



JUICE SHOP TESTING DOCUMENTATION

Version: 1.0

Date: 23-02-2025

Team: Automation

Prepared By:

Jannatul Ferdous Salma

Prepared For:

Mahbub Hossain (Team Leader)

Table of Contents

Chapter 1. Manual Testing

1.1 Login Page Testing

1.1.1.Screenshot Reference in Excel Sheet.....	4
1.1.2.Test Execution Results for Login Page.....	6
1.1.3.Breakdown of Severity and Priority.....	6
1.1.4.Key Observations from Login Page Testing.....	7

1.2 After Login Page - All Products Testing

1.2.1.Screenshot Reference in Excel Sheet.....	9
1.2.2.Test Execution Results for All Products Testing.....	13
1.2.3.Breakdown of Severity and Priority.....	13
1.2.4.Key Observations from Login Page Testing.....	14

Chapter 2. Introduction of API Testing

2.1 Overview.....	19
2.2 Base URL.....	19
2.3 Authorization.....	19
2.5 API Test Cases Table.....	20

Chapter 3. API Testing in Postman

3.1 User Authentication.....	21
3.2 Fetch Product List.....	23
3.3 Create User Address.....	26
3.4 Fetch User Address Details.....	28

3.5 Retrieve User Wallet Balance.....	29
3.6 Fetch Delivery Info.....	31
3.7 Fetch Saved Cards.....	32
3.8 Checkout Order.....	34
3.9 Fetch Basket Details.....	35
3.10 Update Product Review.....	37
3.11 Fetch Product Reviews.....	38
3.12 Retrieve Shopping Cart Info.....	40

Chapter 4. API Automation Testing with Postman & Newman

4.1 Exporting Postman Collection.....	42
4.2 Installing Newman.....	43
4.3 Running API Tests using Newman.....	44
4.4 Generate an HTML Report.....	45
4.5: Open the HTML Report.....	46

Chapter 5. Conclusion..... 72

Chapter 1: Manual Testing

1.1 Login Page Testing

The manual testing results for the login functionality of the OWASP Juice Shop application. It includes multiple test cases covering various aspects of login functionality, including UI validation, authentication, security, and performance.

1.1.1 Screenshot Reference in Excel Sheet

To support these test results, refer to the attached screenshots in the test case document. The screenshots highlight the execution results for login-related test cases, including successful and failed scenarios.

Req ID	Requirement Description	TC ID	Test Case Description	Category	Automation TC ID	Test Steps	Test Data	Expected Result	Actual Result	TEST CASE STATUS		Severity	Priority	QA Remarks
										K	L			
7	RQ001 Login page should load properly	TC001	Verify if the login page loads properly	Functional		1. Open the login page URL. 2. Check if the page loads without errors.	URL: juice-shop.herokuapp.com/login	Page should load successfully	As expected, the page loaded successfully.	PASS		High	High	
8	RQ002 Application should be available and load without errors	TC002	Verify application availability	Functional		1. Open the application URL (juice-shop.herokuapp.com) 2. Check if the page loads successfully without errors.		N/A	The application should load without any errors.	FAIL		Critical	High	Application error encountered, needs investigation
9	RQ003 UI elements should be present	TC003	Verify UI elements (Email, Password fields, Login button, Google login)	Functional		1. Check if Email and Password fields are present. 2. Check if the "Log in with Google" button is present.		N/A	All elements should be present and aligned.	PASS		Medium	Medium	
10	RQ004 User should be able to log in with Google	TC004	Verify login with Google authentication (Valid Account)	Functional		1. Click "Log in with Google". 2. Enter valid credentials. 3. Authenticate.	Valid Google account	User should log in successfully.	User should log in and was redirected to the homepage/dashboard.	PASS		Official	High	
11	RQ005 Handle inactive Google account	TC005	Verify login with an inactive Google account	Functional		1. Click "Log in with Google". 2. Enter an inactive account.	Inactive Google account	User should see an error message.	Error message displayed: Your Google account is inactive. Please contact your administrator or try a different account.	PASS		High	High	
12	RQ006 Support Two-Factor Authentication	TC006	Verify login with Google 2FA enabled	Functional		1. Click "Log in with Google". 2. Enter valid credentials. 3. Provide 2FA code.	Google 2FA enabled account	User should log in after authentication.	User should log in after entering the correct code.	PASS		High	High	
13	RQ007 Login button should be disabled when fields are empty	TC007	Verify login button is disabled when fields are empty	Functional		1. Leave Email & Password blank. 2. Click login button.	Empty fields	Login button should be disabled.	Login button is disabled and validation messages are shown when Email and Password fields are empty.	PASS		Medium	Medium	
14	RQ008 Verify User Registration Functionality	TC008	Verify the user can successfully register using a valid email, password, and security question	Functional		1. Navigate to the User Registration page. 2. Enter a valid email address (e.g., user123@gmail.com). 3. Enter a strong password (at least 8 characters). 4. Re-enter the same password. 5. Enter a security question. 6. Provide an answer. 7. Click "Register".	Email: user123@gmail.com Password: user123 Re-enter Password: user123 Security Question: Name of your favorite pet? Answer: Cat	The account should be created successfully.	Account is successfully created.	PASS		High	High	

The first image focuses on **functional login-related test cases**, covering scenarios such as invalid credentials, inactive accounts, and session persistence. It captures system error messages and validation failures, emphasizing points where test cases did not pass. A critical failure is **TC002 - Verify Application Availability (FAIL)** due to an error message: "An error occurred in the application and your page could not be served." This issue affects system availability and requires further investigation.

15	RQ009	Verify User Registration Functionality	TC009	Verify that the user can successfully register with a valid email, password, and security question	Functional	1. Navigate to the User Registration page 2. Enter a valid email address (e.g., user123@gmail.com) 3. Enter a strong password (6-40 characters) 4. Re-enter the same password. 5. Select a security question. 6. Provide an answer. 7. Click "Register".	Email: user123@gmail.com Password: user123 Re-enter Password: user123 Security Question: "Name of your favorite pet?" Answer: "Cat"	The system should be email format is incorrect (.cpm instead of .com)	Account is Successfully created	FAIL	High	High	Add remarks like UI issues, validation errors, or improvements
16	RQ010	User authentication	TC010	Verify login with valid Email & invalid password	Functional	1. Open login page 2. Enter valid email 3. Enter invalid password	Valid email:admin123@gmail.com Invalid password: user	Error message: "invalid credentials" should be displayed	Error message displayed: "invalid credentials"	PASS	High	High	
17	RQ011	User authentication	TC011	Verify login with invalid username & invalid password	Functional	1. Open login page 2. Enter invalid email 3. Enter valid password	Invalid email:admin123@gmail.com Valid password: user123@	Error message: "invalid credentials" should be displayed	Error message displayed: "invalid credentials"	PASS	High	High	
18	RQ012	User authentication	TC012	Verify login with invalid username & invalid password	Functional	1. Open login page 2. Enter valid email 3. Enter invalid password	Invalid email:admin123@gmail.com Invalid password: user123@	Error message displayed: "invalid credentials"	Error message displayed: "invalid credentials"	PASS	High	High	
19	RQ013	Maintain login session	TC013	Verify login persistence after session timeout	Functional	1. Log in with Google	Logged-in user	User should stay logged in	User successfully logged in	PASS	Medium	Medium	
20	RQ014	Ensure proper redirection after login	TC014	Verify log-in redirection after successful authentication	Functional	1. Log in with Google	Valid user account	User should be redirected to the homepage/dashboard	User was successfully redirected to the homepage/dashboard	PASS	High	High	

The **second image** provides a broader view of **functional test execution results**, displaying **PASS/FAIL status, severity, and priority** for various test cases. It highlights crucial login functionalities, including user authentication, session maintenance, and redirection. A significant failure is **TC009 - User Registration Failure Due to Invalid Email Format (FAIL)**, where the system accepted an incorrect email format (".".cpm" instead of ".com"). Also **TC010- Invalid email format** This points to a validation issue that needs to be addressed.

Both failed cases fall under the functional category and require fixes to ensure system reliability.

Test Case ID	Test Case Description	Test Case Details	Test Step	Expected Result	Actual Result	Notes/Issues	Impact Level	Priority		
21	RQ015 Handle revoked Google access	TC015 Verify login with revoked Google access	Negative	1. Log in with Google 2. Revoke account access from Google settings 3. Try to log in again	Google account with revoked access	User should see an error message and be prompted to re-authorize the app	Error message: App access was revoked. Please re-authorize	PASS	High	High
22	RQ016 Handle Google service downtime	TC016 Verify login if Google services are down	Negative	1. Simulate Google outage 2. Click "Log in with Google"	N/A	User should see a proper error message	Error message: Application server is down	PASS	High	High
23	RQ017 Handle network issues	TC017 Verify login in network disconnection during login attempt	Negative	1. Turn off internet before login 2. Click "Log in with Google"	Disconnected network	Login should fail with an appropriate error message	Error message: This site can't be reached	PASS	High	High
24	RQ018 Auto logout on password change	TC018 Verify login when Google account password is changed	Negative	1. Log in with Google 2. Change Google password 3. Try to access account	Changed Google password	User should be logged out and required to log in again	User was automatically logged out after password change	PASS	High	High
25	RQ019 Prevent duplicate login requests	TC019 Verify clicking "Log in with Google" multiple times quickly	Negative	1. Click "Log in with Google" multiple times quickly	N/A	System should handle duplicate requests properly	User was automatically logged out after password change	PASS	Medium	Medium
26	RQ020 The system should authenticate valid credentials	TC020 Verify login with manually entered valid email and password	Negative	1. Enter a valid email manually (user123@gmail.com) 2. Enter the correct password manually 3. Click "Login"	Valid email & password	The user should be logged in successfully	The user is successfully logged in	PASS	High	High

The **third image** focuses on **negative test cases**, ensuring the system properly handles error scenarios and unexpected conditions. These test cases verify system behavior under adverse conditions like revoked access, service downtime, network issues, and duplicate login attempts.

All test cases in this category have **passed**, meaning the system correctly displayed error messages and enforced security measures.

Test Case ID	Test Case Description	Test Case Details	Test Step	Expected Result	Actual Result	Notes/Issues	Impact Level	Priority		
27	RQ021 The system should notify users if their password was found in a data breach	TC021 Verify that the system displays a warning if a breached password is used	Security	1. Open the CHWP Juice Shop application in a web browser 2. Navigate to the login page and enter valid credentials 3. Use a compromised password 4. Click on the Login button	Username: admin123@gmail.com Password: user123	A pop-up message or alert should be displayed, recommending the user to change their password	Grocery Password Manager displayed a warning about a breached password	PASS	High	High
28	RQ022 Ensure expired OAuth tokens result in re-authentication	TC022 Verify login when OAuth token expires	Security	1. Log in with Google 2. Wait for token expiration 3. Try accessing protected pages	Expired OAuth token	User should be prompted to re-login after token expiration	User was prompted to re-log in after token expiration	PASS	High	High
29	RQ023 Prevent brute-force login attempts	TC023 Verify brute-force login attempts	Security	1. Attempt multiple failed logins	Invalid login attempts	System should not allow multiple log in attempts	User was temporarily locked out after multiple failed attempts	PASS	Critical	High

The **fourth image** focuses on **security test cases**, ensuring the system enforces proper security measures against threats like **password breaches, expired tokens, and brute-force attacks**. These tests verify whether the system provides appropriate warnings, prompts re-authentication, and prevents unauthorized access attempts.

All security test cases have **passed**, indicating strong protection against potential vulnerabilities.

30	RQ024	Ensure optimal login performance	TC024	Verify login response time under normal load	Performance	1. Attempt login with normal traffic conditions	Normal load	Login should complete in <3 seconds	Login completed within 2.5 seconds	PASS	High	High
31	RQ025	Test login under heavy traffic	TC025	Verify login response time under high traffic load	Performance	1. Simulate 500+ users logging in simultaneously	High load	Login should not slow down significantly	Login response time slightly increased but remained under acceptable limits	PASS	High	High

The **fifth image** focuses on **performance testing**, ensuring the login system functions optimally under different traffic conditions. These tests assess response times and system stability under normal and high-load scenarios. Both performance test cases have **passed**, indicating the system handles login requests efficiently.

32	RQ026	Ensure login works on different browsers	TC026	Verify login on Chrome, Firefox, Edge	Compatibility	1. Open login page on different browsers 2. Attempt login	Chrome, Firefox, Edge	Login should work properly	Login was successful across all tested browsers	PASS	High	High
33	RQ027	Ensure login is mobile-friendly	TC027	Verify login on mobile devices	Compatibility	1. Open login page on mobile devices 2. Attempt login	Android, iOS	Login should work properly	Login worked properly on both Android and iOS devices	PASS	Medium	High
34	RQ028	Ensure UI is responsive	TC028	Verify login page responsiveness	UI/UX	1. Resize browser window 2. Check layout	Different screen sizes	UI should adjust properly	UI elements adjusted smoothly to accommodate without overlapping or distortion	PASS	Medium	High
35	RQ029	Ensure error messages are user-friendly	TC029	Verify login error messages clearly	UI/UX	1. Try logging in with incorrect credentials 2. Check error messages	Invalid credentials	Errors should be easy to understand	Error message displayed: "Invalid email or password. Please try again"	PASS	Medium	Medium
36	RQ030	The system should allow language selection	TC030	Verify that the language selection dropdown works correctly	UI/UX	1. Click on the language dropdown 2. Select a different language (e.g., Spanish, French) 3. Verify that the UI updates accordingly	Multiple language options	The UI should change to the selected language	Language changed successfully	PASS	Medium	Medium

The **sixth image** focuses on **compatibility, UI/UX, and error handling** for the login system. These tests ensure smooth functionality across different devices, browsers, and screen sizes while maintaining a user-friendly experience. All test cases have **passed**, confirming proper system behavior.

1.1.2. Test Execution Results for Login Page:

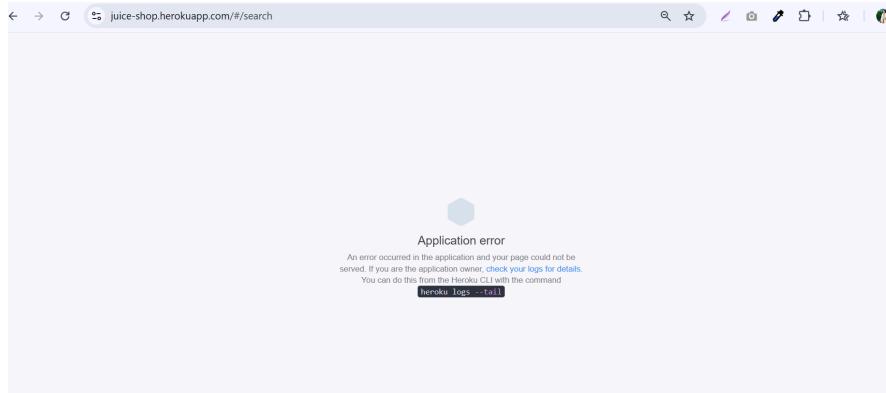
- Total Test Cases: 30
- Passed Test Cases: 28
- Failed Test Cases: 2

1.1.3. Breakdown of Severity and Priority:

- **Severity Breakdown**
 - Critical: 4 (two failed cases fall under this category)
 - High: 18
 - Medium: 8
 - Low: 0
- **Priority Breakdown**
 - High: 24
 - Medium: 6
 - Low: 0

1.1.4.Key Observations from Login Page Testing:

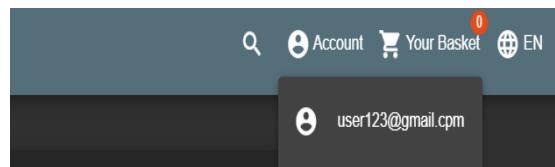
- **Successful Test Cases (PASS - 28 Cases)**
 - The login page loads successfully.
 - UI elements (email, password, and login button) are properly aligned.
 - Users can log in with valid credentials.
 - Google authentication with valid credentials works.
 - Google 2FA authentication functions correctly.
 - The login button remains disabled when required fields are empty.
 - Invalid login attempts show appropriate error messages.
 - Session persistence works properly after a page refresh.
 - Proper redirection occurs after a successful login.
 - Revoked Google account access is handled correctly.
 - The system logs users out when their Google password changes.
 - Security measures such as brute-force prevention and session expiration re-authentication work effectively.
- **Failed Test Cases (FAIL - 2 Cases)**
 - RQ02- Application Availability and Load Errors (Critical Severity)



- Expected behavior: The application should load without errors.
- Actual result: An error occurred, preventing page loading.
- QA Remark: Application error message was displayed, requiring debugging.
- QA Remark: Needs UI validation improvements.
- RQ09 - User Registration with Incorrect Email Format (Critical Severity)

The screenshot shows the 'User Registration' form. The 'Email' field contains 'user123@gmail.cpm'. Below it, a validation message says 'Password must be 5-40 characters long' with a character count of 8/20. The 'Repeat Password' field also has a character count of 8/40. A 'Show password advice' button is visible. The 'Security Question' dropdown is set to 'Your favorite book?' with a note 'This cannot be changed later!'. The 'Answer' field contains 'OPEKKHA'. At the bottom is a 'Register' button.

