



# Restful Booker Demo Project

## ( Testing Phase )

# Team Members

**TEAM ID : YAT384B\_G(2)**



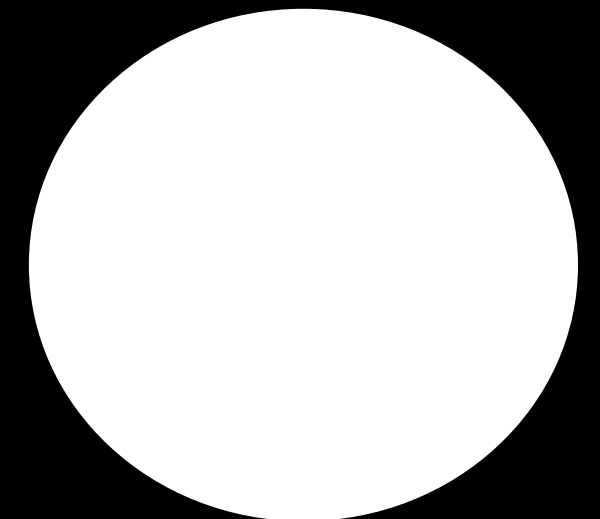
**Merna Tarek**



**Farida Mohamed**



**Maria Kaiser**



**Wessam**

**Under Supervision :**

**Eng / Mariam Zayed**

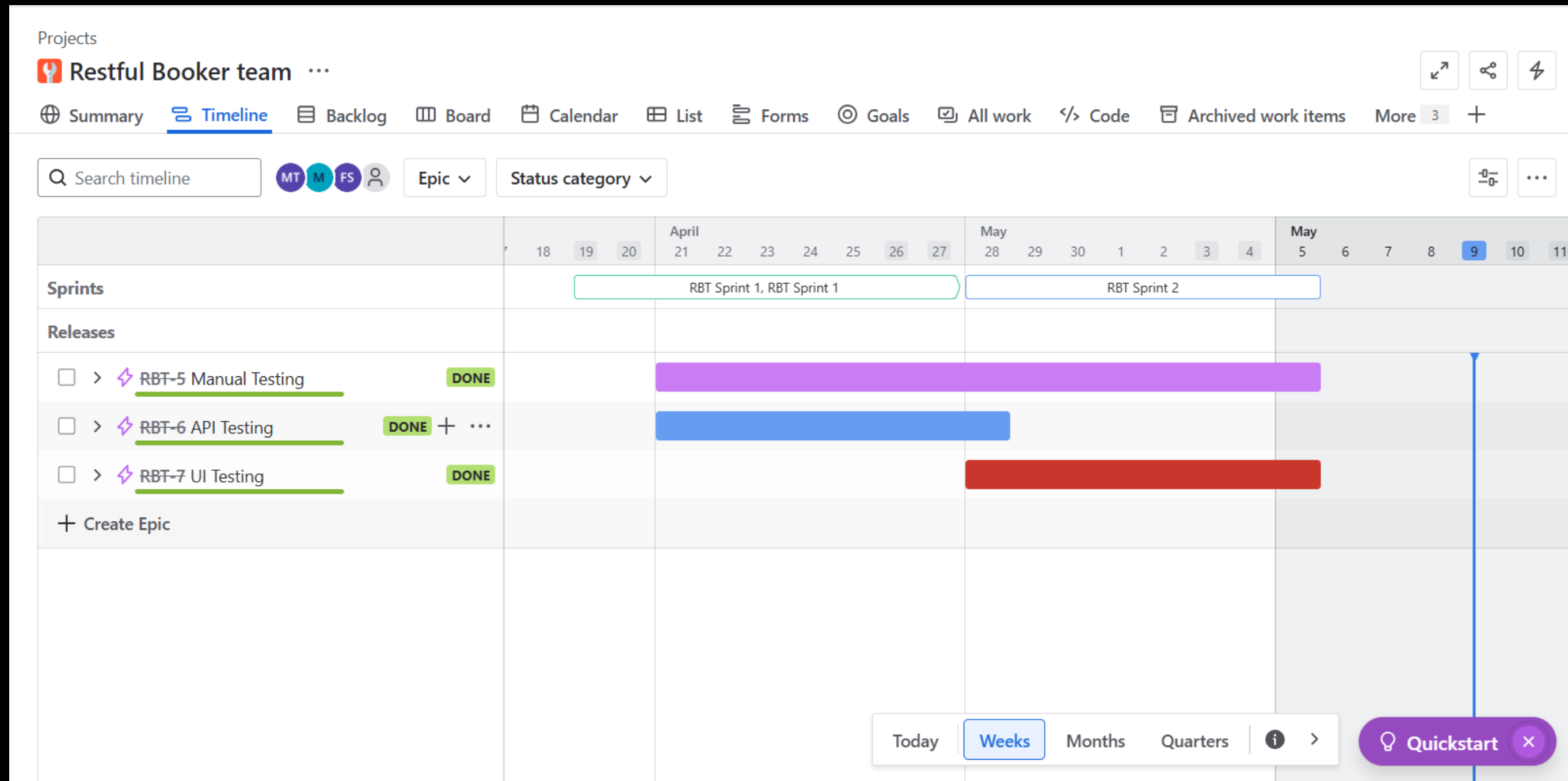
## GitHub

<https://github.com/mernatarek1/DEPI-project.git>

Drive Link : [https://drive.google.com/drive/folders/15ckVjvL1\\_gHVvEKaNaMFpejGaVP62w-N?usp=sharing](https://drive.google.com/drive/folders/15ckVjvL1_gHVvEKaNaMFpejGaVP62w-N?usp=sharing)

# Jira

<https://mernatarekahmed.atlassian.net/jira/software/projects/RBT/boards/36?atlOrigin=eyJpIjoiOWI1ZmY5NWl2NWRkNDcxZWJhMDhhY2JmOGYxN2U2OWYiLCJwIjoiajI9>



# Agenda

**01 Project Overview**

**02 Test Scope**

**03 Test Approach**

**04 Test Scenarios**

**05 Automation Architecture**

**06 Demo**

**07 Results & Reports**

**08 Defect Management**

**09 Challenges Faced**

**10 Conclusion & Learnings**

# 1. Project Overview

- **Project Title:**

*Functional & Non-Functional Testing of Restful Booker platform.*

- **Application Under Test:**

Restful-Booker platform (deployed at *automationintesting.online*)  
a public, open-source hotel booking demo site and API designed for automation.

