

# Testing

## Frontend

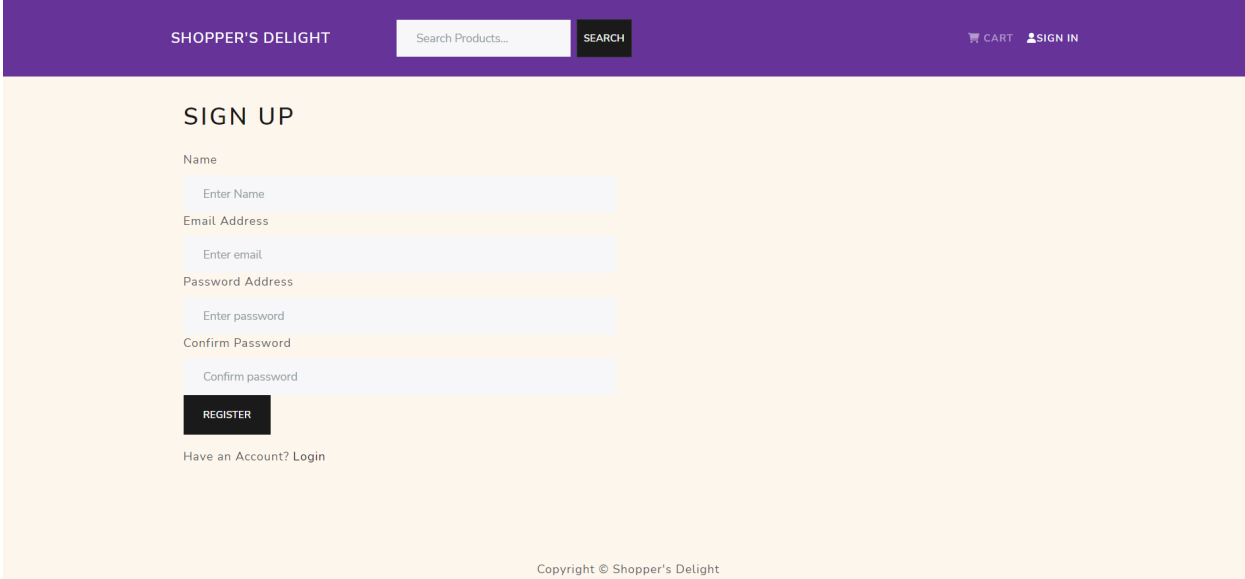
### 1. Sign Up

#### 1.1 Registration Form

Expectations: **It shall render a registration form.**

Result: **Passed**

Screen Shots/Logs:



The screenshot displays the 'SIGN UP' form on the Shopper's Delight website. The form is set against a light orange background. At the top, a purple navigation bar contains the site name 'SHOPPER'S DELIGHT', a search bar with the placeholder 'Search Products...' and a 'SEARCH' button, and links for 'CART' and 'SIGN IN'. The form itself includes input fields for 'Name', 'Email Address', 'Password Address', and 'Confirm Password', each with a corresponding placeholder text. A black 'REGISTER' button is positioned below the password fields. At the bottom of the form, there is a link that says 'Have an Account? Login'. The footer of the page, located at the bottom right, reads 'Copyright © Shopper's Delight'.

#### 1.2 Registering a new user

Expectations: **It shall show a success or error message.**

Result: **Passed**

## Screen Shots/Logs:

The screenshot shows a web browser window with a purple header bar containing the text "SHOPPER'S DELIGHT" and a search bar. Below the header is a "SIGN UP" section on a light orange background. The form includes fields for "Name" (filled with "John Doe"), "Email Address" (filled with "johndoe@gmail.com"), "Password Address" (filled with "\*\*\*\*\*"), and "Confirm Password" (filled with "\*\*\*\*\*"). A black "REGISTER" button is at the bottom of the form, followed by the text "Have an Account? Login". A dark grey error message box from Heroku is overlaid on the top right, stating "shoppers-delight.herokuapp.com says Only IIIT-Guwahati registered email ids are allowed" with an "OK" button.

## 2. On Logging in

Expectations: When Logging in with wrong credentials, it will show a failure message.