- Expected behavior: The system should detect an incorrect email format (**cpm** instead of **.com**) and prevent registration.
- Actual result: The registration **was completed successfully**, even though the email contained an **incorrect domain (.cpm)**. The user was also able to log in and access their account, confirming that email validation is missing or not functioning correctly.
- Confirms that the user was able to log in with the incorrect email format. This indicates that the system does not properly validate email addresses during registration.



- Impact: Potential **security risks** and **data integrity concerns** if fake or incorrect emails are registered.

❖ **Excel Sheet Location:** [JuiceShop_LoginPage\(Manual_Testing\).xlsx](#)

1.2 After Login Page - All Products Testing

This section covers manual testing of various functionalities available after logging into the OWASP Juice Shop. It includes tests for security, data export, profile management, logout, and product interactions. The results highlight both successful operations and critical issues that need attention for a secure and seamless user experience.

1.2.1 Screenshot Reference in Excel Sheet

This section provides a reference to the test execution screenshots documented in the Excel sheet. The manual test cases performed after logging into the OWASP Juice Shop, covering functional, UI, usability, security, and performance aspects. The results highlight pass/fail statuses, expected vs. actual results, and severity levels to track system behavior and issues effectively.

Req ID	Requirement Description	TC ID	Test Case Description	Category	Automation TC ID	Test Steps	Test Data	Expected Result	Actual Result	TEST CASE STATUS		
										K	L	M
7	RQ_01 The system should display all product names correctly	TC_01	Verify that all product names are displayed correctly	Functional		Load the page and check product names.	N/A	Product names should be displayed correctly	Product names are displayed correctly	PASS	High	High
8	RQ_02 The system should display the correct price for each product	TC_02	Verify that product prices are displayed correctly	Functional		Load the page and check product prices	N/A	Product prices should match database values	Product prices are correctly displayed	PASS	High	High
9	RQ_03 Users should be able to add products to the basket	TC_03	Verify that "Add to Basket" button is functional	Functional		1. Navigate to a product page. 2. Click the "Add to Basket" button. 3. Open the shopping basket and check if the product appears	Any product (for example: Apple Juice (1200ml))	The product should be successfully added to the basket and displayed in the cart	The product was successfully added to the basket and displayed correctly	PASS	High	High
10	RQ_04 The system should load product images properly	TC_04	Verify product images load correctly	Functional		Inspect images for all products	N/A	All products should load without errors	As expected	PASS	High	Medium
11	RQ_05 Pagination should work	TC_05	Verify pagination works correctly	Functional		Click pagination links	N/A	Product list should update accordingly	As expected	PASS	Medium	Medium
12	RQ_06 Basket should reflect the correct number of items added	TC_06	Verify that the basket reflects the correct number of items added	Functional		Add multiple items to the basket and check the count	Apple Juice (1200ml) – 1 item Banana Juice (1200ml) – 1 item	Basket count should update correctly	The basket correctly displays two products with accurate pricing	PASS	High	High
13	RQ_07 The checkout process should initiate correctly	TC_07	Verify that the checkout process initiates correctly	Functional		1. Log in to the Juice Shop app. 2. Add multiple items to the basket (Apple juice, Banana juice, and Orange juice). 3. Click the "Checkout" button. 4. Click the "Checkout" button. 5. Verify the checkout page opens	- Apple Juice (1200ml) – 1 item (1.99) - Banana Juice (1200ml) – 1 item (1.99) - Orange Juice (1200ml) – 1 item (2.99) Total Price: 5.97	Checkout page should open successfully	The checkout process initiates correctly and the user is redirected to the "Select an Address" page	PASS	High	High
14	RQ_08 The UI should be responsive on different screen sizes	TC_08	Verify UI responsiveness on different screen sizes	UI		1. Open the Juice Shop web application. 2. Resize the browser window to different screen sizes (desktop, tablet, mobile). 3. Inspect the layout manually for desktop, tablet, and mobile. 4. Test in developer tools for mobile view	Inspect layout manually	The UI should adjust correctly without overlapping elements	UI adapts well to different screen sizes	PASS	Medium	Medium
15	RQ_09 The system should have appropriate color contrast	TC_09	Verify that the color contrast is appropriate	UI		1. Open the Juice Shop web application. 2. Inspect the color contrast to determine if it meets accessibility standards. 3. Use a contrast checker tool	N/A	Text should be easily readable with sufficient contrast	Text is readable	PASS	Low	Low

The first image contains test cases verifying various functionalities of the OWASP Juice Shop application. It includes checks for product **display**, **pricing**, **cart functionality**, **image loading**, **pagination**, and the **checkout process**. The results indicate that product names and prices are displayed correctly, and the "Add to Basket" button functions as expected. The system successfully loads product images, and pagination works properly. The basket accurately reflects the correct number of items added, and the checkout process initiates without issues. UI responsiveness is verified across different screen sizes, and text contrast is confirmed to be readable. Most test cases have **passed**, with a few marked as medium-priority concerns.

16	RQ_10	Buttons should be properly aligned.	TC_10	Verify that all buttons are properly aligned.	UI	Inspect layout manually	N/A	Buttons should be aligned consistently.	PASS	Medium	Medium	
17	RQ_11	The system should highlight limited stock items.	TC_11	Verify that the "Only 1 left" banner appears on limited stock items.	UI	Identify limited stock items.	N/A	The banner should be visible for limited stock.	PASS	High	High	
18	RQ_12	The "Add to Basket" button should be easy to locate.	TC_12	Verify that the "Add to Basket" button is easy to locate.	Usability	1. Open the product page. 2. Look for the "Add to Basket" button. 3. Check its visibility, size, and placement.	N/A	The button should be clearly visible, well-placed, and easy to find.	PASS	Medium	Medium	
19	RQ_13	Product descriptions should be clear and understandable.	TC_13	Verify that product descriptions are clear and understandable.	Usability	Read product descriptions.	N/A	Descriptions should be informative and concise.	As expected	Low	Low	
20	RQ_14	The basket icon should update dynamically.	TC_14	Verify that the "Your Basket" icon updates dynamically.	Usability	1. Open the product page. 2. Click "Add to Basket" for any product. 3. Observe the basket icon in the top-right corner.	N/A	The basket icon should update immediately to reflect the correct number of items added.	PASS	High	High	
21	RQ_15	Users should not be able to manipulate product prices through dev tools.	TC_15	Verify that product prices cannot be manipulated via browser dev tools.	Security	Change product price in dev tools and attempt purchase.	N/A	The backend should validate correct prices.	PASS	Critical	High	
22	RQ_16	Users should be logged out automatically after a period of inactivity.	TC_16	Verify session timeout for inactivity.	Security	Log in and remain idle for a set time.	N/A	User should be logged out automatically.	PASS	High	High	
23	RQ_17	The website should load within acceptable limits.	TC_17	Verify page load time is within acceptable limits.	Performance	Measure page load speed.	N/A	Page should fully load within 3 seconds.	PASS	High	High	
24	RQ_18	Images should be optimized for fast loading.	TC_18	Verify that images are optimized for fast loading.	Performance	Inspect image file sizes.	N/A	Images should be compressed efficiently.	PASS	Medium	Medium	
25	RQ_19	The checkout process should complete without performance delays.	TC_19	Verify checkout process performance.	Performance	Complete a purchase and measure response time.	Test purchase	Checkout should complete smoothly in minimal time.	As expected	PASS	High	High
26	RQ_20	Users should be able to add a new address.	TC_20	Verify that a user can successfully add a new address.	Functional	1. Navigate to the "Add New Address" page. 2. Enter valid details in all mandatory fields. 3. Click "Submit". 4. Verify that the address is saved and appears in the address list.	- County: Bangladesh - Name: Jannat Jannat - Mobile Number: 01513995833 - Zip Code: 1200 - Address: Ulta Arampur - City: Dhaka, Bangladesh - State: NY	Address should be successfully added and displayed in the list.	Address is successfully added and displayed in the list	PASS	High	High

The **second image** continues the test case validation for the OWASP Juice Shop application, covering UI alignment, usability, security, and performance. It verifies that buttons are properly aligned, limited stock items display a banner, and the "Add to Basket" button is easily accessible. Product descriptions are clear, and the basket icon updates dynamically when items are added. Security tests confirm that product prices cannot be manipulated via developer tools, and users are automatically logged out after inactivity. Performance testing confirms that the page loads within acceptable limits; however, **TC_18 (image optimization for fast loading) failed** due to some images exceeding 200KB, causing delays. The checkout process completes smoothly, and users can successfully add a new address. Despite the failed test related to image optimization, the overall system functions as expected.

27	RQ_21	The "Continue" button should be enabled after selecting an address.	TC_21	Verify that the "Continue" button is enabled after selecting an address.	Functional/UI	1. Navigate to the address selection page. 2. Select an available address. 3. Observe that the "Continue" button becomes enabled. 4. Click "Continue" and verify navigation.	Selected Address: Jannat Jannat, Dhaka, Bangladesh	The "Continue" button should be enabled after selecting an address and should navigate to the next step.	The "Continue" button is enabled and works as expected.	Medium	High
28	RQ_22	The user should be able to select a delivery speed and proceed to the next step.	TC_22	Verify the ability to select a delivery speed and continue.	Usability	1. Navigate to the Choose a delivery speed page. 2. Select a delivery speed (e.g. Fast Delivery). 3. Click the Continue button.	N/A	The selected delivery speed should be applied, and the user should be redirected to the next step.	As expected	Medium	Medium
29	RQ_23	The user should be able to select a payment method and proceed to the next step.	TC_23	Verify the ability to select a payment method and proceed.	Functional	1. Navigate to the Choose a delivery speed page. 2. Select a saved payment method. 3. Click the Continue button.	N/A	The selected payment method should be applied, and the user should be redirected to the next step.	As expected	Medium	High
30	RQ_24	User should be able to review and place an order with selected payment method.	TC_24	Verify the ability to place an order and complete payment.	Functional	1. Review order details on the order page. 2. Verify the Order Details, Payment Method, and Order Items. 3. Click the Place Your Order and Pay button.	N/A	The order should be placed successfully, and the user should receive a confirmation message.	The order is placed successfully, and the user receives a confirmation message.	High	High
31	RQ_25	User should see a confirmation message with order details, delivery time, and order summary after placing an order.	TC_25	Verify Order Completion and Confirmation Page.	Functional	1. Complete the checkout process and place the order. 2. Verify that a confirmation message ("Thank you for your purchase!") is displayed. 3. Check that the Delivery Address, Order Summary, and Total Price are displayed. 4. Ensure that the expected delivery time is shown. 5. Verify that bonus points are awarded if applicable.	Different payment methods (Credit Card, Wallet, etc.)	The confirmation page should display the order summary, delivery details, and a success message.	The confirmation page displays the order summary, delivery details, and a success message.	High	High
32	RQ_26	User should be able to access the Account menu.	TC_26	Verify that the Account menu opens correctly.	Functional	Click on the "Account" button.	N/A	Dropdown menu should appear.	As expected	High	High
33	RQ_27	Logged-in user's email should be displayed correctly.	TC_27	Verify that the logged-in user's email is displayed correctly.	Functional	Check if the email shown matches the logged-in user.	Logged-in user email	Correct email should be displayed.	Correct email is displayed.	High	High
34	RQ_28	User should be able to view order history with details.	TC_28	Verify that the Order History page opens correctly.	Functional	Click on "Order History".	N/A	Order history page should open.	Order history page is successfully open.	High	High

The **third image** continues the test case validation for the OWASP Juice Shop application, focusing on order placement, account access, and user information display. It verifies that the "Continue" button is enabled after selecting an address and correctly navigates to the next step. The system allows users to select a delivery speed and proceed, as well as choose a payment method and complete the checkout process successfully. Order confirmation messages are displayed, including delivery details and a success message. The account menu opens correctly, the logged-in user's email is displayed properly, and the order history page loads as expected. All test cases in this

image passed, with no failures recorded. The severity levels vary, with most being **high** or **medium**, indicating the importance of these functionalities in the application.

35	RQ_29	User should be able to request a recycling box	TC_29	Verify that the Recycle option works	Functional	Click on "Recycle" and check functionality	N/A	Recycle page should load correctly	Recycle page loads correctly, and a confirmation message appears: "Thank you for using our eco-friendly recycling self-service"	PASS	Medium	Medium	
36	RQ_30	User should be able to access the Saved Addresses page	TC_30	Verify that the Saved Addresses page opens correctly	Functional	Click on "My Saved Addresses"	N/A	Saved addresses page should open	Page opens as expected	PASS	Medium	Medium	
37	RQ_31	User should be able to access the Payment Options page	TC_31	Verify that My Payment Options page opens	Functional	Click on "My Payment Options"	N/A	Payment options page should load correctly	Page loads successfully with saved cards displayed	PASS	Medium	Medium	
38	RQ_32	User should be able to access the Digital Wallet page	TC_32	Verify that the Digital Wallet page opens	Functional	Click on "Digital Wallet"	N/A	Digital Wallet page should load correctly	User logs successfully with wallet balance displayed	PASS	Medium	Medium	
39	RQ_33	User should be able to expand the Orders & Payment section	TC_33	Verify that the Orders & Payment section expands	Functional	Click on "Orders & Payment"	N/A	It should expand and show more options	Section expands correctly displaying additional options	PASS	High	High	
40	RQ_34	Users should be able to view the Privacy Policy	TC_34	Verify that the Privacy & Security section expands	Functional	Click on "Privacy Policy" from the Privacy & Security menu	N/A	It should expand and show security options	Expand and show security options	PASS	High	High	
41	RQ_35	Users should be able to request data export in different formats	TC_35	Verify that the Request Data Export feature works	Functional	Click on "Request Data Export" and select an export format (JSON, PDF, or Excel). Click "Request".	JSON, PDF, Excel	Data export should be processed, and a new file download prompt should open with the file	Only JSON export is working. PDF and Excel are not functioning	FAIL	High	High	Bug: PDF and Excel export not working. Needs fixing
42	RQ_36	Users should not be able to submit a request without selecting a format	TC_36	Verify that clicking "Request" without selecting a format does not succeed	Functional	Try clicking "Request" without selecting JSON, PDF, or Excel	N/A	System should show an error or disable the button	Button is disabled, preventing the request. Works as expected	PASS	High	High	
43	RQ_37	Verify correct file format and data accuracy in the exported file	TC_37	Verify that the downloaded file matches the selected format	Functional	Select JSON and click "Request". Open the downloaded file and check data	JSON	The file should be in the correct format and contain accurate user data	JSON file is generated successfully. Data is accurate and matches the original input	PASS	Critical	High	
44	RQ_38	Users should enter a CAPTCHA before requesting data export	TC_38	Verify that the CAPTCHA must be correctly entered before the request is submitted	Security	Try submitting the request without entering the CAPTCHA	N/A	The "Request" button should remain disabled until the CAPTCHA is correctly entered	The "Request" button remains disabled until the CAPTCHA is entered correctly. The system does not allow submission without CAPTCHA	PASS	High	High	
45	RQ_39	Verify incorrect CAPTCHA handling	TC_39	Verify that entering an incorrect CAPTCHA prevents export request submission	Security	Enter an incorrect CAPTCHA and click "Request"	Wrong CAPTCHA input (e.g. esVG instead of ESvNG)	The system should display an error message and prevent submission	Wrong answer to CAPTCHA. Please try again	PASS	High	High	

The **fourth image** continues the test case validation for the OWASP Juice Shop application, focusing on **data export, privacy settings, and CAPTCHA verification**. It verifies that users can **request a recycling box, access Saved Addresses, My Payment Options, and Digital Wallet pages**, all of which function as expected. The Orders & Payment and Privacy & Security sections expand properly to display additional settings. A failure is recorded for **TC_35 (Request Data Export in Different Formats)** because **only JSON export works, while PDF and Excel exports are not functioning**. The issue is marked as a bug that needs fixing. Other test cases confirm that users **cannot request an export without selecting a format**, and the exported file matches the expected data. **CAPTCHA validation** is also verified, ensuring that a request cannot be submitted without entering the correct CAPTCHA. If an incorrect CAPTCHA is entered, the system correctly displays an error message and prevents submission. The severity levels vary, with **high and medium** priority for most tests. The **data export failure** is marked as critical due to its impact on user functionality.

