driver.find\_element(By.XPATH, "//button[text()='Continue']")

continue\_button.click()

# Select "One Way" trip

# Locate the "Continue" button using its class and click it

continue\_button = driver.find\_element(By.XPATH, "//button[contains(@class, 'gb-primary-solid-btn') and text()='Continue']")

continue\_button.click()

# Enter Pickup location

pickup\_input = driver.find\_element(By.NAME, "pickup")

pickup\_location = "Banani DOHS, Dhaka, Bangladesh" # Replace with your pickup location

pickup\_input.send\_keys(pickup\_location)

time.sleep(2) # Wait for location suggestions to appear

pickup\_input.send\_keys(Keys.DOWN, Keys.RETURN)

# Enter Drop-off location

dropoff\_input = driver.find\_element(By.NAME, "dropoff")

dropoff\_location = "Nikunja 1, Dhaka, Bangladesh" # Replace with your drop-off location

dropoff\_input.send\_keys(dropoff\_location)

time.sleep(2) # Wait for location suggestions to appear

dropoff\_input.send\_keys(Keys.DOWN, Keys.RETURN)

# Select Date & Time

date\_input = driver.find\_element(By.NAME, "date")

pickup\_date = "2025-02-20" # Replace with your desired date

date\_input.send\_keys(pickup\_date)

time\_input = driver.find\_element(By.NAME, "time")

pickup\_time = "10:00 AM" # Replace with your desired time

time\_input.send\_keys(pickup\_time)

# Apply Promo Code

promo\_input = driver.find\_element(By.NAME, "promo")

promo\_code = "" # Replace with your promo code

promo\_input.send\_keys(promo\_code)

# Submit trip request

submit\_button = driver.find\_element(By.XPATH, "//button[contains(text(), 'Submit')]")

submit\_button.click()

# Wait for confirmation

confirmation\_message = WebDriverWait(driver, 10).until(

EC.presence\_of\_element\_located((By.CLASS\_NAME, "confirmation"))

)

print("Booking confirmed:", confirmation\_message.text)

finally:

# Close the WebDriver

driver.quit()