Result: Passed

Screen Shots/Logs

The screenshot shows the 'SIGN IN' page of 'SHOPPER'S DELIGHT'. At the top, there is a purple navigation bar with the site name, a search bar, and links for 'CART' and 'SIGN IN'. The main content area is light yellow. It features a 'SIGN IN' heading, a red error message 'Invalid email or password', and two input fields: 'Email Address' (containing 'aditya@iitg.ac.in') and 'Password Address' (containing five dots). A 'SIGN IN' button is below the fields, and a link 'New Customer? Register' is at the bottom.

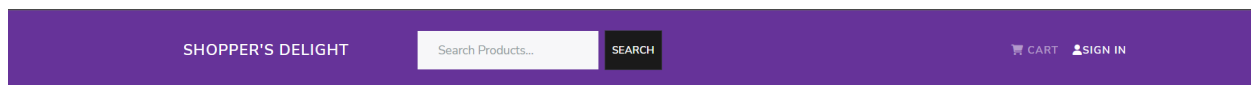
### 3. Navigation Bar

#### 3.1 Logged Out State

Expectations: Shall show Login/SignUp button.

Result: Passed

Screen Shots/Logs:

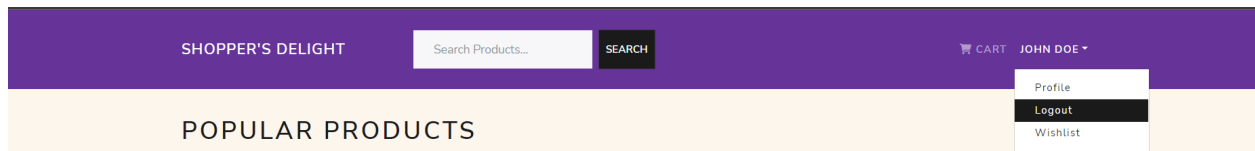


#### 3.2 Logged In State

Expectations: Shall show Navigation Links, Logged In User Details and Logout Button.

Result: Passed

Screen Shots/Logs:



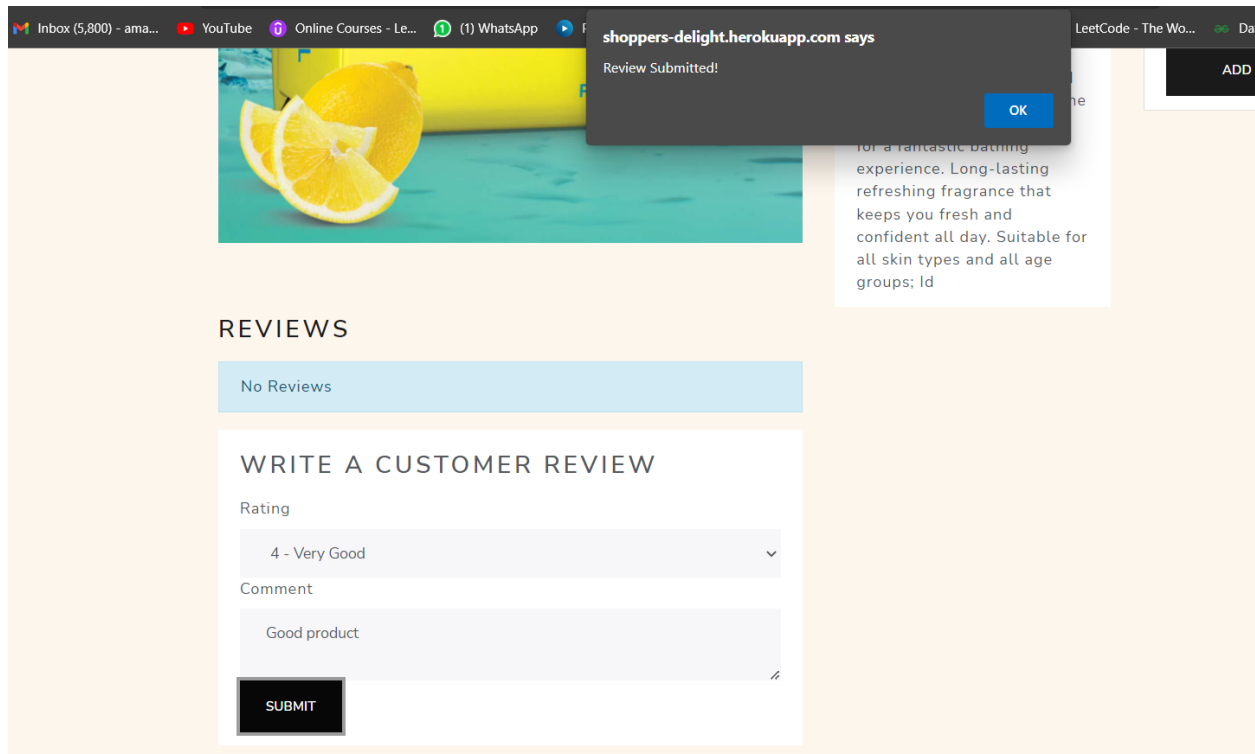
## 4. Reviews

### 4.1 Review submission

Expectations: Shall show a message saying review has been submitted

Result: Passed

Screen Shots/Logs:

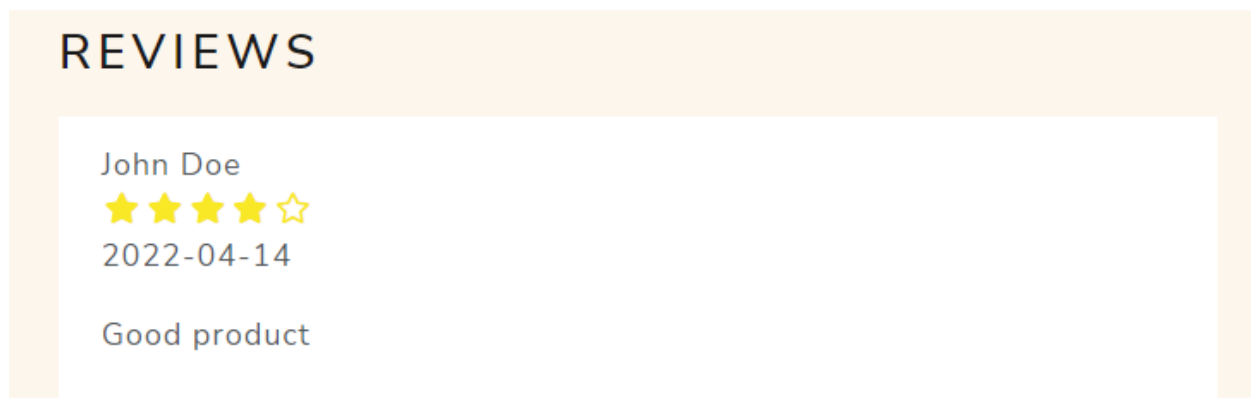


## 4.2 Review Display

Expectations: **Shall show the new posted review for the product**

Result: **Passed**

Screen Shots/Logs:



## 5. Payments

Expectations: Shall show payment successful after completing the payment with valid credentials

Result: Passed

Screen Shots/Logs:

### PAYMENT METHOD

Method: PayPal

Paid on 2022-04-14T14:19:58.119Z

# SYSTEM TESTING

## 1) USER REGISTRATION

TEST CASES	RESULT
User registration with new email id	Successful user registration
User registration with already registered email id	User already existing error
User registration with wrong email id	Enter valid email id

## 2) LOGIN

TEST CASES	RESULT
Login with correct email and password	Logged in Successfully
Login with wrong email and password	Wrong credentials error
Login with correct email and wrong password	Wrong credentials error

## 3) SEARCH PRODUCTS

TEST CASES	RESULT
------------	--------

Enters all the required field in search box	Displays search result based on location
Enters wrong information	No results shown and 'Go back' option is available.

#### 4) WISHLIST

TEST CASES	RESULT
Access wishlist	Wishlist opens
Click on 'Add to wishlist'	Product added to wishlist successfully
Click on the delete button	Item removed from wishlist successfully

#### 5) CART

TEST CASES	RESULT
Access cart without login	Asks to login
Access cart after login	Cart opens
Click on 'Add to cart'	Product added to cart successfully
Click on the delete button	Item removed from cart successfully

#### 6) PLACE ORDER



TEST CASES	RESULT
Click on checkout after adding all items in cart	Redirects to login page if not logged in. Else asks for shipping info.
Click on place order	Asks for payment mode
Select preferred mode of payment and pay	Order placed successfully
Order gets placed	Buyer receives an email confirming the order.

After running the above test cases we fixed all detected irregularities between the unit tests that are integrated together. We made sure that none of the features are breaking and the system as a whole is working smoothly in all circumstances .