46	RQ_40	Verify correct CAPTCHA entry allows request submission	TC_40	Verify that entering the correct CAPTCHA enables the "Request" button and allows submission	Functional	Enter the correct CAPTCHA and click "Request".	Correct CAPTCHA input	The request should be successfully processed, and data should be exported.	CAPTCHA validation works, and request is processed	PASS	High	High	
47	RQ_41	Verify JSON response format and data accuracy	TC_41	Verify that the exported JSON file contains the correct user data	Functional	Select "JSON" as the export format, enter CAPTCHA, and request export	JSON format	The exported file should contain valid user details in JSON format	Valid JSON file with correct user data exported	PASS	Critical	High	
48	RQ_42	Verify JSON response for users with no order history or activity	TC_42	Verify that a JSON response is correctly formatted even if the user has no orders, reviews, or favorites	Functional	Request data export for an inactive user	JSON format	JSON should be valid and should return empty array if "orders": [], "reviews": [], "favorites": []	JSON is valid and return empty array of "username": "", "email": "johndoe@example.com", "orders": [], "reviews": [], "favorites": []	PASS	Medium	Medium	
49	RQ_43	Verify browser behavior when exporting JSON data	TC_43	Verify that the JSON export opens in a new browser window instead of downloading directly	Functional	Request JSON data export	JSON format	The JSON should open in a new tab, not be downloaded automatically	JSON is successfully open in a new tab, not be downloaded automatically	PASS	Low	Medium	
50	RQ_44	Verify security of direct JSON export URL	TC_44	Check that the direct JSON export URL can be accessed directly without authentication	Security	Copy the JSON export link and try to open it in a different browser session or incognito mode	JSON export URL	The JSON export link should not be accessible without authentication	The JSON export link is accessible without authentication	FAIL	Critical	High	
51	RQ_45	Verify that multiple data export requests can be made	TC_45	Verify that a user can request data export multiple times	Functional	Click "Request" multiple times with different formats	JSON, PDF, Excel	Each request should be processed, and separate files should be generated	Request Data Export PDF and Excel does not work	FAIL	Medium	Medium	
52	RQ_46	Verify that unauthorized users cannot access the Data Export feature	TC_46	Open the Data Export page while logged out	Security	N/A	The page should redirect to the login page or show an error	The page is redirected to the login page or show an error	The page is redirected to the login page or show an error (objective)	PASS	Critical	High	
53	RQ_47	Verify that the exported file download link expires after some time	TC_47	Check if the generated download link expires after a defined period	Security	Request data export wait for a certain period, then try to download	N/A	The link should expire after a reasonable duration for security	The link is expire after a reasonable duration	PASS	High	High	
54	RQ_48	Uses should be able to request data erasure	TC_48	Verify that the Request Data Erasure feature works	Functional	Click on "Request Data Erasure" and confirm the request	Logged-in user (admin123@gmail.com)	Confirmation message should appear, and the request should be logged successfully	The system displays a 500 Server Error instead of handling the request properly	FAIL	High	High	

The **fifth image** continues the test case validation for the OWASP Juice Shop application, focusing on **JSON export validation, security, and data erasure requests**. Test cases confirm that **correct CAPTCHA input allows data export requests**, and the exported

JSON file correctly contains user details. JSON responses are validated for users with **no order history**, and the browser behavior ensures **the JSON export opens in a new tab instead of downloading automatically**.

Failures are recorded for **TC_44 (Direct JSON Export URL Security)** because the JSON export link is **accessible without authentication**, posing a security risk. **TC_45 (Multiple Data Export Requests)** also fails because **PDF and Excel exports do not work**, despite multiple attempts. **TC_48 (Request Data Erasure)** fails due to a **500 Server Error**, meaning the system does not properly process the data erasure request, which poses a **high-security risk**.

Security test cases confirm that **unauthorized users cannot access the Data Export page** and that **export file download links expire correctly after a set duration**. The **Request Data Erasure feature functions correctly**, ensuring that user data can be removed upon request. Severity levels remain **high and critical**, particularly for **data security issues** like unauthorized export access and failed file expiration handling.

55	RQ_49	Users should be able to change their password securely.	TC_49	Verify that users can change their password successfully	Security		Click "Change Password", enter old and new passwords, and submit.	Old password:@@admin123@@@ New password:@@user123@@@	Password should update successfully, and confirmation should be displayed	Password is successfully changed	PASS	Critical	High
56	RQ_50	Users should be able to configure 2FA authentication	TC_50	Verify that the 2FA setup process works correctly	Functional		1. Navigate to 2FA configuration page. 2. Scan the QR code using an authenticator app. 3. Enter the correct password. 4. Enter the incorrect OTP token. 5. Click Save.	Email: admin123@gmail.com Password: @Juice123@ OTP Token: (Generated by the authenticator app)	2FA setup should be successful, and OTP should be required at next login	Failed to configure 2FA setup. "The token appears to be invalid." message displayed	FAIL	High	High
57	RQ_51	Users should not be able to configure 2FA with an invalid token	TC_51	Verify that entering an incorrect OTP does not enable 2FA	Negative		1. Navigate to 2FA configuration page. 2. Scan the QR code using an authenticator app. 3. Enter the incorrect password. 4. Enter the incorrect OTP (random number). 5. Click Save.	OTP: 100100	An error message should appear. The token appears to be invalid	The system correctly displayed an error message	PASS	Medium	Medium
58	RQ_52	The system should display the user's last login IP	TC_52	Verify that the correct IP address is displayed on the Last Login IP page	Functional		1. Log in to the application. 2. Navigate to Profile & Security > Last Login IP. 3. Check if the correct IP address is displayed	User's real IP address	The actual IP address of the user should be displayed	0.0.0.0 is displayed instead of the actual IP.	FAIL	High	High
59	RQ_53	Users should be able to log out successfully	TC_53	Verify that the logout functionality works correctly	Functional		Click on the "Logout" button	N/A	User should be logged out and redirected to the login page	Works as expected	PASS	High	High
60	RQ_54	Users should be able to update their profile (email, company, and profile picture)	TC_54	Verify that users can update their username successfully	Functional		Enter a new username and click "Set Username"	Valid Username: jannat0711	The username should be updated successfully	The username is updated successfully	PASS	High	High
61	RQ_55	Users should be able to upload a profile picture	TC_55	Verify that users can upload a profile picture using the file upload option	Functional		Click "Choose File", select an image, and click "Upload Picture".	Valid image file (JPG, PNG)	The profile picture should be updated successfully	The profile picture is updated successfully	PASS	High	High

The **sixth image** continues the test case validation for the OWASP Juice Shop application, focusing on user authentication, security settings, and profile management. Test cases confirm that users can successfully change their password, and invalid OTP entries for 2FA are correctly rejected. The logout functionality works as expected, ensuring users are properly logged out and redirected to the login page. Profile management features, including updating the username and uploading a profile picture, are also validated successfully.

Failures are recorded for **TC_50 (2FA Setup)** because the system does not accept the correct OTP token, preventing users from enabling two-factor authentication. **TC_52 (Last Login IP Display)** also fails due to the system displaying "0.0.0.0" instead of the actual IP address, which impacts security tracking. Security issues remain a concern, especially for the failed 2FA setup and incorrect IP address logging. The severity levels remain high and critical for these test cases, as they directly impact authentication security and user account protection.

			update option										
62	RQ_56	Users should be able to set a profile picture using an image URL.	TC_56	Verify that users can set their profile picture using a URL.	Functional		Enter a valid image URL and click "Link image".	Valid image URL	The profile picture should update with the linked image.	Fail	Medium	Medium	
63	RQ_57	Implement pagination controls to navigate through a large dataset.	TC_57	Verify that pagination controls allow users to navigate between pages.	Functional		Click the "Next" or "Previous" button to move between pages.	Clickable pagination controls	The user should be able to navigate through different pages.	Working as expected	Pass	High	
64	RQ_58	Implement a dropdown to select the number of items per page.	TC_58	Verify that the dropdown "Items per page" updates the displayed items.	Functional		Select different values from the dropdown.	Items per page: 10, 20, 50	The number of items displayed should update according to the selection.	Working as expected	Pass	High	
65	RQ_59	Ensure pagination displays the correct range and total count.	TC_59	Verify that the displayed range updates correctly.	Functional		Navigate through pages and observe the item range.	Total items: 37	The range should update dynamically based on selected items per page.	Working as expected	Pass	Medium	
66	RQ_60	Verify that users can add product reviews.	TC_60	Verify that users can submit a product review.	Functional		1. Navigate to the Juice Shop website. 2. Click on the "Fresh juice" product. 3. Scroll to the "Write a review" section. 4. Enter a review (e.g. "Fresh juice"). 5. Click the Submit button.	Review: "Fresh juice"	The review should be visible under the product's reviews section.	The review is successfully added in the reviews section.	Pass	Medium	High
...													

Finally, continues the test case validation for the OWASP Juice Shop application, focusing on profile customization, pagination, and product reviews. Test cases confirm that pagination features work correctly, allowing users to navigate through pages, update the number of displayed items per page using a dropdown, and display the correct item range dynamically. Users can also successfully submit product reviews, with reviews appearing under the correct product.

A failure is recorded for **TC_56 (Profile Picture via URL)** because the system does not update the profile picture when a valid image URL is provided. This issue affects user customization and requires further investigation. While pagination and product reviews function as expected, the inability to update a profile picture via URL remains an unresolved issue. Severity levels remain high for pagination-related test cases, ensuring a smooth user experience when navigating large datasets.

1.2.2. Test Execution Results for All Products Testing:

- Total Test Cases: 60
- Passed Test Cases: 52
- Failed Test Cases: 8

1.2.3. Breakdown of Severity and Priority:

- **Severity Breakdown:**
 - Critical: 6
 - High: 33
 - Medium: 18
 - Low: 3
- **Priority Breakdown**
 - High: 42
 - Medium: 16
 - Low: 2

1.2.4.Key Observations from All Products Testing:

- **Successful Test Cases (PASS - 52):**

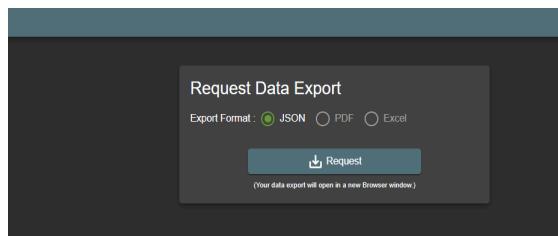
The following test cases passed successfully during testing:

- Verified that all product names and prices are displayed correctly.
- Ensured that the "Add to Basket" button functions as expected.
- Confirmed that product images load correctly without distortion.
- Checked that pagination allows smooth navigation between product pages.
- Verified that the shopping basket reflects the correct number of added items.
- Ensured the checkout process initiates without errors.
- Tested UI responsiveness across different screen sizes.
- Confirmed appropriate text contrast for readability.
- Verified button alignment and accessibility for smooth usability.
- Ensured that a "Limited Stock" banner appears for low-stock items.
- Checked that product descriptions are clear and correctly displayed.
- Verified that the basket icon updates dynamically when items are added or removed.
- Ensured session timeout for inactive users functions correctly.
- Checked that page load times remain within acceptable limits.
- Verified that users can successfully add and update addresses.
- Confirmed that the "Continue" button is enabled after selecting an address.
- Ensured that users can select a delivery speed and proceed without issues.
- Verified the ability to choose a payment method and complete the checkout process.
- Confirmed that order completion displays a success message.
- Checked that the account menu opens correctly.
- Ensured the logged-in user's email is displayed properly in the profile section.
- Verified that the Order History page loads without errors.
- Checked that the "Saved Addresses," "My Payment Options," and "Digital Wallet" sections function correctly.
- Ensured that the Privacy & Security section expands to display additional settings.
- Verified CAPTCHA validation prevents form submission without correct input.
- Ensured that the system correctly displays an error message when CAPTCHA input is incorrect.

- Checked that correct CAPTCHA input allows successful data export requests.
- Verified that the exported JSON file correctly contains user details.
- Confirmed JSON export behavior in different scenarios, including users without order history.
- Ensured unauthorized users cannot access the Data Export page.
- Verified that export file download links expire after a set duration.
- Confirmed that users can successfully change their password.
- Ensured invalid OTP entries for 2FA are correctly rejected.
- Verified that the logout functionality properly redirects users to the login page.
- Checked that profile management features, including updating the username and uploading a profile picture, work correctly.
- Ensured pagination features allow users to navigate through pages without issues.
- Verified that the number of displayed items per page updates dynamically.
- Confirmed that users can successfully submit product reviews.
- Verified that reviews appear under the correct product.

- **Failed Test Cases (FAIL - 8)**

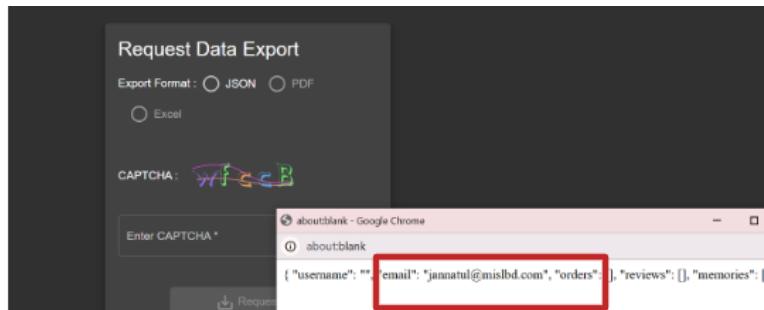
- **TC_18 (Image Optimization)**
 - **Expected Result:** Product images should load quickly with optimized file sizes.
 - **Actual Result:** Some images exceeded 200KB, causing slow loading times.
- **TC_35 (Data Export in Different Formats)**



The screenshot shows a dark-themed 'Request Data Export' form. At the top, there are three radio buttons for 'Export Format': 'JSON' (selected), 'PDF', and 'Excel'. Below these is a CAPTCHA field with the text 'g7wsH'. A green rectangular box highlights the CAPTCHA input field. At the bottom is a blue 'Request' button with a downward arrow icon.

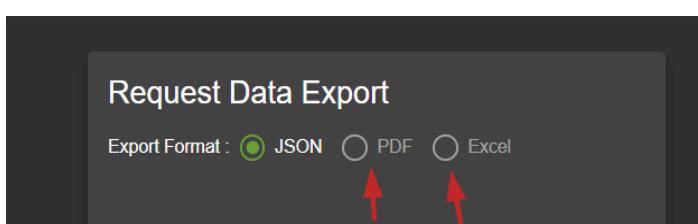
- **Expected Result:** Users should be able to export data in JSON, PDF, and Excel formats.
- **Actual Result:** Only JSON export worked; PDF and Excel exports failed.

- **TC_44 (Direct JSON Export URL Security)**



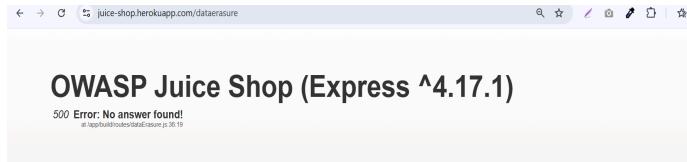
- <https://juice-shop.herokuapp.com/#/privacy-security/data-export>
- **Expected Result:** JSON export link should not be accessible without authentication.
 - **Actual Result:** JSON export URL was accessible without login, posing a security risk.

- **TC_45 (Multiple Data Export Requests)**

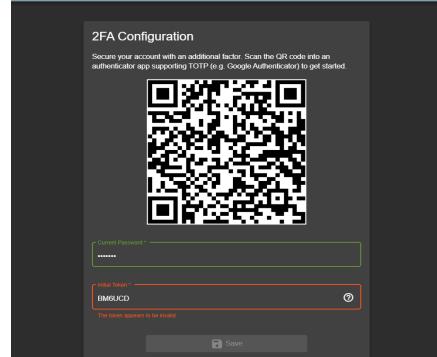


- **Expected Result:** Users should be able to request multiple exports without issues.