- **Objective:**

End-to-end testing of the Restful-Booker web application (UI and API), covering key user flows (Room booking, Admin login, contact) and ensuring quality via both manual and automated tests. The goal is to validate functionality and quality attributes using industry tools.

- **Tools & Frameworks:**

Selenium WebDriver (UI automation), TestNG (test framework), Java, Maven (build management), Page Object Model (design pattern), Postman (API testing), JIRA (defect tracking), ExtentReports (HTML & htmlextra reporting), IntelliJ IDEA (IDE).

## 2. Test Scope

- **Functional Testing (UI & API):**

- **Room Booking flow** (entering booking details and confirming reservation).
- **Admin Login** (authentication via the login page and API endpoint).
- Contact Form Validation (ensuring required fields and validations).

- **Non-Functional Testing:**

- **Usability & UI Consistency:** Verify that the interface is intuitive and elements are aligned; the application should be easy to use (**Other non-functional aspects** like performance or security are *out-of-scope* for this project.)

- **In-Scope Modules:**

- **Booking Page:** UI where users reserve rooms.
- **Admin Dashboard/Login:** UI/API area for admin authentication and management.
- **Contact Form:** UI form for user inquiries (email, name, comments).

- **Out-of-Scope:**

- Backend database testing; security/penetration testing; any module not listed above (to keep the focus on front-end and API flows).

## 3. Test Approach

### •Manual Testing:

- Design and execute comprehensive test cases for each in-scope feature.
- Perform exploratory testing of flows (e.g. booking variations, form inputs).
- Log and track defects.
- Make a Traceability matrix.

### •Automation Testing:

- UI Automation:** Selenium WebDriver (Java + TestNG).
  - We implement the Page Object Model (POM) design pattern – modeling each page as a class to reduce code duplication and simplify maintenance.
  - Selenium tests navigate the UI, enter data, and verify results.
  - TestNG orchestrates test execution and grouping.
- API Automation:** Postman (and Newman) for REST API testing.
  - We automate calls to endpoints.



## 4. Test Scenarios

### •API :

#### 1- Booking Room

- TC1: Verify Successful GET /booking (List Bookings)
- TC2: Verify Successful GET /booking/:id (Get Booking Details)
- TC3: Verify Successful POST /booking (Create Booking)
- TC5: Verify Successful POST /auth (Authentication)
- TC6: Verify Successful PUT /booking/:id (Update Booking)
- TC7: Verify Successful PATCH /booking/:id (Partial Update)
- TC8: Verify POST /booking with Missing Required Field
- TC9: Verify POST /booking with Invalid Date Format
- TC10: Verify GET /booking/:id with Non-Existent ID
- TC11: Verify POST /booking with Empty Body
- TC12: Verify POST /auth with Invalid Credentials
- TC13: Verify PUT /booking/:id without Token

#### 2- Admin Login

- TC14: Verify Successful Admin Login with Valid Credentials
- TC15: Verify Admin Can Access Bookings After Login
- TC16: Verify Invalid Admin Credentials Return an Error
- TC17: Verify Missing Username in Admin Login
- TC18: Verify Missing Password in Admin Login
- TC19: Verify Token Usage for Protected Endpoint (PUT /booking/:id)