## Manual Testing

Test accounts were created to test the flow of entire web application

### Customer account

- Searched for the product by typing in the navigation bar
- Browsed items over the pages
- Reviewed and added comments to products
- Added and removed products from wishlist
- Added and removed products from cart
- Order placed of the products
- Entered shipping details in a form
- Selected payment options as paypal or credit card

- Placed final order

### **Admin Account**

- Admin has access to all the users, products and orders
- Edited/Updated (by rendering admin status) information of a user
- Edited product information, removed product from the list
- Created a new product, by filling the details in the product form
- Viewed order details for all the orders placed by customers

By testing every feature available on the website based on the two types of account we ensured that the work flow of the website is smooth and free from errors and bugs.

### **Integration Testing**

We ran all unit test modules. We found that all the modules are loosely coupled and don't require separate integration testing.

To perform manual integration testing, we have individually tested all the modules and made sure that they are working properly in combination with each other.

## **Backend**

The backend testing was **white box testing**, where tests were written to ensure that data always returned in a predetermined format. We used npm library mocha and express to write automated tests. Outcome of our tests is as follows, each line shall show the endpoint called followed by the request name, description and latency of each API call.

C:\Windows\System32\cmd.exe

```
Testing User Routes
error already user
POST /api/users 405 40.518 ms - 248
  ✓ /api/users (46ms)
GET /api/users 200 85.951 ms - 2889
  ✓ /api/users (92ms)
POST /api/users/login 200 132.402 ms - 291
  ✓ /api/users/login (135ms)
GET /api/users/profile 200 88.179 ms - 97
  ✓ /api/users/profile (92ms)
PUT /api/users/profile 200 155.343 ms - 279
  ✓ /api/users/profile (161ms)
DELETE /api/users/625842a7ae07801f6b2834dc 404 77.933 ms - 239
  ✓ /api/users/:id (81ms)
GET /api/users/62310ea8eb1f5f85150449f5 200 80.138 ms - 181
  ✓ /api/users/:id (85ms)
PUT /api/users/62310ea8eb1f5f85150449f5 200 147.002 ms - 95
  ✓ /api/users/:id (153ms)
```

```
Testing Email Routes
  ✓ /api/email
POST /api/email 201 14.773 ms - 17
```

22 passing (3s)

D:\Sem 6\SE Project\New folder (3)>\_

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.593]
(c) Microsoft Corporation. All rights reserved.

D:\Sem 6\SE Project\New folder (3)>npm test

> se-project@1.0.0 test
> mocha --exit

Server running in development mode on port 5000

Testing Product Routes
MongoDB Conncted: clusterse-shard-00-01.tangq.mongodb.net
GET /api/products 200 1037.490 ms - 6937
  ✓ /api/products (1052ms)
GET /api/products/top 200 40.261 ms - 2539
  ✓ /api/products/top (43ms)
GET /api/products/623d98b0ca0cba3827b2dd22 200 40.938 ms - 816
  ✓ /api/products/:id (44ms)
DELETE /api/products/625842a7ae07801f6b2834dc 404 80.741 ms - 247
  ✓ /api/products/:id (88ms)
PUT /api/products/623d98b0ca0cba3827b2dd22 200 135.335 ms - 816
  ✓ /api/products/:id (139ms)
POST /api/products/623d98b0ca0cba3827b2dd22/reviews 400 81.165 ms - 262
  ✓ /api/products/:id/reviews (86ms)
POST /api/products 201 136.295 ms - 357
  ✓ /api/products/ (141ms)

Testing Order Routes
POST /api/orders 400 44.215 ms - 692
  ✓ /api/orders (50ms)
GET /api/orders 200 205.019 ms - 31306
  ✓ /api/orders (208ms)
GET /api/orders/myorders 200 78.572 ms - 947
  ✓ /api/orders/myorders (82ms)
GET /api/orders/623ae46d403558bf621ce926 200 120.898 ms - 794
  ✓ /api/orders/:id (125ms)
PUT /api/orders/623ae46d403558bf621ce926/pay 200 135.436 ms - 728
  ✓ /api/orders/:id/pay (139ms)
PUT /api/orders/623ae46d403558bf621ce926/deliver 200 124.875 ms - 738
  ✓ /api/orders/:id/deliver (128ms)

Testing User Routes
error already user
POST /api/users 405 40.518 ms - 248
  ✓ /api/users (46ms)
GET /api/users 200 85.951 ms - 2889
  ✓ /api/users (92ms)
POST /api/users/login 200 132.402 ms - 291
```