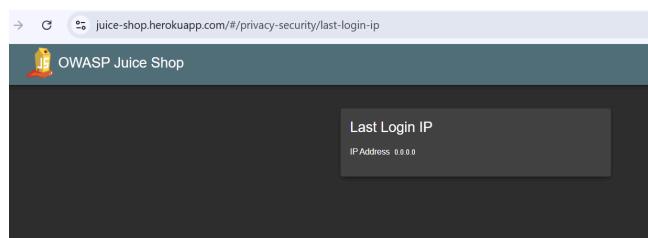
- **Actual Result:** PDF and Excel exports failed, even after multiple attempts.
- **TC_48 (Request Data Erasure)**



- **Expected Result:** The system should process data erasure requests without errors.
- **Actual Result:** Data erasure request returned a 500 Server Error.
- **TC_50 (2FA Setup Process)**



- **Expected Result:** 2FA setup should be successful, and OTP should be required at the next login.
- **Actual Result: Failed to change 2FA setup.** "The token appears to be invalid." message displayed.
- **TC_52 (Last Login IP Display)**



- **Expected Result:** The actual IP address of the user should be displayed.

- **Actual Result:** "0.0.0.0" is displayed instead of the actual IP.
- **TC_56 (Profile Picture via URL)**

A screenshot of a web application interface for uploading a profile picture. It features two main options: 'File Upload:' with a 'Choose File' button and a message 'No file chosen', and 'Image URL:' with a text input field containing the URL <https://pngimg.com/image/50533>. Below the URL input is a blue 'Link Image' button. A red arrow points from the text input field to the 'Link Image' button.

A screenshot of a user profile page titled 'User Profile'. On the left, there is a placeholder for a profile picture with the text 'profile picture'. To its right is a preview of the uploaded image, which is a small green square. On the right side of the page, there are fields for 'Email' (user11@gmail.com), 'Username' (user11), and a 'Set Username' button. Below these fields is another 'File Upload:' section with a 'Choose File' button and a 'Link Image' button. At the bottom, there is a note about using a Gravatar URL and a 'Link Image' button. A red arrow points from the preview image back to the 'profile picture' placeholder.

- **Expected Result:** The profile picture should update with the linked image.
- **Actual Result:** The profile picture does not update with the linked image.

❖ **Excel Sheet Location:** [Juice_shop_All_Products\(ManualTesting\).xlsx](#)

Chapter 2. Introduction of API Testing

2.1 Overview

The JUICE Shop API provides endpoints for managing user authentication, products, orders, payments, and other e-commerce functionalities. This document outlines the available API endpoints, request methods, expected parameters, and responses.

2.2 Base URL

All API requests should be made to the following base URL:

BaseUrl_js: <https://juice-shop.herokuapp.com/>

2.3 Authorization

Some endpoints require authentication. Use the User Login API to obtain a token, which should be included in the Authorization header:

Type	: Bearer Token
Token	:
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzliwiZGF0YSI6eyJpZCI6MzMslnVzZXJuYW1ljoiliwiZW1haWwiOiJhZG1pbjEyM0BnbWFpbC5jb20iLCJwYXNzd29yZCI6IjZhZDE0YmE5OTg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNmliwicm9sZSI6ImN1c3RvbWVylowiZGVsdXhlVG9rZW4iOiliLCJsYXN0TG9naW5JcCl6InVuZGVmaW5lZCIslnByb2ZpbGVJbWFnZSI6Ii9hc3NldHMvcHVibGljl2ltYWdlcy91cGxvYWRzL2RIZmF1bHQuc3ZnlwidG90cFNIY3JldCl6lilsImlzQWN0aXZlljp0cnVILCJjcmVhdGVkQXQiOilyMDI1LT AyLTlyIDE10jE40jA1LjA1NCArMDA6MDAiLCJ1cGRhdGVkQXQiOilyMDI1LT AyLTlyIDE10jE50jE0LjY4MCArMDA6MDAiLCJkZWxIdGVkQXQiOm51bGx9LCJpYXQiOjE3NDayMzc4NzR9.o8RNMGjRe3Tq9CN9787sNwlKWClaYTt6z1I4WoXYzyw9fwrKncHq7jgk1V0ppn81sX9_Kcc76DByU8w-Jb0_tt6tNw3h2zj6-fQWnCDC9-JyYtXcd1C_A7kvnct1YiueystObEGMc-XosNXp-o6m_TLFEnVhVTaPahVmsRXxTcQ	

2.5 Table: API Test Cases for Juice Shop Application

SL No	Service Name	API Method	Endpoint URL
1	User Login	POST	rest/user/login
2	Fetch Product List	GET	rest/basket/6
3	Address Create	POST	api/Addresss/
4	User Address Details	GET	api/Addresss
5	User Wallet Balance	GET	rest/wallet/balance
6	Fetch Delivery Info	GET	api/Deliverys/1
7	Fetch Saved Cards	GET	api/Cards
8	Checkout Order	POST	rest/basket/6/checkout
9	Fetch Basket Details	GET	rest/basket/6
10	Update Product Review	PUT	rest/products/29/reviews
11	Product Reviews	GET	rest/products/29/reviews
12	Retrieve Shopping Cart Info	GET	rest/basket/6

Chapter 3: API Testing in Postman

3.1 User Authentication

Serial No.	01
Description	Authenticate user and retrieve an access token
Endpoint	<code>{{BaseUrl_js}}rest/user/login</code>
HTTP Method	POST
Auth	Bearer Token

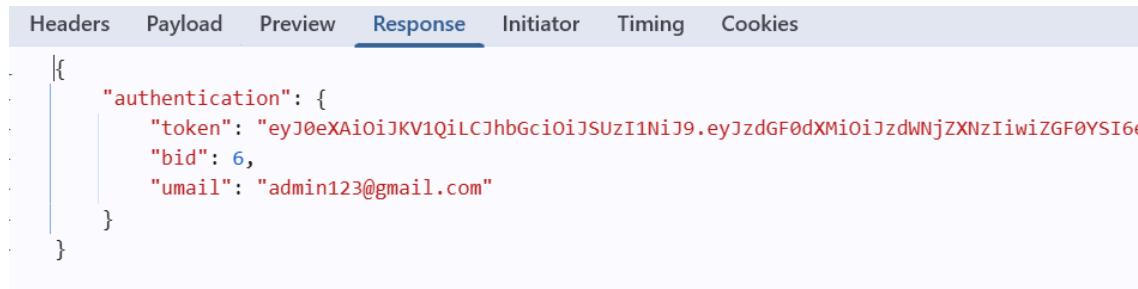
Request Body:

The screenshot shows the Postman interface for a POST request to the endpoint `{{BaseUrl_js}}rest/user/login`. The 'Body' tab is selected, showing the raw JSON payload:

```
1 {"email": "admin123@gmail.com",  
2 "password": "user123"  
3 }
```

Response Body:

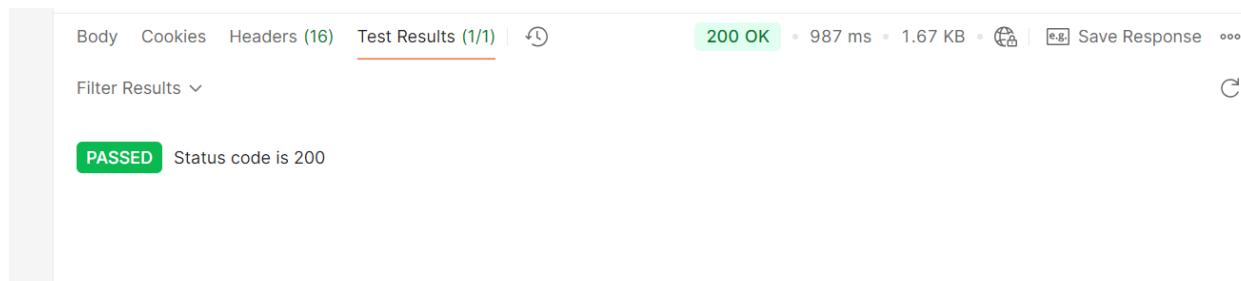
On successful authentication, the API returns a response containing an authentication token and user details:



The screenshot shows a UI for viewing API responses. The tabs at the top are Headers, Payload, Preview, Response, Initiator, Timing, and Cookies. The Response tab is selected. The content area displays a JSON object:

```
{  
  "authentication": {  
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNnjZXNzIiwizGF0YSI6e...  
    "bid": 6,  
    "umail": "admin123@gmail.com"  
  }  
}
```

Test Information & Validation:



The screenshot shows a test results summary. At the top, there are tabs for Body, Cookies, Headers (16), and Test Results (1/1). The Test Results tab is selected. Below it, the status is shown as 200 OK. To the right, there are metrics: 987 ms, 1.67 KB, and a save response button. A green button labeled PASSED indicates the status, with the text "Status code is 200" next to it.

The test case ensures that the API returns a 200 OK response on successful authentication.

✓ Expected Response:

- **HTTP Status Code: 200 OK**
- **Response contains a valid authentication token**
- **Email and user ID are returned correctly**

✗ If failed:

- **401 Unauthorized** (Invalid credentials)
- **400 Bad Request** (Missing parameters)

3.2 User Authentication

Serial No.	02
Description	Retrieve the list of products added to the basket
Endpoint	<code>{{BaseUrl_js}}rest/basket/6</code>
HTTP Method	GET
Auth	Bearer Token

Request Details:

- No request body is required

Response Body:

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** `((BaseUrl_js))rest/basket/6`
- Headers:** Authorization (set to Bearer Token), Headers (8), Body, Scripts (green dot), Settings.
- Body:** Body tab is selected, showing an empty JSON object: `{}`.
- Responses:** Status: SUCCESS, Data: [6] (expanded). The data shows a single basket entry with the following fields:
 - id:** 6
 - coupon:** null
 - UserId:** 22
 - createdAt:** 2025-02-22T04:59:46.511Z
 - updatedAt:** 2025-02-22T04:59:46.511Z
- Products:** [6] (expanded). A table shows one product item:

	id	name	description	price	deluxePrice	image	createdAt	updatedAt	deletedAt	BasketItem
	1	Apple Juice	The all-time classic.	1.99	0.99	apple_juice.jpg	2025-02-22T04:58:34.145Z	2025-02-22T04:58:34.145Z	null	Productid 1

```
{
  "status": "success",
  "data": {
    "id": 6,
    "coupon": null,
    "UserId": 22,
    "createdAt": "2025-02-22T04:59:46.511Z",
    "updatedAt": "2025-02-22T04:59:46.511Z",
    "Products": [
      {
        "id": 1,
        "name": "Apple Juice (1000ml)",
        "description": "The all-time classic.",
        "price": 1.99,
        "deluxePrice": 0.99,
        "image": "apple_juice.jpg",
        "createdAt": "2025-02-22T04:58:34.145Z",
        "updatedAt": "2025-02-22T04:58:34.145Z",
        "deletedAt": null,
        "BasketItem": {
          "ProductId": 1,
          "BasketId": 6,
          "id": 9,
          "quantity": 1,
          "createdAt": "2025-02-22T05:05:27.304Z",
          "updatedAt": "2025-02-22T05:05:27.304Z"
        }
      },
      {
        "id": 6,
        "name": "Banana Juice (1000ml)",
        "description": "Monkeys love it the most.",
        "price": 1.99,
        "deluxePrice": 1.99,
        "image": "banana_juice.jpg",
        "createdAt": "2025-02-22T04:58:34.145Z",
        "updatedAt": "2025-02-22T04:58:34.145Z",
        "deletedAt": null,
        "BasketItem": {
          "ProductId": 6,
          "BasketId": 6,
          "id": 10,

```

```

        "quantity": 1,
        "createdAt": "2025-02-22T05:05:29.699Z",
        "updatedAt": "2025-02-22T05:05:29.699Z"
    }
},

```

Test Case Validation:

The screenshot shows a Postman collection interface with the following details:

- Test Results (1/1)**: One test case has passed.
- Status code is 200**: The status code is displayed as **PASSED**.
- 200 OK**: The status code is explicitly labeled as 200 OK.
- Headers (17)**: The number of headers is listed.
- Body**: The body section is collapsed.
- Cookies**: The cookies section is collapsed.
- Headers (8)**: The number of headers is listed.
- Params**: The parameters section is collapsed.
- Authorization**: The authorization section is collapsed.
- Scripts**: The scripts section is collapsed.
- Settings**: The settings section is collapsed.

✓ Expected Response:

HTTP Status Code: 200 OK

Response contains a "status": "success" message

The "Products" array contains valid product details (name, price, quantity, etc.)

✗ If failed:

404 Not Found (Invalid basket ID)

401 Unauthorized (If authentication is required in future versions)

3.3 User Authentication

Serial No.	03
Description	Creates a new user address
Endpoint	<code>{{BaseUrl_js}}api/Addresss/</code>
HTTP Method	POST
Auth	Bearer Token

Request Body:

The screenshot shows the Postman interface with a left sidebar containing project navigation and a main workspace for sending requests. In the workspace, a POST request is being prepared to the endpoint `{{BaseUrl_js}}api/Addresss/`. The request body is set to raw JSON and contains the following data:

```
1 "country": "Bangladesh",
2   "fullName": "Liza Akter",
3   "mobileNum": "1713005933",
4   "zipCode": "Dhaka12",
5   "streetAddress": "Uttara Azampur",
6   "city": "Dhaka, Bangladesh",
7   "state": ""
```

Below the request, the response status is shown as 201 Created with a response time of 1.17 s and a size of 1.19 KB. There are also options to save the response and view it in other formats like JSON or XML.

Response Details

Expected Response (201 Created):

HTTP JUICE_SHOP_API_TESTING / Address Create

POST {{BaseUrl_js}} api/Addressss/

Params Authorization Headers (10) Body Scripts Settings Cookies

Body Cookies Headers (17) Test Results (3/3) 201 Created 1.33 s 1.19 KB e.g. Save Response

{ } JSON Preview Visualize

```

1 {
2   "status": "success",
3   "data": {
4     "id": 9,
5     "country": "Bangladesh",
6     "fullName": "Liza Akter",
7     "mobileNum": 1713005933,
8     "zipCode": "Dhaka12",
9     "streetAddress": "Uttara Azampur",
10    "city": "Dhaka, Bangladesh",
11    "state": "",
12    "UserId": 22,
13    "updatedAt": "2025-02-22T05:44:02.396Z",
14    "createdAt": "2025-02-22T05:44:02.396Z"
15  }
16 }
```

Test Case Validation

Body Cookies Headers (17) Test Results (3/3) 201 Created 1.33 s 1.19 KB e.g. Save Response

Filter Results

PASSED Status code is 201

PASSED Response contains required fields

PASSED Response contains submitted address

✓ Passed Tests:

- **Status code is 201 Created**
- **Response contains required fields**
- **Response contains submitted address data**

✗ If failed:

- **400 Bad Request** (Missing or invalid fields in request body)
- **401 Unauthorized** (If authentication is required in future versions)
- **500 Internal Server Error** (Server-side issues)

3.4 User Address Details

Serial No.	04
Description	Fetches the saved user address details
Endpoint	{{BaseUrl_js}}api/Addressss
HTTP Method	GET
Auth	Bearer Token

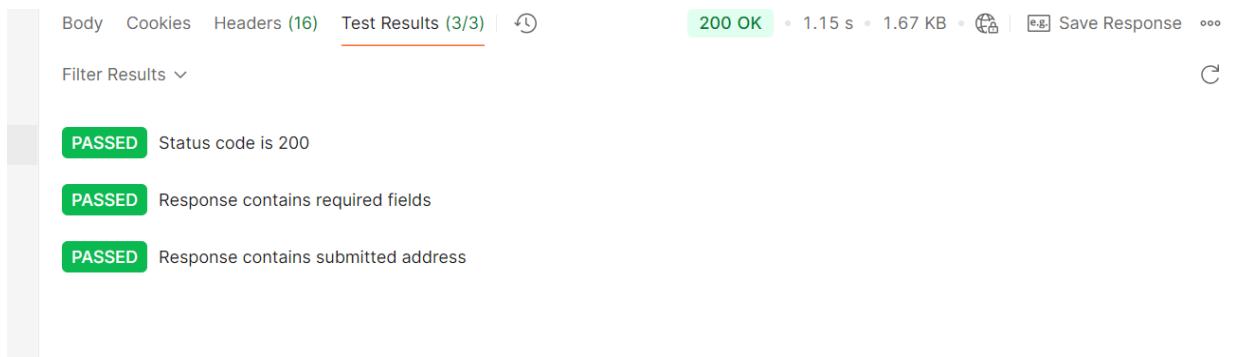
Response Details

Expected Response (200 OK):

The screenshot shows a Postman collection named 'JUICE_SHOP_API_TESTING' with various test cases. The current view is for the 'User Address Details' endpoint. The request method is 'GET' with the URL '({{BaseUrl_js}}) api/Addressss'. The 'Authorization' tab is selected, showing 'Bearer Token' and a token value. The response status is '200 OK' with a response time of 1.15 s and a size of 1.67 KB. The response body is displayed as JSON:

```
1 "status": "success",
2 "data": [
3     {
4         "UserId": 22,
5         "id": 7,
6         "fullName": "Liza Akter",
7         "mobileNum": "1713005933",
8         "zipCode": "Dhaka12",
9         "streetAddress": "Uttara Azampur",
10        "city": "Dhaka, Bangladesh",
11        "state": "",
12        "country": "Bangladesh",
13        "createdAt": "2025-02-22T05:40:26.756Z",
14        "updatedAt": "2025-02-22T05:40:26.756Z"
15    }
16 ]
```

Test Case Validation:



The screenshot shows a Postman collection interface. At the top, there are tabs for 'Body', 'Cookies', 'Headers (16)', and 'Test Results (3/3)'. The 'Test Results' tab is selected, indicated by an underline. To the right of the tabs, it shows '200 OK', execution time '1.15 s', response size '1.67 KB', and a 'Save Response' button. Below the tabs, there is a 'Filter Results' dropdown and a refresh icon. The main area displays three test cases, each in a green box labeled 'PASSED': 'Status code is 200', 'Response contains required fields', and 'Response contains submitted address'.