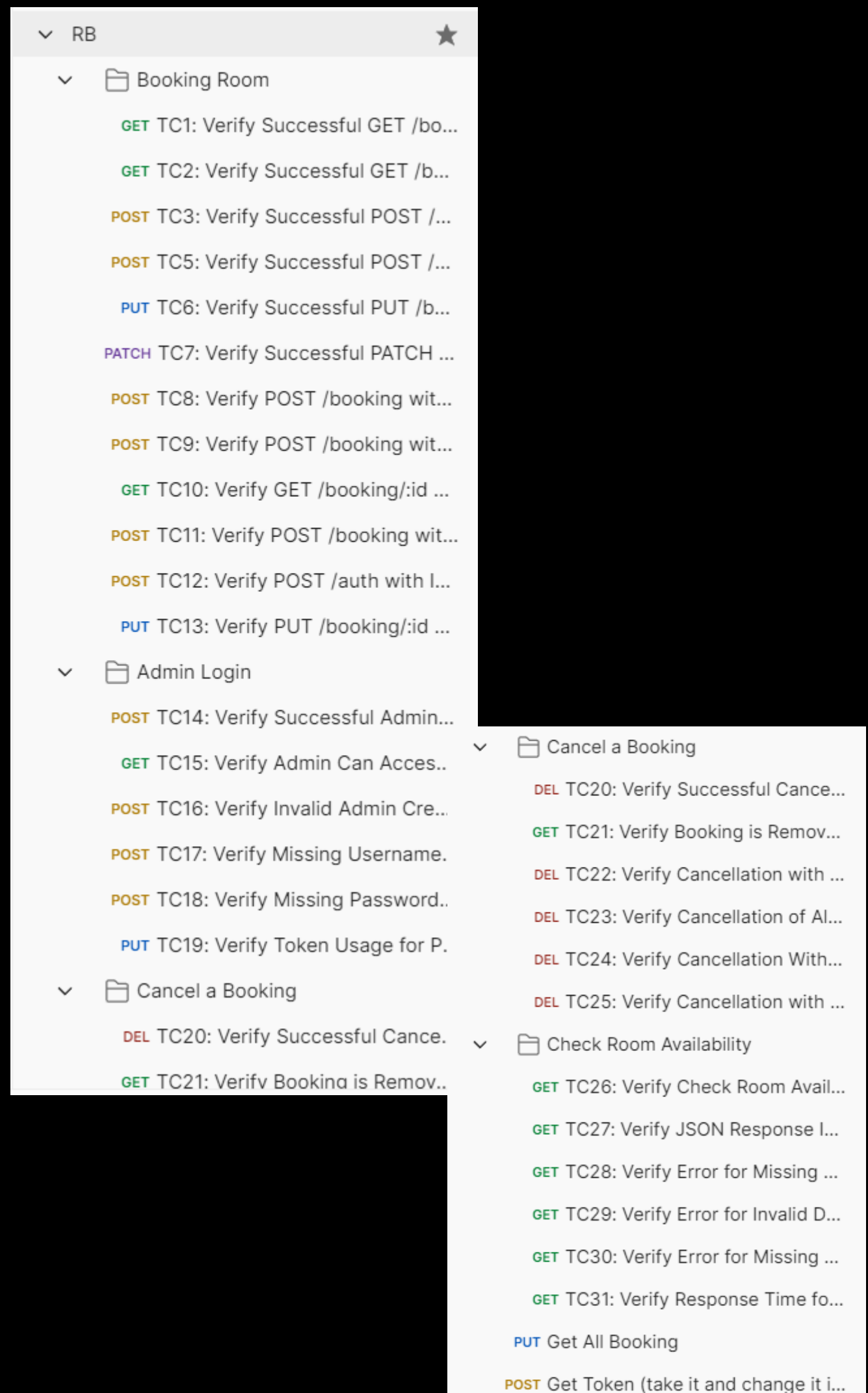
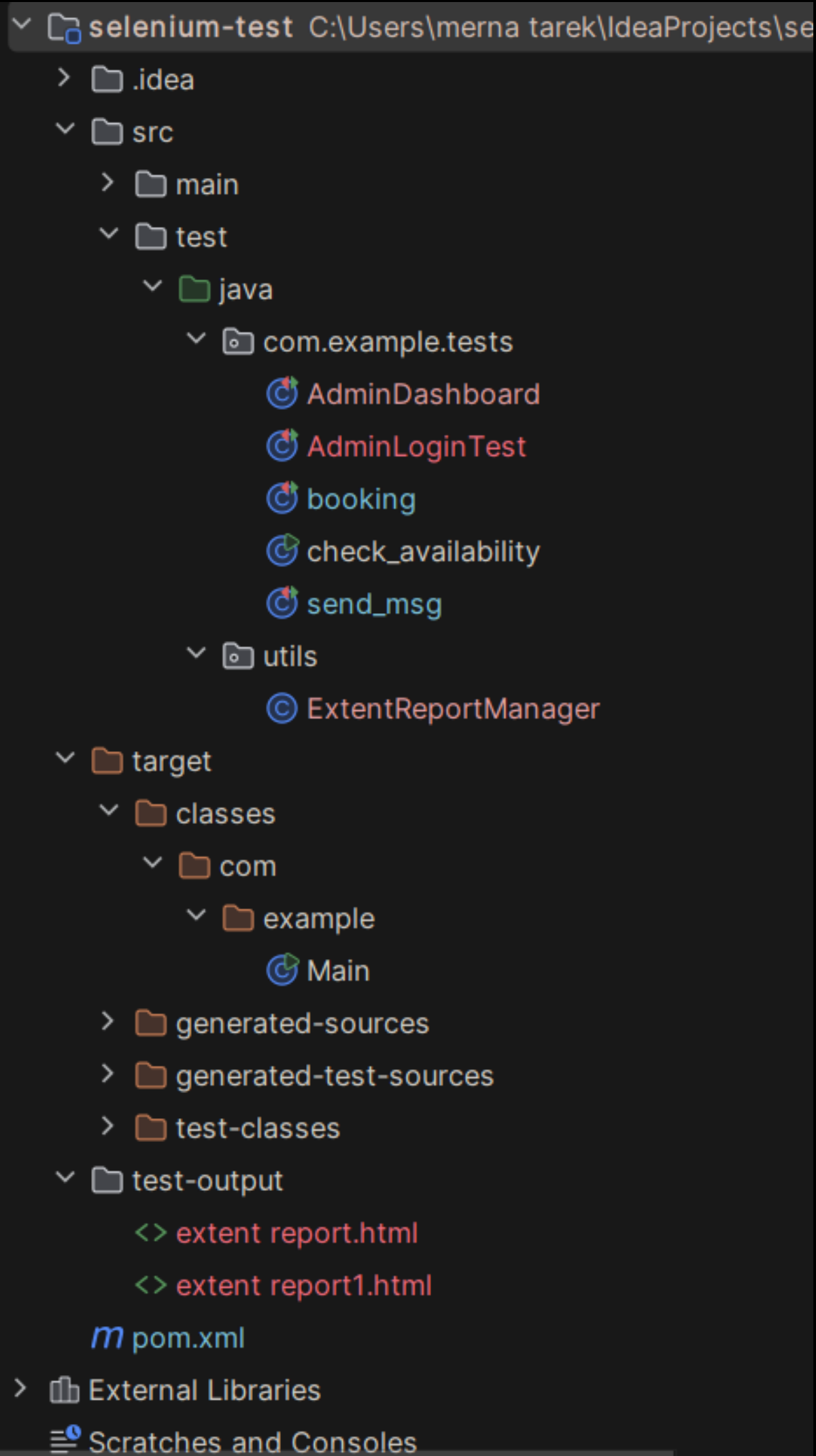
#### 3- Cancel a Booking

- TC20: Verify Successful Cancellation of a Booking
- TC21: Verify Booking is Removed After Cancellation
- TC22: Verify Cancellation with Invalid Booking ID
- TC23: Verify Cancellation of Already Deleted Booking
- TC24: Verify Cancellation Without Auth Token
- TC25: Verify Cancellation with Invalid Auth Token

#### 4- Check Room Availability

- TC26: Verify Check Room Availability with Valid Date Range
- TC27: Verify JSON Response Includes Room Type, Availability Status, and Price
- TC28: Verify Error for Missing Checkin Date Parameter
- TC29: Verify Error for Invalid Date Format
- TC30: Verify Error for Missing Checkout Date Parameter
- TC31: Verify Response Time for Availability Check

# 5. Automation Architecture



## 6.3. Snapshots of Manual Testing

### Test Plan :

Test Plan	
Section	Description
Project Name	Restful Booker
Version	v1.0
Prepared By	Merna Tarek / Farida / Maria / Wessam
Date	[23/04/2025]