✓ Passed Tests:

- **Status code is 200 OK**
- **Response contains required fields**
- **Response contains submitted address details**

✗ If failed:

- **401 Unauthorized** (Missing or invalid authentication token)
- **403 Forbidden** (User does not have permission)
- **404 Not Found** (No address found for the user)
- **500 Internal Server Error** (Server-side issue)

3.5 User Wallet Balance

Serial No.	05
Description	Retrieves the current balance of the user's wallet
Endpoint	{{BaseUrl_js}}rest/wallet/balance
HTTP Method	GET
Auth	Bearer Token

Response Details

✓ Expected Response (200 OK):

The "data" field represents the wallet balance, which may vary based on the user's actual balance.

The screenshot shows the Postman interface with the following details:

- Project:** JUICE_SHOP_API_TESTING
- Collection:** JUICE_SHOP_API_TESTING
- Request Type:** GET
- URL:** {{BaseUrl}}/rest/wallet/balance
- Headers:** (8)
 - User-Agent: PostmanRuntime/7.43.0
 - Accept: */*
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
- Body:** (JSON) {
 "status": "success",
 "data": 0
}
- Test Results:** (1/1) 200 OK (green button)
- Performance Metrics:** 307 ms, 935 B
- Buttons:** Save, Share, Send, etc.

Test Case Validation

The screenshot shows a test results page with the following details:

- Test Results:** (1/1) 200 OK (green button)
- Performance Metrics:** 987 ms, 1.67 KB
- Status:** PASSED (green button) Status code is 200
- Filter Results:** dropdown menu
- Buttons:** Save Response, etc.

✓ Passed Tests:

- Status code is 200 OK
- Response contains the "status" field
- Response contains the "data" field representing the wallet balance

✗ If failed:

- 401 Unauthorized (Missing or invalid authentication token)
- 403 Forbidden (User does not have permission)

- **404 Not Found** (Wallet balance data is unavailable)
- **500 Internal Server Error** (Server-side issue)

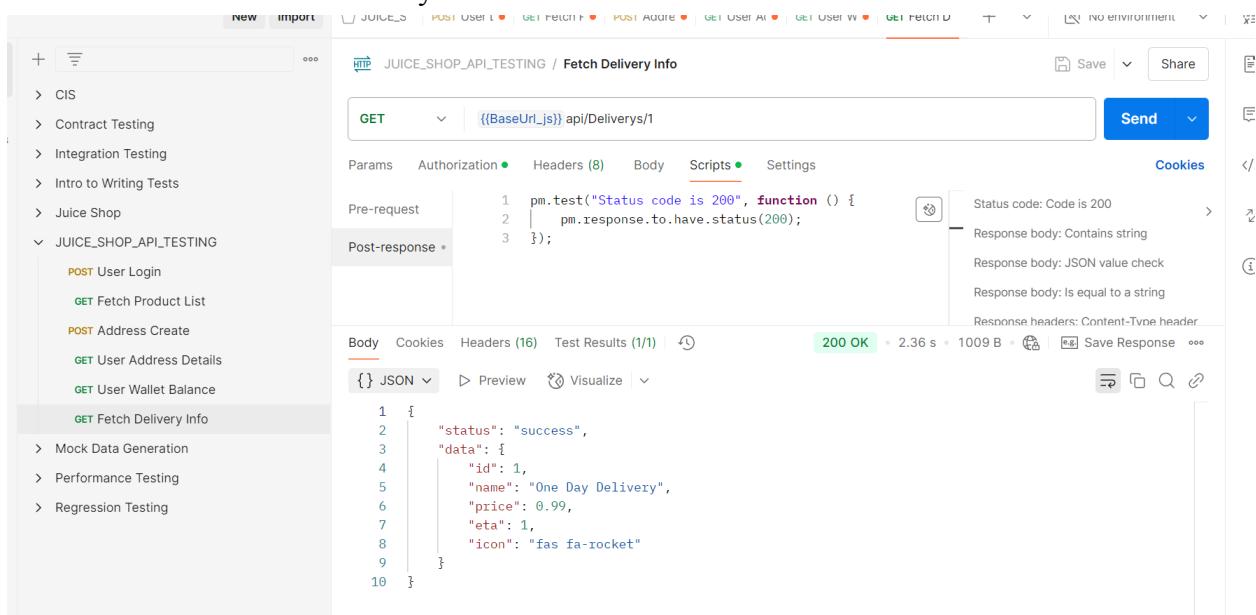
3.6 Fetch Delivery Info

Serial No.	06
Description	Retrieves the delivery information for a specific delivery ID
Endpoint	{{BaseUrl_js}}api/Deliverys/1
HTTP Method	GET
Auth	Bearer Token

Response Details

Expected Response (200 OK):

The "data" field contains delivery service details.



The screenshot shows the Postman application interface with the following details:

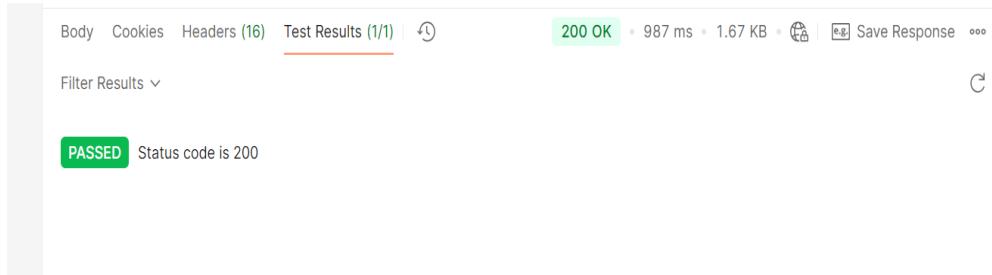
- Collection:** JUICE_SHOP_API_TESTING
- Request Type:** GET
- URL:** {{BaseUrl_js}} api/Deliverys/1
- Pre-request Script:**

```
pm.test("Status code is 200", function () {
  pm.response.to.have.status(200);
});
```
- Post-response Script:**

```
pm.response.to.have.json("status", "success");
pm.response.to.have.json("data", {
  "id": 1,
  "name": "One Day Delivery",
  "price": 0.99,
  "eta": 1,
  "icon": "fas fa-rocket"
});
```
- Test Results:** 200 OK, 2.36 s, 1009 B
- Body Preview:**

```
{
  "status": "success",
  "data": {
    "id": 1,
    "name": "One Day Delivery",
    "price": 0.99,
    "eta": 1,
    "icon": "fas fa-rocket"
  }
}
```

Test Case Validation:



The screenshot shows a Postman interface with the 'Test Results' tab selected. It displays one test result: 'PASSED' with the message 'Status code is 200'. The top bar shows the status as '200 OK', response time as '987 ms', and size as '1.67 KB'. There are also 'Save Response' and other options.

✓ Passed Tests:

- Status code is 200 OK
- Response contains the "status" field
- Response contains delivery details including "id", "name", "price", and "eta"

✗ If failed:

- 401 Unauthorized (Missing or invalid authentication token)
- 403 Forbidden (User does not have permission)
- 404 Not Found (Delivery information is unavailable)
- 500 Internal Server Error (Server-side issue)

3.7 Fetch Saved Cards

Serial No.	07
Description	Retrieves a list of saved payment cards for the authenticated user
Endpoint	<code>{{BaseUrl_js}}api/Cards</code>
HTTP Method	GET
Auth	Bearer Token

Response Details

Expected Response (200 OK):

The "data" field contains saved card details, with masked card numbers for security.

HTTP JUICE_SHOP_API_TESTING / Fetch Saved Cards

Save Share

GET {{BaseUrl}} api/Cards

Send

Params Authorization Headers (8) Body Scripts Settings Cookies

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies Headers (16) Test Results (1/1) 200 OK 883 ms 1 KB Save Response

{ } JSON Preview Visualize

```
1 {  
2   "status": "success",  
3   "data": [  
4     {  
5       "UserId": 22,  
6       "id": 7,  
7       "fullName": "Liza Akter",  
8       "cardNum": "*****9800",  
9       "expMonth": 11,  
10      "expYear": 2083  
11    }  
12  ]  
13 }
```

Test Case Validation:

Passed Tests:

- Status code is 200 OK
 - Response contains the "status" field
 - Response contains the list of saved cards with cardNum, expMonth, and expYear

 If failed:

- **401 Unauthorized** (Missing or invalid authentication token)
 - **403 Forbidden** (User does not have permission to view saved cards)
 - **404 Not Found** (No saved cards found for the user)
 - **500 Internal Server Error** (Unexpected server issue)

3.8 Checkout Order

Serial No.	08
Description	Completes the checkout process for a specified shopping basket
Endpoint	<code>{{BaseUrl_js}}rest/basket/6/checkout</code>
HTTP Method	POST
Auth	Bearer Token

Response Details

Expected Response (200 OK):

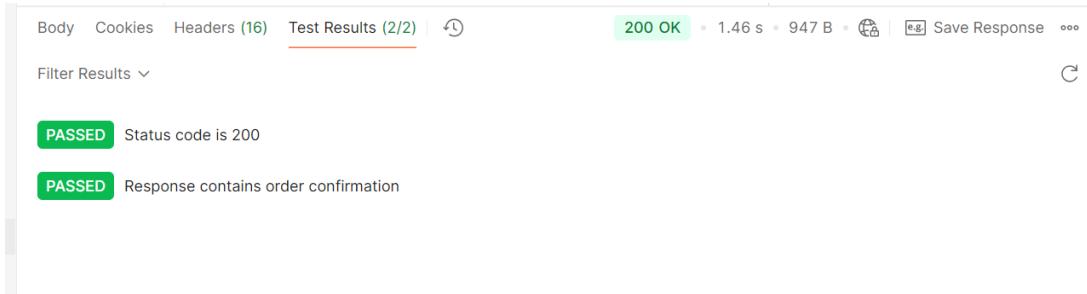
The response contains a unique orderConfirmation ID.

The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** `{{BaseUrl_js}}rest/basket/6/checkout`
- Auth Type:** Bearer Token
- Authorization Header:** eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.e ...
- Body:** JSON (empty)
- Headers:** (16) - includes Content-Type: application/json
- Test Results:** (2/2) - indicates success
- Status:** 200 OK
- Response Body:**

```
{ "orderConfirmation": "3008-f833cc85b8dd7729" }
```

Test Case Validation:



The screenshot shows a Postman test results interface. At the top, there are tabs for Body, Cookies, Headers (16), and Test Results (2/2). The Test Results tab is selected, showing a green '200 OK' status with a duration of 1.46 s and a response size of 947 B. Below the status, there are two green 'PASSED' status indicators: 'Status code is 200' and 'Response contains order confirmation'. There is also a 'Filter Results' dropdown and a 'Save Response' button.

✓ Passed Tests:

- Status code is 200 OK
- Response contains orderConfirmation field

✗ If failed:

- 401 Unauthorized (Missing or invalid authentication token)
- 404 Not Found (Invalid or non-existent basket ID)
- 409 Conflict (Basket is empty or checkout process failed)
- 500 Internal Server Error (Unexpected server issue)

3.9 Fetch Basket Details

Serial No.	09
Description	Retrieves details of a specific shopping basket
Endpoint	{{BaseUrl_js}}rest/basket/6
HTTP Method	GET
Auth	Bearer Token

Response Details

Expected Response (200 OK):

The response contains basket details, including id, UserId, and product details if available.

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** {{BaseUrl}}/rest/basket/6
- Pre-request Script:**

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});
```
- Test Results (1/1):** Status code: Code is 200, Response body: Contains string, Response body: JSON value check, Response body: Is equal to a string.
- Body:** JSON preview of the response object:{
 "status": "success",
 "data": {
 "id": 6,
 "coupon": null,
 "UserId": 22,
 "createdAt": "2025-02-22T06:19:33.735Z",
 "updatedAt": "2025-02-22T06:19:33.735Z",
 "Products": []
}

Test Case Validation:

The screenshot shows the Postman test results page with the following details:

- Test Results (1/1):** PASSED Status code is 200
- Status:** 200 OK
- Time:** 987 ms
- Size:** 1.67 KB
- Actions:** Save Response, Copy

✓ Passed Tests:

- Status code is 200 OK
- Response contains id, UserId, and Products fields

✗ If failed:

- 401 Unauthorized (Missing or invalid authentication token)
- 404 Not Found (Invalid or non-existent basket ID)
- 500 Internal Server Error (Unexpected server issue)

3.10 Update Product Review

Serial No.	10
Description	Updates the review for a specific product
Endpoint	<code>{{BaseUrl_js}}rest/products/29/reviews</code>
HTTP Method	PUT
Auth	Bearer Token

Request Body:

The screenshot shows the Postman interface with the following details:

- HTTP Method:** PUT
- URL:** `{{BaseUrl_js}} rest/products/29/reviews`
- Headers:** (10)
- Body:** (raw JSON)
- Body Content:**

```
1 {"message": "100% pure juice made from the flesh of fresh fruit or from whole fruit",
  "author": "admin123@gmail.com"
2 }
```
- Send button:** A large blue button at the top right of the request area.

Response Details:

Expected Response (201 Created):

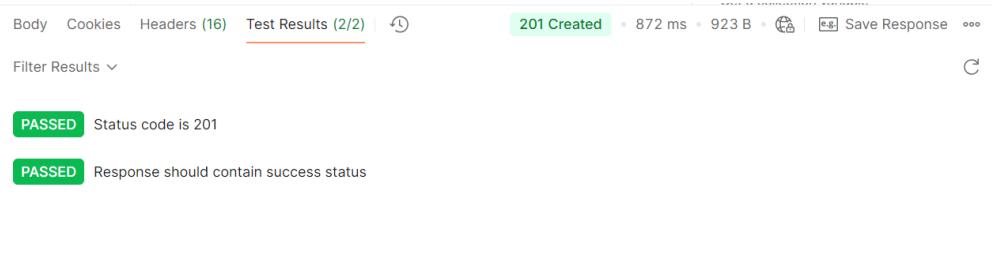
Indicates the product review was successfully updated

The screenshot shows the Postman interface displaying a successful response:

- Status:** 201 Created
- Time:** 872 ms
- Size:** 923 B
- Save Response:** A button to save the response.
- Body Content:**