Constraints :

[Restful Booker](#)

Manual Testing Scenarios :

[Restful Booker Test Scenarios \( Manual Testing \)](#)

( Test Cases / Test Cases Execution / Bug Report / Traceability Matrix ) Sheets :

[Restful Booker test cases v1 \(1\)](#)

Jira :

<https://mernatarekahmed.atlassian.net/jira/software/projects/RBT/boards/36?atlOrigin=eyJpIjoiZDI4ZmQ3ZWRRmMjlkNGI5Yjk2ZTU1YTlxZjBmNjlhNWUiLCJwIjoiajI9>

GitHub :

<https://github.com/mernatarek1/Restful-Booking-Demo-Project.git>

## 6.3. Snapshots of Manual Testing

### Test Cases :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Requirement ID	TC_ID	TC Title/ Objective	Test Steps	Test Data	Expected Result	Status	Attachment	Environment						
1	Req_AdminLogin_001	Req_AdminLogin_001	Valid Login and Dashboard Access ( by valid Username and valid Password )	1.Navigate to Admin login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter the corresponding valid password in the "Password" field. 4.Click the "Login" button.	Username : admin Password : password123	The admin is successfully logged in and redirected to the admin dashboard.	Approved		Windows 11 Chrome : Version 135.0.7049.96 (Official Build) (64-bit)						
2		Req_AdminLogin_002	Valid Login and Booking List Visibility. ( by valid Username and valid Password )	1.Navigate to login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter the corresponding valid password in the "Password" field. 4.Click the "Login" button. 5.On the admin dashboard, navigate to the "Rooms" section.	Username : admin Password : password123	The admin can see a list of all hotel rooms booked.	Approved		Windows 11 Chrome : Version 135.0.7049.96 (Official Build) (64-bit)						
3		Req_AdminLogin_003	Invalid Username	1.Navigate to login page ( By Click on Admin ) . 2.Enter an invalid username in the "Username" field. 3.Enter a valid password in the "Password" field. 4.Click the "Login" button.	Username : Admin Password : password123	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.	Approved		Windows 11 Chrome : Version 135.0.7049.96 (Official Build) (64-bit)						
4		Req_AdminLogin_004	Invalid Password	1.Navigate to login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter an invalid password in the "Password" field. 4.Click the "Login" button.	Username : admin Password : password1234	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.	Approved		Windows 11 Chrome : Version 135.0.7049.96 (Official Build) (64-bit)						
5		Req_AdminLogin_005	Invalid Username and Password	1.Navigate to login page ( By Click on Admin ) . 2.Enter a invalid username in the "Username" field. 3.Enter an invalid password in the "Password" field. 4.Click the "Login" button.	Username : Admin Password : password1234	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.	Approved		Windows 11 Chrome : Version 135.0.7049.96 (Official Build) (64-bit)						
6		Req_AdminLogin_006	Empty Username Field	1.Navigate to login page ( By Click on Admin ) . 2.Leave the "Username" field empty. 3.Enter the valid password in the "Password" field. 4.Click the "Login" button. 5.On the admin dashboard, navigate to the "Rooms" section .	Username : Password : password123	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.	Approved		Windows 11 Chrome : Version 135.0.7049.96 (Official Build) (64-bit)						

< >

Test Cases

Test Execution Report

Bug Report

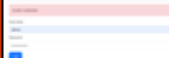

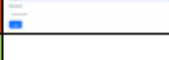





Traceability matrix

+










## 6.3. Snapshots of Manual Testing

# Test Execution Report :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1	Requirement ID	TC_ID	TC Title/ Objective	Test Steps	Test Data	Expected Result	Actual Result	Status	Execution Status	Attachment	Environment	Tester Name										
2	Req_AdminLogin_001	Req_AdminLogin_001	Valid Login and Dashboard Access ( by valid Username and valid Password )	1.Navigate to Admin login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter the corresponding valid password in the "Password" field. 4.Click the "Login" button.	Username : admin Password : password123	The admin is successfully logged in and redirected to the admin dashboard.	The admin is not logged in but sometimes logged with trial  The Error Message : invalid credentials	Approved	Fail		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
3		Req_AdminLogin_002	Valid Login and Booking List Visibility. ( by valid Username and valid Password )	1.Navigate to login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter the corresponding valid password in the "Password" field. 4.Click the "Login" button. 5.On the admin dashboard, navigate to the "Rooms" section.	Username : admin Password : password123	The admin can see a list of all hotel rooms booked.	The admin is not logged in but sometimes logged with trial  The Error Message : invalid credentials	Approved	Fail		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
4		Req_AdminLogin_003	Invalid Username	1.Navigate to login page ( By Click on Admin ) . 2.Enter an invalid username in the "Username" field. 3.Enter a valid password in the "Password" field. 4.Click the "Login" button.	Username : Admin Password : password123	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	Approved	Pass		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
5		Req_AdminLogin_004	Invalid Password	1.Navigate to login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter an invalid password in the "Password" field. 4.Click the "Login" button.	Username : admin Password : password1234	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	Approved	Pass		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
6		Req_AdminLogin_005	Invalid Username and Password	1.Navigate to login page ( By Click on Admin ) . 2.Enter an invalid username in the "Username" field. 3.Enter an invalid password in the "Password" field. 4.Click the "Login" button.	Username : Admin Password : password1234	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	Approved	Pass		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
7		Req_AdminLogin_006	Empty Username Field	1.Navigate to login page ( By Click on Admin ) . 2.Leave the "Username" field empty. 3.Enter the valid password in the "Password" field. 4.Click the "Login" button. 5.On the admin dashboard, navigate to the "Rooms" section .	Username : Password : password123	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	Approved	Pass		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
8		Req_AdminLogin_007	Leave the "Password" field empty.	1.Navigate to login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Leave the "Password" field empty. 4.Click the "Login" button.	Username : admin Password :	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	Approved	Pass		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
9		Req_AdminLogin_008	Empty Username and Password Fields	1.Navigate to login page By Click on Admin . 2.Leave both the "Username" and "Password" fields empty. 3.Click the "Login" button.	Username : Password :	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	An appropriate error message is displayed to the admin, indicating that the login failed due to (invalid credentials) . The admin remains on the login page.  The Error Message : invalid credentials	Approved	Pass		Windows 11 Chrome : Version 135.0.7043.36 (Official Build) (64-bit)	Merna Tarek										
10	Req_BookingForm_002	TC_BookingForm_001	Valid Data Submission	1.Navigate to the booking form ( By Click on Contact ) . 2.Enter a valid Name (alphabetic, <= 50 characters) 3.Enter valid Email (format: name@domain.com)	Name : Merna Tarek Email : test@gmail.com	The booking form is submitted successfully, and the user receives confirmation ( success message )	The booking form is not submitted , and the user receives ( error message )															
<div><div><div>&lt;</div><div>&gt;</div></div><div>Test Cases</div><div>Test Execution Report</div><div>Bug Report</div><div>Traceability matrix</div><div>+</div><div></div><div></div></div>																						

## 6.3. Snapshots of Manual Testing

### Bug Report :

Requirement ID	TC_ID	Defect_ID	Defect Title/Issue	Pre-condition	Test Steps	Test Data	Expected Result	Actual Result	Attachment	Priority	Probability	Severity	Categorization	Defect Status
Req_AdminLogin_001	Req_AdminLogin_001	Bug_AdminLogin_001	Admin Login Failed - Incorrect Credentials Message	User is on the Home page.	1.Navigate to login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter the corresponding valid password in the "Password" field. 4.Click the "Login" button. 5.On the admin dashboard, navigate to the "Rooms" section.	Username : admin Password : password123	The admin is successfully logged in and redirected to the admin dashboard.	The admin is not logged in but sometimes logged with trial		High	High	Major	Functional	Open
	Req_AdminLogin_002	Bug_AdminLogin_002	Admin Login Failed - Incorrect Credentials Message	User is on the Home page.	1.Navigate to login page ( By Click on Admin ) . 2.Enter a valid username in the "Username" field. 3.Enter the corresponding valid password in the "Password" field. 4.Click the "Login" button. 5.On the admin dashboard, navigate to the "Rooms" section.	Username : admin Password : password123	The admin can see a list of all hotel rooms booked.	The admin is not logged in but sometimes logged with trial		High	High	Major	Functional	Open
Req_BookingForm_002	TC_BookingForm_001	Bug_BookingForm_001	Form Submission Failed	User is on the Home page.	1.Navigate to the booking form ( By Click on Contact ) . 2.Enter a valid Name (alphabetic, <= 50 characters) 3.Enter a valid Email Format 4.Enter a valid Phone Format 5.Enter an optional Subject (alphanumeric) 6.Enter an optional Message (up to 250 characters) 7.Click the "Submit" button.	Name : Merna Tarek Email : test@example.com Phone : 0123456789 Subject : This is the subject Message : This is the message	The booking form is submitted successfully, and the user receives confirmation ( success message ) . *Booking appears in the booking list (if admin).	The booking form is not submitted , and the user receives ( error message ) .  <b>The Error Message :</b> 1- Phone must be between 11 and 21 characters. (The Expected From Constraints : Phone 10-15 digits)  2- Message must be between 20 and 2000 characters. (The Expected From Constraints : Message is optional)		High	High	Critical	Functional	Open
	TC_BookingForm_002	Bug_BookingForm_002	Form Submission Failed	User is on the Home page.	1.Navigate to the booking form ( By Click on Contact ) . 2.Enter valid Name 3.Enter valid Email format 4.Enter valid Phone format 5.Leave the Subject empty 6.Leave the Message empty. 7.Click the "Submit" button.	Name : Merna Tarek Email : test@example.com Phone : 0123456789 Subject : Message :	The booking form is submitted successfully, and the user receives confirmation ( success message ) . *Booking appears in the booking list (if admin).	The booking form is not submitted , and the user receives ( error message ) .  <b>The Error Message :</b> 1- Subject may not be blank 2- Subject must be between 5 and 100 characters. (The Expected From Constraints : Subject is optional) 3- Message may not be blank 4- Message must be between 20 and 2000 characters. (The Expected From Constraints : Message is optional)		High	High	Critical	Functional	Open
	TC_BookingForm_003	Bug_BookingForm_003	Submission Succeeded With Wrong Field Constraints	User is on the Home page.	1.Navigate to the booking form ( By Click on Contact ) . 2.Enter a invalid Name containing numeric or special characters 3.Enter a valid Email format 4.Enter a valid Phone format 5.Enter an optional Subject 6.Enter an optional Message 7.Click the "Submit" button.	Name : Merna123 Email : test@example.com Phone : 0123456789 Subject : This is the subject Message : This is the message This is the message	An error message is displayed indicating that the "Name" must contain only alphabetic characters. The form is not submitted	The booking form is submitted successfully, and the user receives confirmation ( success message ) . *Booking appears in the booking list (if admin).		Medium	Medium	Major	Functional	Open
	TC_BookingForm_009	Bug_BookingForm_004	Submission Succeeded With Wrong Field Constraints	User is on the Home page.	1.Navigate to the booking form ( By Click on Contact ) . 2.Enter a invalid Name with more than 50 alphabetic characters. 3.Enter a valid Email format 4.Enter a valid Phone format 5.Enter an optional Subject 6.Enter an optional Message 7.Click the "Submit" button.	Name : MernaMernaMernaMernaMernaMernaMernaMernaMernaM Email : test@example.com Phone : 0123456789 Subject : This is the subject Message : This is the message This is the message	An error message is displayed indicating that the " Name" cannot exceed 50 characters. The form is not submitted.	The booking form is submitted successfully, and the user receives confirmation ( success message ) . *Booking appears in the booking list (if admin).		Medium	Medium	Major	Functional	Open
	TC_BookingForm_013	Bug_BookingForm_005	Submission Succeeded With Wrong Field Constraints	User is on the Home page.	1.Navigate to the booking form ( By Click on Contact ) . 2.Enter a valid Name 3.Enter a valid Email format 4.Enter a invalid Phone format 5.Enter an optional Subject 6.Enter an optional Message 7.Click the "Submit" button.	Name : Merna Email : test@example.com Phone : 0123456789M Subject : This is the subject Message : This is the message This is the message	An error message is displayed indicating that the "Phone" number must be numeric. The form is not submitted.	The booking form is submitted successfully, and the user receives confirmation ( success message ) . *Booking appears in the booking list (if admin).		Medium	Medium	Major	Functional	Open