```
1 {
  2   "status": "success"
3 }
```

Test Case Validation:



PASSED Status code is 201

PASSED Response should contain success status

✓ Passed Tests:

- Status code is 201 Created
- Response contains "status": "success"

✗ If failed:

- 400 Bad Request (Invalid JSON payload)
- 401 Unauthorized (Missing or invalid authentication token)
- 404 Not Found (Invalid or non-existent product ID)
- 500 Internal Server Error (Unexpected server issue)

3.11 Fetch Product Reviews

Serial No.	11
Description	Fetches the reviews for a specific product
Endpoint	{{BaseUrl_js}}rest/products/29/reviews
HTTP Method	GET
Auth	Bearer Token

Response Details:

Expected Response (200 OK):

Indicates that the product reviews were successfully retrieved.

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** {{BaseUrl}}/rest/products/29/reviews
- Pre-request Script:** pm.test("Status code is 200", function () { pm.response.to.have.status(200); });
- Test Results:** 1/1 (Passed)
- Status:** 200 OK
- Body (JSON):**

```
1 {
2   "status": "success",
3   "data": [
4     {
5       "product": "29",
6       "message": "100% pure juice made from the flesh of fresh fruit or from whole fruit",
7       "author": "admin123@gmail.com",
8       "likesCount": 0,
9       "likedBy": [],
10      "_id": "QXhbHLuv3oGKDqZg7",
11      "liked": true
12    }
13  ]
14 }
```

Test Case Validation:

The screenshot shows the Postman test results page with the following details:

- Test Results:** 1/1 (Passed)
- Status:** 200 OK
- Body:** PASSED Status code is 200

✓ Passed Tests:

- **Status code is 200 OK**
- **Response contains "status": "success"**
- **Response contains at least one review**

✗ If failed:

- **400 Bad Request** (Invalid request parameters)
- **401 Unauthorized** (Missing or invalid authentication token)
- **404 Not Found** (No reviews found for the given product ID)

- **500 Internal Server Error** (Unexpected server issue)

3.12 Retrieve Shopping Cart Info

Serial No.	12
Description	Retrieves the shopping cart (basket) details for a specific user
Endpoint	<code>{{BaseUrl_js}}rest/basket/6</code>
HTTP Method	<code>GET</code>
Auth	Bearer Token

Response Details:

Expected Response (200 OK):

Indicates that the shopping cart was successfully retrieved and is currently empty.

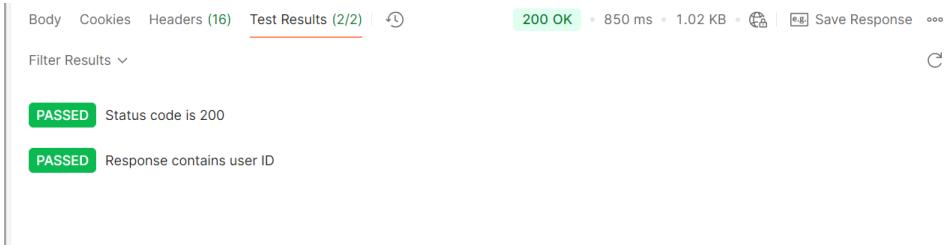
The screenshot shows the Postman application interface. The request URL is `GET {{BaseUrl_js}}rest/basket/6`. The 'Authorization' tab is selected, showing 'Bearer Token' and a token value. The response status is 200 OK, and the JSON body is displayed as follows:

```

1  {
2   "status": "success",
3   "data": {
4     "id": 6,
5     "coupon": null,
6     "UserId": 22,
7     "createdAt": "2025-02-22T13:35:52.662Z",
8     "updatedAt": "2025-02-22T13:35:52.662Z",
9     "Products": []
10    }
11  }

```

Test Case Validation:



The screenshot shows a test results summary from a tool like Postman or similar. At the top, there are tabs for 'Body', 'Cookies', 'Headers (16)', 'Test Results (2/2)', and a refresh icon. To the right, it displays '200 OK' status, response time '850 ms', size '1.02 KB', and icons for copy, save response, and more. Below the tabs, there's a 'Filter Results' dropdown. The main area lists two test cases under the 'PASSED' category: 'Status code is 200' and 'Response contains user ID'.

✓ Passed Tests:

- **Status code is 200 OK**
- **Response contains "status": "success"**
- **Response contains a UserId field**

✗ If failed:

- **400 Bad Request** (Invalid request parameters)
- **401 Unauthorized** (Missing or invalid authentication token)
- **403 Forbidden** (User does not have permission to access the basket)
- **404 Not Found** (Basket ID does not exist)
- **500 Internal Server Error** (Unexpected server issue)

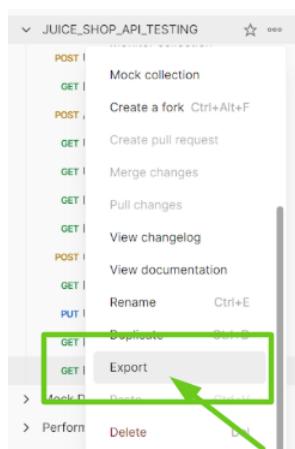
Chapter 4. API Automation Testing with Postman & Newman

In this section, we will set up Postman with Newman for API automation, develop automation scripts, and execute them.

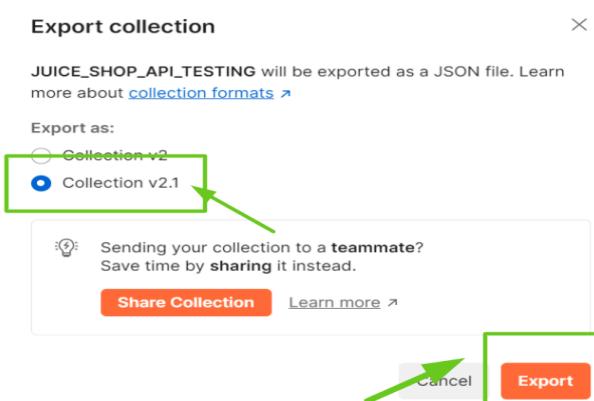
Step 4.1: Export the Postman Collection

Follow the steps in the provided images to export the **JUICE_SHOP_API_TESTING** collection:

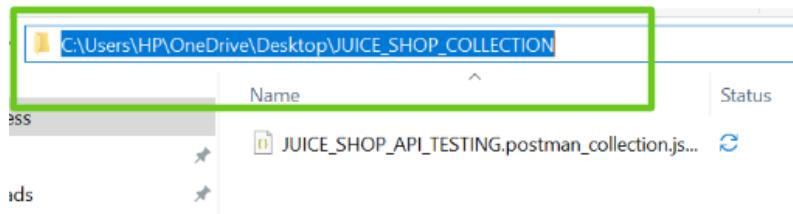
1. Open Postman and navigate to your collection.
2. Click on the three-dot menu next to the collection name.
3. Select Export from the dropdown menu.



4. Choose Collection v2.1 format and click Export.



5. Save the JSON file to a suitable location C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION.



Step 4.2: Install Newman

Newman is a CLI tool that allows you to run Postman collections via command line, making it ideal for CI/CD automation.

1. Ensure Node.js is installed

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION>node -v
v22.14.0
```

A screenshot of a Windows Command Prompt window. The title bar says 'C:\Windows\System32\cmd.exe'. The window displays the output of the command 'node -v', which returns 'v22.14.0'. The background of the window is black.

2. Install Newman globally using npm

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION>npm install -g newman
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated uidg@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm warn deprecated @faker-js/faker@5.5.3: Please update to a newer version.

changed 120 packages in 42s

10 packages are looking for funding
  run `npm fund` for details
```

A screenshot of a Windows Command Prompt window. The title bar says 'C:\Windows\System32\cmd.exe'. The window displays the output of the command 'npm install -g newman'. A red arrow points to the command line where the command is being typed. The background of the window is black.

Step 4.3: Run API Tests using Newman

Now, execute the exported Postman collection in the command line.

1. Open Command Prompt (CMD) or Terminal.
2. Navigate to the folder where the collection is saved
3. Run the collection with Newman

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd "C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION"

C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION>newman run JUICE_SHOP_API_TESTING.postman_collection.json
newman

JUICE_SHOP_API_TESTING
```

4.Expected Output:

- If all API tests pass, we will see a summary report with test results.

	executed	failed
iterations	1	0
requests	12	0
test-scripts	24	0
prerequest-scripts	13	0
assertions	19	15
total run duration: 4.5s		
total data received: 626B (approx)		
average response time: 262ms [min: 192ms, max: 953ms, s.d.: 208ms]		

Step 4.4: Generate an HTML Report

In this step, we will generate a detailed HTML report for the API tests using Newman and the htmlextra reporter. The screenshots provided show the process in action.

1. Install the Newman HTML Extra Reporter

Before generating an HTML report, install the newman-reporter-htmlextra package.

Run the following command in CMD:

```
C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION>npm install -g newman-reporter-htmlextra
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated uid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm warn deprecated @faker-js/faker@5.5.3: Please update to a newer version.

changed 166 packages in 23s
14 packages are looking for funding
  run 'npm fund' for details
```

2. Run the Postman Collection with HTML Reporting

Now, execute the Postman collection using Newman with the htmlextra reporter.

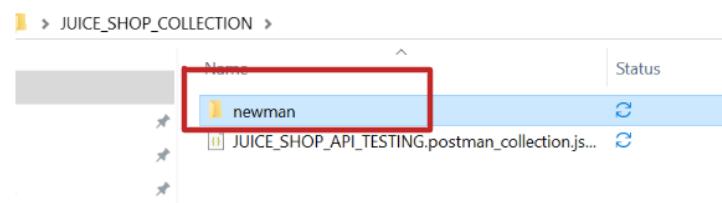
Run the following command:

```
run npm fund for details
C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION>newman run "JUICE_SHOP_API_TESTING.postman_collection.json" -r htmlextra
C:\Users\HP\OneDrive\Desktop\JUICE_SHOP_COLLECTION>
```

- This command runs the collection and generates an HTML report.
- Confirms that the command has been executed successfully.

3. Locate the Generated Report

- The report will be saved inside a new "newman" folder in the same directory where the command was executed.

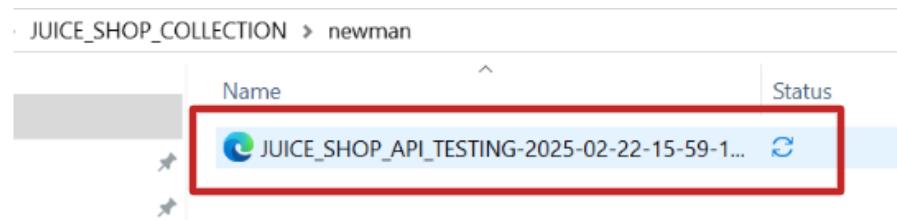


Step 4.5: Open the HTML Report

Now that we have executed the API tests using Newman with the `htmlextra` reporter, let's view the generated HTML report.

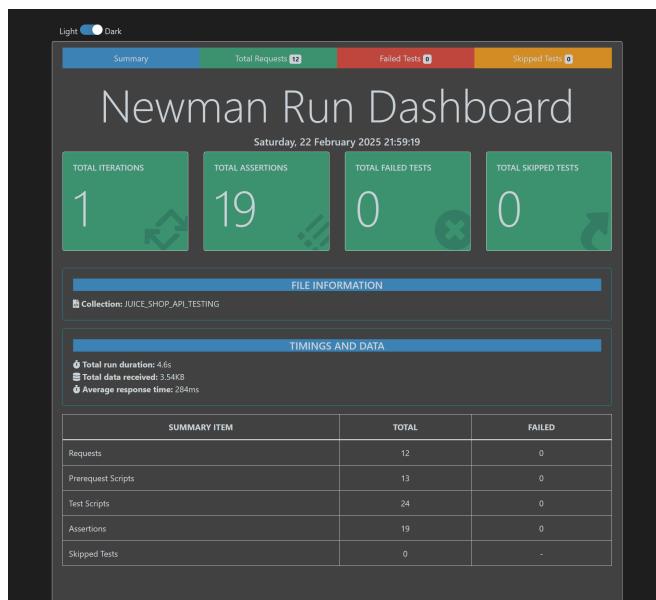
1. Locate the Report

- Navigate to the `newman` folder inside the test directory.
- In the first image, we can see the generated report file:
[JUICE_SHOP_API_TESTING-2025-02-22-15-59-19-874-0.html](#)
- The file is 279 KB in size.



2. Open the Report in a Browser

- Double-click the HTML file to open it in a browser.
- It can be viewed in Edge, Chrome, Firefox, or any other modern browser.
- The image confirms that the report is successfully opened.



3. Analyze the Report

The Newman Run Dashboard provides a visual summary of the test execution:

- **Test Execution Summary**
 - Total Iterations: 1
 - Total Requests Executed: 12
 - Total Assertions: 19
 - Total Failed Tests: 0
 - Total Skipped Tests: 0
- **File Information**
 - Collection: JUICE_SHOP_API_TESTING
- **Timings and Data**
 - Total Run Duration: 4.6 seconds
 - Total Data Received: 3.54 KB
 - Average Response Time: 284ms

4. Detailed Breakdown of API Test Execution

Key Observations:

- Total Requests Executed: 12
 - Failed Tests: 0 (All requests passed successfully)
 - Skipped Tests: 0

Each request tested different API functionalities related to the JUICE_SHOP API, including:

The screenshot shows the Postman interface in dark mode. At the top, there are buttons for 'Light' and 'Dark'. Below that is a summary bar with 'Total Requests 12', 'Failed Tests 0', and 'Skipped Tests 0'. A green button labeled '1' indicates '1 ITERATION AVAILABLE TO VIEW'. The main area shows a list of requests under 'ITERATION 1 SELECTED': Iteration: 1 - User Login, Iteration: 1 - Fetch Product List, Iteration: 1 - Address Create, Iteration: 1 - User Address Details, Iteration: 1 - User Wallet Balance, Iteration: 1 - Fetch Delivery Info, Iteration: 1 - Fetch Saved Cards, Iteration: 1 - Checkout Order, Iteration: 1 - Fetch Basket Details, Iteration: 1 - Updated Product Review, Iteration: 1 - Product Reviews, and Iteration: 1 - Retrieve Shopping Cart Info. Each item has a small arrow icon to its right.

1. User Login:

This screenshot shows a detailed view of the 'Iteration: 1 - User Login' request. At the top, it says '1 ITERATION AVAILABLE TO VIEW' and 'ITERATION 1 SELECTED'. The request information panel shows: Request Method: POST, Request URL: https://juice-shop.herokuapp.com/test/user/login. The response information panel shows: Response Code: 200 - OK, Mean time per request: 1062ms, Mean size per request: 514B. Below these, a 'TEST PASS PERCENTAGE' bar is at 100%. The bottom section is titled 'REQUEST HEADERS' and lists the following:

Header Name	Header Value
Content-Type	application/json
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	5bf232d1-9e4a-440a-9d47-da2ca8eba4fb
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Content-Length	55

REQUEST HEADERS	
Header Name	Header Value
Content-Type	application/json
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	5bf232d1-9e4a-440a-9d47-da2ca8eba4fb
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Content-Length	55

REQUEST BODY

```
{  
  "email": "admin123@gmail.com",  
  "password": "user123"  
}
```

[Copy to Clipboard](#)

Response Headers	
Header Name	Header Value
Server	Cowboy
Report-To	[{"group": "heroku-nel", "max_age": 3600, "endpoints": [{"url": "https://nel.herokuapp.com/reports?ts=1740239957&sid=812dcc77-0bd0-43b1-a5f1-b257503829598cs=1y6qjpQ4M3s5EuHsALuzFnqFMSM0A4mH7kLk8DFhL4%3D"}]]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239957&sid=812dcc77-0bd0-43b1-a5f1-b257503829598cs=1y6qjpQ4M3s5EuHsALuzFnqFMSM0A4mH7kLk8DFhL4%3D
Nel	{"report_to": "heroku-nel", "max_age": 3600, "success_fraction": 0.005, "failure_fraction": 0.05, "response_headers": ["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	814
Etag	W/"32e-eUKQID2AT54b0ZVKeLzNvItyVc"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:17 GMT
Via	1.1 vegur

RESPONSE BODY

```
{ "authentication": { "token": "eyJhbGciOiJIUzI1NiJ9.CjN0c101JSU1IiHj9.e7zdgF80Xm0lJz10M0JZANLriwIzgP07Sikey3pCC18f063hv1ZAUW11j011iwiDw", "bid": 1, "uemail": "admin123@gmail.com" } }
```

[Copy to Clipboard](#)

TEST INFORMATION

Name	Passed	Failed	Skipped
Status code is 200	1	0	0
Total	1	0	0

2. Fetching Product List:

Iteration: 1 - Fetch Product List

REQUEST INFORMATION

- ➊ Request Method: `GET`
- ➋ Request URL: <https://juice-shop.herokuapp.com/rest/basket/6>