## 6.3. Snapshots of Manual Testing

### Traceability matrix :

1	A	B	C	D	E
	Requirement ID	Req_AdminLogin_001	Req_BookingForm_002	Req_BookingRoom_003	
2	Requirement Description	As an admin, I want to log in securely so that I can manage hotel bookings and view customer messages.	As a guest, I want to fill out the contact form .	As a guest, I want to book a room by filling out a form so that I can reserve accommodation online.	
3	Test Case ID(s)	Req_AdminLogin_001 Req_AdminLogin_002 Req_AdminLogin_003 Req_AdminLogin_004 Req_AdminLogin_005 Req_AdminLogin_006 Req_AdminLogin_007 Req_AdminLogin_008	TC_BookingForm_001 TC_BookingForm_002 TC_BookingForm_003 TC_BookingForm_004 TC_BookingForm_005 TC_BookingForm_006 TC_BookingForm_007 TC_BookingForm_008 TC_BookingForm_009 TC_BookingForm_010 TC_BookingForm_011 TC_BookingForm_012 TC_BookingForm_013 TC_BookingForm_014 TC_BookingForm_015 TC_BookingForm_016	TC_BookingRoom_001 TC_BookingRoom_002 TC_BookingRoom_003 TC_BookingRoom_004 TC_BookingRoom_005 TC_BookingRoom_006 TC_BookingRoom_007 TC_BookingRoom_008 TC_BookingRoom_009 TC_BookingRoom_010 TC_BookingRoom_011 TC_BookingRoom_012 TC_BookingRoom_013 TC_BookingRoom_014 TC_BookingRoom_015 TC_BookingRoom_016 TC_BookingRoom_017 TC_BookingRoom_018	
		Req_AdminLogin_001: Fail -> Bug_AdminLogin_001 Req_AdminLogin_002: Fail -> Bug_AdminLogin_002 Req_AdminLogin_003: Pass Req_AdminLogin_004: Pass Req_AdminLogin_005: Pass Req_AdminLogin_006: Pass Req_AdminLogin_007: Pass Req_AdminLogin_008: Pass	TC_BookingForm_001: Fail -> Bug_BookingForm_001 TC_BookingForm_002: Fail -> Bug_BookingForm_002 TC_BookingForm_003: Pass TC_BookingForm_004: Pass TC_BookingForm_005: Deprecated TC_BookingForm_006: Pass TC_BookingForm_007: Pass TC_BookingForm_008: Fail -> Bug_BookingForm_003 TC_BookingForm_009: Fail -> Bug_BookingForm_004 TC_BookingForm_010: Deprecated TC_BookingForm_011: Deprecated TC_BookingForm_012: Pass TC_BookingForm_013: Fail -> Bug_BookingForm_005 TC_BookingForm_014: Pass TC_BookingForm_015: Pass TC_BookingForm_016: Fail -> Bug_BookingForm_006	TC_BookingRoom_001: Fail -> Bug_BookingRoom_001 TC_BookingRoom_002: Fail -> Bug_BookingRoom_002 TC_BookingRoom_003: Fail -> Bug_BookingRoom_003 TC_BookingRoom_004: Fail -> Bug_BookingRoom_004 TC_BookingRoom_005: Fail -> Bug_BookingRoom_005 TC_BookingRoom_006: Fail -> Bug_BookingRoom_006 TC_BookingRoom_007: Pass TC_BookingRoom_008: Pass TC_BookingRoom_009: Fail -> Bug_BookingRoom_007 TC_BookingRoom_010: Fail -> Bug_BookingRoom_008 TC_BookingRoom_011: Fail -> Bug_BookingRoom_009 TC_BookingRoom_012: Fail -> Bug_BookingRoom_010 TC_BookingRoom_013: Pass TC_BookingRoom_014: Fail -> Bug_BookingRoom_011 TC_BookingRoom_015: Fail -> Bug_BookingRoom_012 TC_BookingRoom_016: Fail -> Bug_BookingRoom_013 TC_BookingRoom_017: Fail -> Bug_BookingRoom_014 TC_BookingRoom_018: Deprecated	
4	Requirement Test Status	Fail	Fail	Fail	
5					
6					

< >

Test Cases

Test Execution Report

Bug Report

Traceability matrix

+

:

## 4. Test Scenarios

### •UI :

#### •1. Login Page (Admin Panel)

- TC01: Verify admin login with valid credentials.
- TC02: Verify login with invalid username/password.
- TC03: Attempt login with blank fields → verify validation message.
- TC04: Validate successful login redirects to admin dashboard.

#### 2. Contact Form (Front Page)

- TC05: Submit valid contact form (name, email, phone, subject, description) → verify success message.
- TC06: Submit form with missing fields → verify validation errors.
- TC07: Submit invalid email format → verify error.

#### 3. Book a Room

- TC08: Select room → enter valid booking details (dates, name, email) → submit → verify confirmation.
- TC09: Attempt booking with overlapping dates → check for rejection or proper handling.
- TC10: Book without filling required fields → verify validation.
- TC11: Book with invalid date range (end date before start) → verify error.

#### 4. Room Management (Admin Panel)

- TC12: After login, add a new room (room type, price, accessibility, image) → verify room is listed.
- TC13: Edit an existing room → update info → verify changes persist.
- TC14: Delete a room → confirm it no longer appears in the list.
- TC15: Add room with missing fields → verify validation.



# 6.1. API Demo

The screenshot displays the Postman API client interface. The top navigation bar includes links for Home, Workspaces, and API Network, along with a search bar and user controls. The left sidebar shows the 'My Workspace' section with a tree view of collections and environments. The main panel is divided into two sections: 'Run Sequence' and 'Run configuration'.

**Run Sequence:** A list of 13 test cases (TC1 to TC13) is shown, each with a checkbox and a status indicator. The tests are as follows:

Index	Method	Test Case Description
1	GET	TC1: Verify Successful GET /booking (List Bookings)
2	GET	TC2: Verify Successful GET /booking/:id (Get Booking Details)
3	POST	TC3: Verify Successful POST /booking (Create Booking)
4	POST	TC5: Verify Successful POST /auth (Authentication)
5	PUT	TC6: Verify Successful PUT /booking/:id (Update Booking)
6	PATCH	TC7: Verify Successful PATCH /booking/:id (Partial Update)
7	POST	TC8: Verify POST /booking with Missing Required Field
8	POST	TC9: Verify POST /booking with Invalid Date Format
9	GET	TC10: Verify GET /booking/:id with Non-Existent ID
10	POST	TC11: Verify POST /booking with Empty Body
11	POST	TC12: Verify POST /auth with Invalid Credentials
12	PUT	TC13: Verify PUT /booking/:id without Token

**Run configuration:** This section allows users to configure how the collection is run. It includes options for 'Run manually', 'Schedule runs', and 'Automate runs via CLI'. The 'Iterations' field is set to 1, and the 'Delay' field is set to 0 ms. A 'Test data file' field is also present, with a note that only JSON and CSV files are accepted. A 'Run RB' button is visible at the bottom of the configuration section.

The bottom status bar shows the current state of the application, including 'Online', 'Find and replace', 'Console', and various system icons.



# Admin Login :

## 6.2. UI Demo

The screenshot displays an IDE interface with a project named 'selenium-test' and a file named 'AdminLoginTest.java' open. The code in 'AdminLoginTest.java' includes imports for Selenium WebDriver, TestNG, and ExtentReports, and defines a public class 'AdminLoginTest' with a private static final String 'WEB SITE URL'.

The 'Run' tab shows a failed test suite 'Default Suite' with 2 failed tests and 2 passed tests. The error message indicates that no SLF4J providers were found, and the defaulting to no-operation (NOP) logger implementation was used.

The 'Project' view on the left shows the following structure:

- AdminDashboard
- AdminLoginTest
- booking
- check\_availability
- send\_msg
- utils
  - ExtentReportManager
- target
  - classes
    - com
      - example
        - Main
  - generated-sources
  - generated-test-sources
  - test-classes
- test-output
  - extent report.html
  - extent report1.html
- pom.xml
- External Libraries
- Scratches and Consoles

```
1 import io.github.donigarcia.wdm.WebDriverManager;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6 import org.openqa.selenium.chrome.ChromeDriver;
7 import org.openqa.selenium.support.ui.ExpectedConditions;
8 import org.openqa.selenium.support.ui.WebDriverWait;
9
10 import org.testng.Assert;
11 import org.testng.ITestContext;
12 import org.testng.annotations.AfterClass;
13 import org.testng.annotations.AfterMethod;
14 import org.testng.annotations.BeforeClass;
15 import org.testng.annotations.BeforeMethod;
16 import org.testng.annotations.Test;
17
18 import com.aventstack.extentreports.ExtentReports;
19 import com.aventstack.extentreports.ExtentTest;
20 import utils.ExtentReportManager;
21
22 import java.time.Duration;
23
24 public class AdminLoginTest {
25     private static final String WEB SITE URL = "https://automationintesting.online"; // 1 usage
```

Run booking x

Default Suite 1 min 4 sec

- selenium-test 1 min 4 sec
  - booking 1 min 4 sec
    - testValidBooking 12 sec 921 ms

Tests failed: 2, passed: 2 of 4 tests – 1 min 4 sec

"C:\Program Files\Java\jdk-23\bin\java.exe" ...

SLF4J(W): No SLF4J providers were found.

SLF4J(W): Defaulting to no-operation (NOP) logger implementation

selenium-test > src > test > java > com > example > tests > AdminLoginTest

239:2 CRLF UTF-8 4 spaces





# Booking Room :

## 6.2. UI Demo

The screenshot displays an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `main`, `test`, `java`, `com.example.tests`, `utils`, `target`, `classes`, `com`, `example`, `generated-sources`, and `generated-test-sources`. The `booking` class is highlighted.
- Code Editor:** Displays the `booking.java` file with the following code:

```
26 public class booking { merna *
244 private void selectrange(String startDay, String endDay) throws InterruptedException { 5 usages merna *
342     unavailableEvents.stream().anyMatch(u -> u.getLocation().equals(e.getLocation())) {
343         test.warning( details: "Selection overlaps with unavailable dates. Clearing and retrying.");
344         actions.moveToElement(startButton).click().perform(); // Attempt to deselect
345         selectrange(startDay, endDay); // Retry with the same parameters
346     }
347 }
348 } catch (Exception e) {
349     test.error( details: "Failed to select date range: " + e.getMessage());
350     System.out.println("Failed to select date range: " + e.getMessage());
351     throw e; // Re-throw to fail the test
352 }
353
354 private void navigateToMonth(String targetMonthYear) throws InterruptedException { 1 usage new *
355     test.info( details: "Navigating to month: " + targetMonthYear);
356     try {
357
```
- Run Console:** Shows the test results and error messages:

```
1 sec 942 ms X Stopped. Tests passed: 0 of 4 tests - 1 sec 942 ms
1 sec 942 ms "C:\Program Files\Java\jdk-23\bin\java.exe" ...
1 sec 942 ms SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Extent report initialized.
May 10, 2025 3:00:34 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 136
```



# Booking Form :

## 6.2. UI Demo

The screenshot displays an IDE interface with a project named 'selenium-test' and a file named 'send\_msg.java' open. The code defines a test class with various static constants for URLs, timeouts, and XPath locators. The test run at the bottom shows a failure in the 'testValidAdminLogin' method.