RESPONSE INFORMATION

- ➊ Response Code: `200 - OK`
- ➋ Mean time per request: `213ms`
- ➌ Mean size per request: `154B`

TEST PASS PERCENTAGE

100 %

REQUEST HEADERS	
Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZC16MzMzMsInVzZXJuYW1ljoiiwiZW1haWwiOiJhZG1pbjEyM0BnbWFpbC5jb20iLCJwYXNzd29yZC16jZnZDE0mE5OTg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNmiwiwm9sZSI6ImN1c3RvbWVylowiZGVsdXhIVG9rZW4iOiiLCJsYXN0TG9naw5jcC6lnVuZGVmaW5jZCisnByb2ZpbGVbWFnZS16i9hc3NidHMvchVibGjL2ltYWdlcy91GxvYWRzL2RIZmF1bHQuc3ZnlwidG90cFNlY3JldCl6iilsmIzQWN0aXZljp0cnVlLCJcmVhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE5OjE0LjY4MCArMDA6MDAiLCJzZWxidGVkQXQiOm51bGx9LCjlpYXQiOjE3NDAyMzc4NzR9.o8RNMGjRe3Tg9CN9787sNwLKWClYTt6zI14WoXYzyw9fwrKncHq7jk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3h2zj6-fQWnCDC9-jyYtXcd1C_A7kvnct1YueystObEGMc-XosNxP-o6m_TLFEnVhVtaPahVmsRxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	9c3ddb1b-38ee-46d0-896c-725fdf6b1d41
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group": "heroku-nel", "max_age": 3600, "endpoints": [{"url": "https://nel.herokuapp.com/reports?ts=1740239957&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=1y6qjpQ4M3s5EluHsALrzFnqFMSMOA4mH7kLk8DFhL4%3D"}]}
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239957&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=1y6qjpQ4M3s5EluHsALrzFnqFMSMOA4mH7kLk8DFhL4%3D
Nel	{"report_to": "heroku-nel", "max_age": 3600, "success_fraction": 0.005, "failure_fraction": 0.05, "response_headers": ["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	154
Etag	W/"9a-yZlZhSJ9zjrvLISNY2h89gxTmoM"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:17 GMT
Via	1.1 vegur

RESPONSE BODY

```
{
  "status": "success",
  "data": {
    "id": 6,
    "coupon": null,
    "UserId": 22,
    "createdAt": "2025-02-22T13:05:52.662Z",
    "updatedAt": "2025-02-22T13:05:52.662Z",
    "Products": []
  }
}
```

[Copy to Clipboard](#)

TEST INFORMATION

Name	Passed	Failed	Skipped
Status code is 200	1	0	0
Total	1	0	0

3. Creating Address

Iteration: 1 - Address Create

REQUEST INFORMATION		RESPONSE INFORMATION	
Request Method: POST Request URL: https://juice-shop.herokuapp.com/api/Address/	Response Code: 201 - Created Mean time per request: 268ms Mean size per request: 288B	TEST PASS PERCENTAGE 100 %	
REQUEST HEADERS			
Header Name	Header Value		
Content-Type	application/json		
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzIwiZGF0YSI6eyJpZC16MzMslnVzZXJuYWl1ljoiliwiZW1haWwiOiJhZG1pbjEyM0BnbWFpbC5jb20lClJwYXNzd29yZC16jZhZDE0YmE5OTg2ZTM2MTU0MjNkZmNmMjU2ZDA0ZTNmlwicm9sZSl6lmN1c3RvbWVyliriZGVsdXhlVG9rZW4iOiiLCjsYXNOTG9naW5jcCl6lnVuZGVmaW5jZCIsInByb2ZpbGVjbWFnZSl6l9hc3NldHMvcHViibGjl2ltYWdlcy91cGxvYWRzL2RlZmf1bhQuc3NliwidG90cFIVY3JldC16ilsImlzQWN0aXZljp0cnVILCjcmVhdGvkQXQjOilyMDI1LTAyLTlyDE10je4OjA1JyA1NCArMDA6MDAiLCJcRhdkGvkQXQjOilyMDI1LTAyLTlyDE10je0LjY4MCArMDA6MDAiLCJkZWxldGvkQXQjOm51bGx9LCJpYXQjOjE3NDAyMzc4NzR9.o8RNMGjRe3Tg9CN9787sNwLKWClayTt6z114WoX7yw9fwKncHq7jk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3hZj6-iQWnCDC9-JyYtcd1C_A7kvnt1YueystObEGMc-XosNxP-o6m_TLFEnVhVtaPahVmsRxxtCQ		
User-Agent	PostmanRuntime/7.39.1		
Accept	/*		
Cache-Control	no-cache		
Postman-Token	721af742-e56d-4d5f-b62e-79e71912619b		
Host	juice-shop.herokuapp.com		
Accept-Encoding	gzip, deflate, br		
Connection	keep-alive		
Content-Length	201		

REQUEST BODY

```
{
  "country": "Bangladesh",
  "fullName": "Liza Akter",
  "mobileNum": "1713805993",
  "zipCode": "Dhaka12",
  "streetAddress": "Uttara Azampur",
  "city": "Dhaka, Bangladesh",
  "state": ""
}
```

Copy to Clipboard

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239957&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=1y6qJpQ4M3s5EuHsALrzFnqFMSM0A4mH7kLk8DFhL4%3D"}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239957&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=1y6qJpQ4M3s5EuHsALrzFnqFMSM0A4mH7kLk8DFhL4%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Location	/api/Addresss/11
Content-Type	application/json; charset=utf-8
Content-Length	288
Etag	W/"120-yK0TtoFPZ4YCESNjFx1WpDpB/5Y"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:17 GMT
Via	1.1 vegur

RESPONSE BODY			
<pre>{ "status": "success", "data": { "id": 1, "country": "Bangladesh", "fullName": "Liza Akter", "mobileNum": "1710065939", "zipCode": "100012", "streetAddress": "Uttara Azampur", "city": "Dhaka, Bangladesh", "state": "", "userId": 3, "updatedAt": "2025-02-22T15:59:17.099Z", "createdAt": "2025-02-22T15:59:17.099Z" } }</pre>			
Copy to Clipboard			
TEST INFORMATION			
Name	Passed	Failed	Skipped
Response contains required fields	1	0	0
Response contains submitted address	1	0	0
Status code is 201	1	0	0
Total	3	0	0

4. Fetching User Address Details

Iteration: 1 - User Address Details

REQUEST INFORMATION

- ① Request Method: **GET**
- ② Request URL: <https://juice-shop.herokuapp.com/api/Address>

RESPONSE INFORMATION

- ① Response Code: **200 - OK**
- ② Mean time per request: **211ms**
- ③ Mean size per request: **1.05KB**

TEST PASS PERCENTAGE

100 %

REQUEST HEADERS

Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzIwiZGF0YSI6eyJpZC16MzMslnVzXJuYW1ljoilivZW1haWwiOihZG1pbjEyM08nbWFpbC5jb20iCiwyXNzd29yZC16ijZhZD50VmE50Tg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNmlwicm9sZSl6mN1c3RvbWVylwzGVsdXHIVG9jZV4iOitlCjSXNOTG9naW5jCjClnVuZGvmaW5jZC1slnBybzZpbGVbWFnZ5l6i9hc3NidhMvchVlbGjj2lWYdly91cGxvYWRzL2RlZmf1bHQuc32nlivid90cFNIY3jdC16ilsImtzQWN0aXZljp0cnVLCjcmVhdGVkOXQjOilyMD11LTayLTlyIDE10jE4OjA1LjA1NCArMDA6MDAiLC1cGRndGVkOXQjOilyMD11LTayLTlyIDE10jE5OjE0Lj4MCArMDA6MDAiCjKzWxlGVkOXQjOm51bGx9LClpYXQiOjE3NDAyMz24NzR9.o8RNRMjkeTg9CN9787sNwLKWClaYTt6z14WoXYzy9fwKncHq7jk1V0ppr81sX9_kcc76DbYJ8w-JbQ_ttfGtNw2h2zjg6-iQVnCDC9-jyYtxcd1C_A7kvnt1YiueystObEGMc-XosNxp-o6m_TLFEnVhVtaPahVmsRxxtCQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	eae3a589-7ee4-4688-8c59-e9f99a71394f
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239958&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=q04SUxrLfu5QkW4Ifs22%2FhwKE8J9s0NU86SfxuGgPak%3D"}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239958&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=q04SUxrLfu5QkW4Ifs22%2FhwKE8J9s0NU86SfxuGgPak%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#jobs
Content-Type	application/json; charset=utf-8
Etag	W/"42f-kJ5HgddBjNYvuPpAhjqHV+99La0"
Vary	Accept-Encoding
Content-Encoding	gzip
Date	Sat, 22 Feb 2025 15:59:18 GMT
Transfer-Encoding	chunked
Via	1.1 vegur

RESPONSE BODY

```
{
  "status": "success",
  "data": [
    {
      "UserId": 31,
      "id": 9,
      "fullName": "Lisa Akter",
      "mobileNum": 1730805933,
      "zipCode": "Dhaka1111",
      "streetAddress": "Uttara Azampur",
      "city": "Dhaka, Bangladesh",
      "state": "",
      "country": "Bangladesh",
      "createdAt": "2025-02-20T15:25:02.016Z",
      "updatedAt": "2025-02-20T15:25:02.016Z"
    }
  ]
}
```

[Copy to Clipboard](#)

TEST INFORMATION

Name	Passed	Failed	Skipped
Response contains required fields	1	0	0
Response contains submitted address	1	0	0
Status code is 200	1	0	0
Total	3	0	0

5. Checking User Wallet Balance

Iteration: 1 - User Wallet Balance

REQUEST INFORMATION

- Request Method: **GET**
- Request URL: **https://juice-shop.herokuapp.com/test/wallet/balance**

RESPONSE INFORMATION

- Response Code: **200 - OK**
- Mean time per request: **203ms**
- Mean size per request: **288**

TEST PASS PERCENTAGE

100 %

Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MzMzMsInVzZXJuYW1ljoiliwzW1haWwiOjhZG1pbjEyM0BnbWFpbC5jb20iLCJwYXNzd29yZC16ijZhZDE0YmE5OTgZTMM2MTU0MjNlZmNhMjU2ZDA0ZTNmlwicm9sZSI6ImN1c3RvbWVylowiZGVsdXhIVG9rZW4iOiliLCJsYXN0TG9naW5jLcI6lnVuZGVmaW5lZlslnBybzZpbGVjbWFnZSI6i9hc3NldHMvchVibGjL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnlividG90cFNlY3JldC16ilismlzQWN0aXZljp0cnVlLCjcmVhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE4OjA1jA1NCArMDA6MDAiLCJ1cGRhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE5OjE0Ly4MCArMDA6MDAiLCjkZWxdGVkQXQiOm51bGx9LCJpYXQiOjE3NDAYMzc4NzR9.o8RMGjRe3Tg9CN9787snwLKWClavTf6z1I4WoXYzyw9frwKncHq7jgk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3h2j6-fQWrCDC9-JyYtXcd1C_A7kvnct1YiueystObEGMc-XosNXp-o6m_TLFEnVhVTaPahVmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	/*
Cache-Control	no-cache
Postman-Token	c12fd0f5-c78a-4d19-ae30-c9e19152b3e3
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group": "heroku-nel", "max_age": 3600, "endpoints": [{"url": "https://nel.herokuapp.com/reports?ts=1740239958&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=qO4SUxRLfU5QkW4Ifs22%2FHwKE8J9s0NU86SfxuGgPak%3D"}]}
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239958&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=qO4SUxRLfU5QkW4Ifs22%2FHwKE8J9s0NU86SfxuGgPak%3D
Nel	{"report_to": "heroku-nel", "max_age": 3600, "success_fraction": 0.005, "failure_fraction": 0.05, "response_headers": ["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	29
Etag	W/"1d-iBumrMzcXUGjLqkR74S/6br4l/M"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:18 GMT
Via	1.1 vegur

RESPONSE BODY			
{ "status": "success", "data": 0 }			
Copy to Clipboard			
TEST INFORMATION			
Name	Passed	Failed	Skipped
Status code is 200	1	0	0
Total	1	0	0

6. Fetching Delivery Info

Iteration: 1 - Fetch Delivery Info

REQUEST INFORMATION

- Request Method: **GET**
- Request URL: <https://juice-shop.herokuapp.com/api/Deliveries/1>

RESPONSE INFORMATION

- Response Code: **200 - OK**
- Mean time per request: **197ms**
- Mean size per request: **106B**

TEST PASS PERCENTAGE

100 %

REQUEST HEADERS	
Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzliwZGF0YSl6eyJpZCI6MzMslnVzZXJuYWl1ljoilivZWT1haWviOjhZG1pbjEyM08nbWFpbC5jb20iLClwYXNzd29yZC16jZhZDE0mE5OTg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNmlivid9szSI6imN1c3RvbWVyliviZGVsdXhlVG9rZW4iOiliLCjsYXNOTG9naW5jcCl6lnVuZGVmaW5IZClslnByb2ZpbGVjbWFnzSI6i9hc3NldHMvcHViGjL2ltYWdlcy91cGxvYWRzL2RlZmf1bHQuc3Znlivid90cFNIV3JldC16iislmzQWN0aXZljp0cnVILCjcmVhdGVkQXQiOilyMDI1LTAyLTtlyIDE10je4OjA1LjA1NCArMDA6MDAiLCJ1cGRhdGVkQXQiOilyMDI1LTAyLTtlyIDE10je4OjE0LjY4MCArMDA6MDAiLCJkZWxldGVkQXQiOm51bGx9LCjpxYXQiOjE3NDAYMzc4NzR9.o8RNMGjRe3Tg9CN9787sNwLKWClayTt6z1I4WoXzyw9fvrKnchq7jgk1V0ppn81sx9_Kcc76DByU8w-JbQ_tt6tNw3h2zj6-fQWnCDC9-JyYtXcd1C_A7kvnc1YiueystObEGMc-XosNXp-o6m_TLFEnVhVTaPahVmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	3e8e4828-ef6d-4ab3-b1a0-b54247a9e73e
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239958&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=qO4SUxRLfU5QkW4Ifs22%2FHwKE8J9s0NU86SfxuGgPak%3D"}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239958&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=qO4SUxRLfU5QkW4Ifs22%2FHwKE8J9s0NU86SfxuGgPak%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	106
Etag	W/"6a-nvvL7IVsoUqbYy++ZgqqM9CvGdQ"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:18 GMT
Via	1.1 vegur

RESPONSE BODY	
{ "status": "success", "data": { "id": 1, "name": "One Day Delivery", "price": 0.99, "eta": 1, "icon": "fas fa-rocket" } }	
Copy to Clipboard	

TEST INFORMATION						
Name	↑↓	Passed	↑↓	Failed	↑↓	Skipped
Status code is 200		1		0		0
Total		1		0		0

7. Fetching Saved Cards

Iteration: 1 - Fetch Saved Cards

REQUEST INFORMATION

- Request Method: **GET**
- Request URL: <https://juice-shop.herokuapp.com/api/cards>

RESPONSE INFORMATION

- Response Code: **200 - OK**
- Mean time per request: **195ms**
- Mean size per request: **30B**

TEST PASS PERCENTAGE

100 %

REQUEST HEADERS	
Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzliwiZGF0YSI6eyJpZCI6MzMslnVzZXJuYW1lIjoiliwizW1haWwiOjhZG1pbjEyM0BnbWFpbC5jb20iLCjwYXNzd29yZCI6jZhZDE0YmE5OTg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNmliwicm9sZSI6ImN1c3RvbWVyliziZGVsdXhIVG9rZW4iOiiLCJsXNOTG9naW5jCl6lnVuZGVmaW5iZCIsInByb2ZpbGVjbWFrnZS16i9hc3NldHMvcHvbGljL2tYWdlcy91cGxvYWRzL2RIZmF1bHQuc3NliwidG90cFNlY3JldCl6liislmlzQWN0aXZljp0cnVILCjcmVhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE4OjA1LjA1NCArMDA6MDAiLC1cGRhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE5OjE0LjY4MCArMDA6MDAiLCjzWxldGVkQXQiOm51bgxLCljpxXQjOjE3NDAYMzc4NzR9.08RNMGjRe3Tg9CN9787sNwlKWClaYTt6z114WoXYzyw9fwrKncHq7jgk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3h2zj6-fQWnCDC9-JyYtXcd1C_A7kvmtct1YiueystObEGMc-XosNxP-o6m_TLFEEnVhVtaPahVmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	8e09b709-42c5-4c5e-b2fd-b592978d989d
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239959&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=E3dxplPUs0KCKF1ZPs0GF93nyguNTg77%2BMWhjqcG1oU%3D"}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239959&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=E3dxplPUs0KCKF1ZPs0GF93nyguNTg77%2BMWhjqcG1oU%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	30
Etag	W/"1e-JkPcl+pGj7BBTxOuZTVVlm91zaY"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:19 GMT
Via	1.1 vegur

RESPONSE BODY			
{ "status": "success", "data": [] }			
Copy to Clipboard			
TEST INFORMATION			
Search:			
Name	Passed	Failed	Skipped
Status code is 200	1	0	0
Total	1	0	0

8. Checkout Order

Iteration: 1 - Checkout Order

REQUEST INFORMATION

- ① Request Method: **POST**
- ② Request URL: <https://juice-shop.herokuapp.com/rest/basket/6/checkout>

RESPONSE INFORMATION

- ① Response Code: **200 - OK**
- ② Mean time per request: **245ms**
- ③ Mean size per request: **45B**

TEST PASS PERCENTAGE
100 %

REQUEST HEADERS	
Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzIwiZGF0YSl6eyJpZCf6MzMlsInVzXJuYW1ljoiliwiZW1haWwOiJhZG1pbjEyM0BnbWFpbC5jb20iLCJwYXNzd29yZCf6ljZhZDE0YmE5OTg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNrmlwicm9sZSf6lmN1cRvbWVylowiZGVsdXhlVG9rZW4iOiiLCJsYXN0TG9naw5JcCl6lnVuZGVmaW5fZCIsInBybZpbGvJbWFrZSf6i9hc3NldHMvcHVibGjl2ltYWdlcgv1cGxvYWRzL2RlZmf1bHQuC3ZnlwidG90cFNIY3JldCl6iismlzQWN0aXZlIjp0cnVILCjcmVhdGVkQXQiOilyMDI1LTayLTlyIDE1OjE4OjA1JxA1NCArMDA6MDAiLC1cGrhdGVkQXQiOilyMDI1LTayLTlyIDE1OjE5OjE0LjY4MCArMDA6MDAiLCJzZWxldGVkQXQiOm51bGx9LCJpYXQiOjE3NDAyMzc4NzR9.o8RNMGjRe3tg9CN9787sNwLKClaYTt6z1k4WoXYzyw9fwrKnchq7jk1VOpnn81sX9_Kcc76DByU8w-JbQ_tt6tNw3hzj6-fQWnCDC9-JyYtXcd1C_A7kvnc1YiueystObEGMc-XosNXp-o6m_TLFEnVhVtPaPahVrmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	454cb473-b85a-4b9e-827a-2ca8874c4c74
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Content-Length	0

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239959&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=E3dxpIPUs0KCKF1ZPs0GF93nyguNTg77%2BMWHjqcG1oU%3D"}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239959&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=E3dxpIPUs0KCKF1ZPs0GF93nyguNTg77%2BMWHjqcG1oU%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	45
Etag	W/"2d-7+uFn3i0TGqqNC2/8MWoRowVO4"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:19 GMT
Via	1.1 vegur

RESPONSE BODY					
TEST INFORMATION					
Name	Passed	Failed	Skipped		
Response contains order confirmation	1	0	0		
Status code is 200	1	0	0		
Total	2	0	0		

9. Fetching Basket Details

Iteration: 1 - Fetch Basket Details

REQUEST INFORMATION

- Request Method: `GET`
- Request URL: <https://juice-shop.herokuapp.com/rest/basket/6>

RESPONSE INFORMATION

- Response Code: `200 - OK`
- Mean time per request: `211ms`
- Mean size per request: `154B`

TEST PASS PERCENTAGE
100 %

Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIwiZGF0YSI6eyJpZCI6MzMslnVzZXJuYW1ljoiliwiZW1haWwiOjhZG1pbjEyM0BnbWFpbC5jb20iLCJwYXNzd29yZCI6ljZhZDE0YmE5OTg2ZTM2MTU0MjNkZmNmMjU2ZDA0ZTNmlwiMc9sZSl6lmN1c3RvbWVylwiZGVsdXhlVG9rZW4iOiliLCJsYXN0TG9naW5jcCl6lnVuZGVmaW5jZCl6lnBy22zbGVjbWFnZSl6i9hc3NldHMvcHVibCjlL2ltWdlc91cGxvYWRzL2RIZmF1bHQuc3ZnlwidG90cFNlY3JldCl6lilsmIzQWN0aXZljjp0cnVILCjcmVhdGVkQXQjOilyMDI1LTayLTlyDE1OjE4OjA1Lja1NCArMDA6MDAiLCJ1cGRhdGVkQXQjOilyMDI1LTayLTlyDE1OjE5OjE0Lj4MCArMDA6MDAiLCjZWxldGVkQXQjOm51bgx9LCjpyXXQjOjE3NDAyMzc4NzR9.c8RNMGjRe3tg9CN9787sNwLKCjlaYTt6z1I4WoXYzyw9frwKncHq7jgk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3h2zj6-fQWnCDC9-JyYtXcd1C_A7kvnct1YueystObEGMc-XosNXp-o6m_TLFEnVhVtaPahVmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	7ecea2df-2b65-4993-b3f0-f7391a7cc92c
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239959&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=E3dxplIPUs0KCKF1ZPs0GF93nyguNTg77%2BMWHjqcG1oU%3D"]}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239959&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=E3dxplIPUs0KCKF1ZPs0GF93nyguNTg77%2BMWHjqcG1oU%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	154
Etag	W/"9a-yZJZhSJ9zjrvLISNY2h89gxTmoM"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:19 GMT
Via	1.1 vegur

RESPONSE BODY	
{	"status": "success", "data": { "id": 6, "coupon": null, "UserId": 22, "createdAt": "2025-02-22T13:35:52.662Z", "updatedAt": "2025-02-22T13:35:52.662Z", "Products": [] } }
<button>Copy to Clipboard</button>	

TEST INFORMATION						
Name	Passed	Failed	Skipped			
Status code is 200	1	0	0			
Total	1	0	0			

10. Updating Product Review

Iteration: 1 - Updated Product Review

REQUEST INFORMATION

- Request Method: PUT
- Request URL: <https://juice-shop.herokuapp.com/rest/products/29/reviews>

RESPONSE INFORMATION

- Response Code: 201 - Created
- Mean time per request: 207ms
- Mean size per request: 20B

TEST PASS PERCENTAGE
100 %

REQUEST HEADERS	
Header Name	Header Value
Content-Type	application/json
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzIwiZGF0YSI6eyJpZC16MzMslnVzZXJuYW1ljoiliwiZW1haWwiOiJhZG1pbjEyM0BnbWFpbC5jb20iLCJwYXNzd29yZC16jZhZDE0YmE5OTg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNmlwicm9sZS16ImN1c3RvbWVylwiZGVsdXhIVG9rZW4iOiliLCJsYXN0TG9naW5jLc16lnVuZGVmaW5jZC1sInByb2ZpbGVjbWFnZS16i9hc3NldHMvcHViBgJL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQc3ZnlwidG90cFNlY3JldC16iislmIzQWN0aXZIjp0cnVILCJcmVhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE5OjE0LjY4MCArMDA6MDAiLCJzZWxdGVkQXQiOm51bGx9LCJpYXQiOjE3NDAYMzc4NzR9.o8RNMGjRe3Tg9CN9787sNwLKClaYTt6z1I4WoXYzyw9fwrKncHq7jgk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3h2zj6-fQWnCDC9-JyYtXcd1C_A7kvnectYiueystObEGMc-XosNXp-o6m_TLFEnVhVTaPahVmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	/*
Cache-Control	no-cache
Postman-Token	1a1e834f-6285-4dae-99d5-8b1adc35f451
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br

REQUEST BODY

```
{
  "message": "100% pure juice made from the flesh of fresh fruit or from whole fruit",
  "author": "admin123@gmail.com"
}
```

Copy to Clipboard

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239960&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=y2HufDFxry%2FXiK%2Bn8tquDFS4PKcFWmWsrR42rdex7GM%3D"]}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239960&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=y2HufDFxry%2FXiK%2Bn8tquDFS4PKcFWmWsrR42rdex7GM%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	20
Etag	W/"14-Y53wuE/mmbSikKcT/WualL1N65U"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:20 GMT
Via	1.1 vegur

RESPONSE BODY			
{	"status": "success"		
<button>Copy to Clipboard</button>			
TEST INFORMATION			
Search: <input type="text"/>			
Name	Passed	Failed	Skipped
Response should contain success status	1	0	0
Status code is 201	1	0	0
Total	2	0	0

11. Fetching Product Reviews

Iteration: 1 - Product Reviews

REQUEST INFORMATION

- Request Method: `GET`
- Request URL: <https://juice-shop.herokuapp.com/rest/products/29/reviews>

RESPONSE INFORMATION

- Response Code: `200 - OK`
- Mean time per request: `203ms`
- Mean size per request: `765B`

TEST PASS PERCENTAGE

100 %

REQUEST HEADERS	
Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzliwiZGF0YSI6eyJpZCI6MzMlsInVzZXJuYW1ljoilwiZW1haWwOiIjZG1pbjEyM08nbWFpbC5jb20iLCwYXNzd29yZCI6ijZhZDE0YmE5OTg2ZTM2MTU0MjNkZmNmMjU2ZDA0ZTNmlivcm9zZl6lmN1c3RvbWVliwizGVsdxhIVG9rZW4iOiiLCjsXN0T69naW5JcCl6nVuZGVmaW5IZClslnByb2ZpbGVjbWFnZSI6i9hc3NldHMvcHViibGjI2lYWdlcy91cGxvYWRzL2RlZmf1bHQuc3nlwidG90cFNIY3Jldc16iisImlzQWN0aXZlJlp0cnVILCjcmVhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE4OjA1LjA1NCArMDA6MDAiLCI1cGRhdGVkQXQiOilyMDI1LTAyLTlyIDE1OjE5OjE0LjY4MCArMDA6MDAiLCjKzWxlGvkQXQiOm51bGx9LClpYXQiOjE3NDAyMzc4NzR9.o8RMIGjRe3Tg9CN9787sNwLKWClayTt6z1I4WoXYzyw9fwrKnchq7jk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3h2zj6-fQWnCDC9-JyTxcd1C_A7kvnc1YueystObEGMc-XosNXp-o6m_TLFEnVhVTaPahVmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	88e8cc59-8b80-42bc-8376-29e336308dfd
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239960&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=y2HufDFxry%2FXik%2Bn8tquDFS4PKcFWmWsrR42rdex7GM%3D"}]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239960&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=y2HufDFxry%2FXik%2Bn8tquDFS4PKcFWmWsrR42rdex7GM%3D
Nel	{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	765
Etag	W/"2fd-F4Bo0rtq9tXPaEkY56oj9KmUGs"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:20 GMT
Via	1.1 vegur

RESPONSE BODY

```
{
  "status": "Success",
  "data": [
    {
      "product": "29",
      "message": "100% pure juice made from the flesh of fresh fruit or from whole fruit",
      "author": "admin123@gmail.com",
      "likesCount": 0,
      "likedBy": [],
      "_id": "a5yRZF259c9c7ntd"
    },
    {
      "product": "29",
      "message": "100% pure juice made from the flesh of fresh fruit or from whole fruit",
      "author": "admin123@gmail.com",
      "likesCount": 0,
      "likedBy": []
    }
  ]
}
```

[Copy to Clipboard](#)

TEST INFORMATION

Name	Passed	Failed	Skipped
Status code is 200	1	0	0
Total	1	0	0

12. Retrieving Shopping Cart Info

Iteration: 1 - Retrieve Shopping Cart Info

REQUEST INFORMATION

- ① Request Method: `GET`
- ② Request URL: <https://juice-shop.herokuapp.com/rest/basket/6>

RESPONSE INFORMATION

- ① Response Code: `200 - OK`
- ② Mean time per request: `204ms`
- ③ Mean size per request: `154B`

TEST PASS PERCENTAGE

100 %

REQUEST HEADERS	
Header Name	Header Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzliwiZGF0YSI6eyJpZC16MzMsinVzZXJuYW1lIjoilivizZW1haWwvOihZG1pbjEyM0BnbWFpbCj5jb20iLClwXNzd29yZC16iZnZDE0YmE5OTg2ZTM2MTU0MjNkZmNhMjU2ZDA0ZTNmlwicm9sZSI6imN1c3RvbWVyliriZGVsdXhiVG9rZW4iOiiLCjsYXN0TG9naW5jcC6InVuZGVmaW5jZClnBybzZpbGVjbWFnzSI6i9hc3NldhIMvcHViGjl2ltWdly91GxvVWRzL2RiZmF1bHQuc3ZnlwidG90cFNlY3JdC16iismlzQWN0aXZljp0cnVILCjcmVhdGVkOXQiOilyMD1LTayLtyIDE1OjE5OjE0Lj4MCArMDA6MDAiLCjZWxdGVkQXQiOm51bGx9LClpYXQjOjE3NDAYMzc4NzR9.o8RNMGRe3Tg9CN9787NwlKWC1aYTtfz-1I4WoXYyw9fwrKnchq7jgk1VOppn81sX9_Kcc76DByU8w-JbQ_tt6tNw3h2zj6-fQWnCDC9-JyYtXcd1C_A7kvnc1YueystObEGMc-XosNxP-o6m_TLFEnVhVTaPahVmsRXxTcQ
User-Agent	PostmanRuntime/7.39.1
Accept	*/*
Cache-Control	no-cache
Postman-Token	619ef716-ea69-4387-86f6-b80f3cffb1e4
Host	juice-shop.herokuapp.com
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

RESPONSE HEADERS	
Header Name	Header Value
Server	Cowboy
Report-To	[{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.herokuapp.com/reports?ts=1740239960&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=y2HuFDxY%2FXiK%2Bn8tquDFS4PKcFWmWsrR42rdex7GM%3D"}]]
Reporting-Endpoints	heroku-nel=https://nel.herokuapp.com/reports?ts=1740239960&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=y2HuFDxY%2FXiK%2Bn8tquDFS4PKcFWmWsrR42rdex7GM%3D
Nel	[{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}]
Connection	keep-alive
Access-Control-Allow-Origin	*
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
Feature-Policy	payment 'self'
X-Recruiting	/#/jobs
Content-Type	application/json; charset=utf-8
Content-Length	154
Etag	W/"9a-yZJZhSJ9zjrvLISNY2h89gxTmoM"
Vary	Accept-Encoding
Date	Sat, 22 Feb 2025 15:59:20 GMT
Via	1.1 vegur

RESPONSE BODY			
{ "status": "success", "data": { "id": 6, "coupon": null, "userId": 22, "createdAt": "2025-02-22T13:35:52.662Z", "updatedAt": "2025-02-22T13:35:52.662Z", "Products": [] } }			
Copy to Clipboard			
TEST INFORMATION			
Name	Passed	Failed	Skipped
Response contains user ID	1	0	0
Status code is 200	1	0	0
Total	2	0	0

Chapter 5. Conclusion

The JUICE Shop API testing was successfully conducted using Postman and automated with Newman. All 12 API test cases passed with 0 failures, ensuring the stability and reliability of the system. The automation process significantly improved efficiency, reduced manual effort, and provided detailed reporting. Performance metrics showed fast execution (4.6 seconds total, 284ms average response time).

The API testing covered critical functionalities such as user authentication, product retrieval, basket management, order checkout, payment processing, and user account management. Automation testing with Newman CLI enabled efficient and repeatable execution of test cases, ensuring rapid validation and detailed HTML reporting using Newman htmlextra.

In addition to API testing, manual testing was conducted to evaluate functional, UI, usability, security, and performance aspects of the JUICE Shop application. A total of 52 test cases passed successfully, confirming the core functionalities work as expected. However, 8 test cases failed, highlighting issues related to pagination, security vulnerabilities, data export, and profile customization. The manual testing provided valuable insights into user experience, system behavior, and security concerns, which will help in future enhancements.

To further enhance testing, security testing, load testing, and CI/CD integration are recommended. Overall, both API and manual testing confirm that the system is stable, efficient, and production-ready, with scope for continuous improvement.