**Project Structure:**

- AdminDashboard
- AdminLoginTest
- booking
- check\_availability
- send\_msg
- utils
- ExtentReportManager
- target
  - classes
    - com
      - example
        - Main
  - generated-sources
  - generated-test-sources
  - test-classes
- test-output
  - extent report.html
  - extent report1.html
  - extent report2.html
- pom.xml
- External Libraries
- Scratches and Consoles

**Code in send\_msg.java:**

```
1 package com.example.tests;
2
3 import ...
4
25 public class send_msg {
26     *
27     private static final String WEBSITE_URL = "https://automationintesting.online"; 1 usage
28     private static final int WAIT_TIMEOUT_SECONDS = 60; 1 usage
29     private static final int SHORT_WAIT_SECONDS = 5; 1 usage
30     private static final int FIELD_WAIT_SECONDS = 20; 1 usage
31     private static final int TYPING_SPEED_MS = 50; 1 usage
32     private static final int POLL_INTERVAL_MS = 50; 3 usages
33     private static final int POST_CLICK_WAIT_MS = 1000; // Add this constant for wait after click 2 usages
34
35     // Contact Form Locators
36     private static final String CONTACT_LINK_XPATH = "//a[contains(text(), 'Contact')]"; 1 usage
37     private static final String CONTACT_SECTION_XPATH = "//section[@id='contact']"; 1 usage
38     private static final String NAME_FIELD_XPATH = "//input[@data-testid='ContactName']"; 4 usages
39     private static final String EMAIL_FIELD_XPATH = "//input[@data-testid='ContactEmail']"; 3 usages
40     private static final String PHONE_FIELD_XPATH = "//input[@data-testid='ContactPhone']"; 2 usages
41     private static final String SUBJECT_FIELD_XPATH = "//input[@data-testid='ContactSubject']"; 2 usages
42     private static final String MESSAGE_FIELD_XPATH = "//textarea[@data-testid='ContactDescription']"; 2 usages
43     private static final String SUBMIT_BUTTON_XPATH = "//button[contains(@class, 'btn-primary') and text()='Submit']";
```

**Test Run Results:**

Test Suite	Duration	Status
Default Suite	43 sec 17 ms	Failed
selenium-test	43 sec 17 ms	Failed
AdminLoginTest	43 sec 17 ms	Failed
testValidAdminLogin	6 sec 393 ms	Failed

**Test Failure Details:**

Tests failed: 2, passed: 2 of 4 tests – 43 sec 17 ms

"C:\Program Files\Java\jdk-23\bin\java.exe" ...

SLF4J(W): No SLF4J providers were found.

SLF4J(W): Defaulting to no-operation (NOP) logger implementation

Externally added files can be added to Git  
View Files Always Add Don't Ask Again

## 7. Results & Reports

- **Manual Testing:**

- Total test cases designed: 39 .
- **Passed:** 15; **Failed:** 24 .

- Documented by ( Excel spreadsheets ).

•-----

- **UI Testing:**

- Total test cases designed: 15.
- **Passed:** 9; **Failed:** 6 .

- Documented by ( ExtentReports ).

•-----

- **API Testing:**

- Total test cases designed: 32 .
- **Passed:** 21; **Failed:** 11 .

- Documented by ( **htmlextra** ).



## 7. Results & Reports (More Detailed)

Manual Testing	Admin Login	Total : 8 Pass : 6 Fail : 2
	Booking Form	Total : 16 Pass : 7 Fail : 9
	Booking Room	Total : 18 Pass : 6 Fail : 12

API Testing	Booking Room	Total : 13 Pass : 10 Fail : 3
	Admin Login	Total : 6 Pass : 4 Fail : 2
	Cancel a Booking	Total : 6 Pass : 5 Fail : 1
	Check Room Availability	Total : 8 Pass : 8 Fail : 0

## 7. Results & Reports (More Detailed)

UI Testing	Booking Room	Total : 4 Pass : 2 Fail : 2
	Booking Form	Total : 3 Pass : 2 Fail : 1
	Admin Login	Total : 4 Pass : 2 Fail : 2
	Admin Dashboard	Total : 4 Pass : 3 Fail : 1



## 8. Defect Management

- **Logging:**

All defects were tracked manually using an **Excel-based defect log**, categorized by **severity** (High, Medium, Low), and updated throughout the testing cycle. Each entry included fields such as Defect ID, Summary, Steps to Reproduce, Expected vs Actual Results, Severity, and Status.

- **Examples:**

- *Login Failure (High severity):* Admin login logged with invalid credentials.

- **Traceability:** Each defect in the Excel sheet was **mapped to corresponding test case IDs** to ensure traceability. This helped us confirm bug fixes during re-testing by re-executing the linked test cases.

## 9. Challenges Faced

- **Dynamic Element Locators:** Some UI elements (e.g. date pickers) had dynamic IDs or loaded asynchronously. We resolved this by using robust locators (relative XPaths) and explicit waits (WebDriverWait) to wait for elements to appear. Explicit waits help avoid flaky failures by pausing until the element is present.
- **API Response Delays:** Occasionally the API would respond slowly, causing timeouts. We added retry logic or increased timeout thresholds in Postman/Newman to handle transient delays.
- **Synchronization Issues:** Ensuring tests ran at the right time (waiting for page loads or animations) required careful use of waits. For example, after clicking “Book”,
  - the confirmation appears after a short delay; using WebDriverWait for the confirmation text solved intermittent failures.
- **Data Management:** Because Restful-Booker resets its data every 10 minutes, some tests needed to handle test data cleanup (e.g. deleting a created booking).
  - We wrote teardown steps in automation to clean up test data.

## 10. Conclusion & Learnings

- The project provided experience in end-to-end test lifecycle, from requirements to reporting. We reinforced best practices for both manual and automated testing.
- We gained hands-on skills with tools: Selenium + TestNG for UI automation, Postman for APIs, and Maven/IntelliJ for project setup. We saw the benefits of Page Object Model (better maintainability) and data-driven testing (broader coverage with less code)
- Effective collaboration was practiced by linking test cases and defects, using JIRA for communication. Clear reporting (via ExtentReports and test case summaries) helped the team stay aligned on quality goals.
- Overall, this effort deepened our understanding of how functional and non-functional tests complement each other: verifying core booking workflows (functional) and ensuring a smooth user experience (non-functional)

Any  
Question ?

Thank You