User Controller:

POST/api/User/Add-User

Test Case ID	Description	Input	Expected Result	Comments
1	Adding a user successfully	{"userName": "ahmed", "userEmail": "ahmed@gmail.com", "userPassword": "ahmed123", "userPhone": "78965422", "userRole": "admin"}	200: returning user details.	User details are returned for display except for password as it is hashed.
2	Add a user with a missing userName property	{"userName": "", "userEmail": "testing@gmail.com", "userPassword": "testing123", "userPhone": "78965422", "userRole": "admin"}	400: User name is required	An error is thrown to the user informing them that the name field is required.
3	Add a user with a missing userName property	{ "userName": "testing", "userEmail": "", "userPassword": "testing123", "userPhone": "78965422", "userRole": "admin"}	400: User Email is required	An error is thrown to the user informing them that the email field is required.
4	Add a user	{"userName":	400:	An error is

	with a missing password property	"testing", "userEmail": "testing@gmail.com", "userPassword": "", "userPhone": "78965422", "userRole": "admin"}	Password is required	thrown to the user informing them that the password field is required.
5	Add a user with a missing userPhone property	{ "userName": "testing", "userEmail": "testing@gmail.com", "userPassword": "testing123", "userPhone": "", "userRole": "admin"}	400: Phone number is required	An error is thrown to the user informing them that the Phone number field is required.
6	Add a user with a missing userRole property	{ "userName": "testing", "userEmail": "testing@gmail.com", "userPassword": "testing123", "userPhone": "78965422", "userRole": ""}	400: Role must be entered	An error is thrown to the user informing them that the role field is required.
7	Entering an invalid email pattern	{ "userName": "testing", "userEmail": "testing.gmail.com", "userPassword": "testing123", "userPhone": "78965422",	400: Email must be in this pattern: “example@example.com”	Showing the user the correct format of the email that must be entered.

		"userRole": "admin"}		
8	Entering an invalid password pattern	{ "userName": "testing", "userEmail": "testing@gmail.com", "userPassword": "testing", "userPhone": "78965422", "userRole": "admin"}	400: Password must be 8 to 20 characters, Password must contain at least 1 letter and 1 number	Showing the user the correct format of the password that must be entered.
9	Entering an invalid phone number length	{ "userName": "testing", "userEmail": "testing@gmail.com", "userPassword": "testing123", "userPhone": "78965", "userRole": "admin"}	400: Phone number length must be between 8 to 20 characters.	Showing the user the correct length of the phone number that must be entered.
10	Entering an invalid role name	{ "userName": "testing", "userEmail": "testing@gmail.com", "userPassword": "testing123", "userPhone": "7896589498", "userRole": "test"}	422: User role must be normal user or admin	Role entered is compared to the allowed role names, and return a message informing the user that the chosen role is invalid.

GET/api/User/Login

Test Case ID	Description	Input	Expected Result	Comment
11	Logging in successfully	Email: ahmed@gmail.com Password: ahmed123	200: generate a jwt for the user	A token is generated for the user containing user id, name, and role.
12	Missing email	Email: N/A Password: ahmed123	400: The email field is required	Throwing an error to inform the user that the email is required to login.
13	Missing password	Email: ahmed@gmail.com Password: N/A	400: The password field is required	Throwing an error to inform the user that the password is required to login.
14	Incorrect credentials	Email: nonexistantuser@gmail.com Password: ahmed123	401: unauthorized user	Throwing an exception to inform the user that the user entered is not registered.

GET/api/User/GetUserInfo

Test Case ID	Description	Input	Expected Result	Comment
15	Using endpoint after logging in	Execute method	200: Display User details	If the user is logged in and authorized, display the user details
16	Using the endpoint without logging in	Execute method	401: unauthorized	If the user is not logged in, display an unauthorized exception.

GET/api/User/ViewCart

Test Case ID	Description	Input	Expected Result	Comment
17	With items available in the cart	Execute method	200: display the cart items	The items details are displayed if the cart has items.
18	Without items in the cart	Execute method	200: empty cart	The cart is shown but with empty content.
19	Using the endpoint without logging in	Execute method	401: unauthorized	If the user is not logged in, display an unauthorized

				exception.

Product Controller:

POST/api/Product/Add-Product

Test Case ID	Description	Input	Expected Result	Comment
20	Successfully adding a product	{"productName": "testing123", "productDescription": "testing", "productPrice": 23, "productStock": 20}	200: display the created product details.	{"productName": "testing123", "productDescription": "testing", "productPrice": 23, "productStock": 20, "productRating": null}
21	Adding a product without logging in	-	401: unauthorized	Show an invalid token error in the response
22	Adding a product as a normal user	{"productName": "testing555", "productDescription":	403: unauthorized user	Only users with admin role are allowed to add items.

		"testing", "productPrice": 23, "productStock": 20}		
23	Adding a product with a missing name	{"productName": "", "productDescription": "testing", "productPrice": 23, "productStock": 20}	400: Product name is required	"status": 400, "errors": { "productName": ["Product name is required"] },
24	Adding a product with a missing description	{"productName": "testing827", "productDescription": "", "productPrice": 23, "productStock": 20}	200: display product details with empty description	Null description is allowed, and can be added later by updating.
25	Adding a product with	{"productName": "testing827",	400: Price must be	"status": 400, "errors": { "productPrice":

	invalid price	"productDescription": "test", "productPrice": 0, "productStock": 20}	greater than 0	["Price must be greater than 0"] },
26	Adding a product with invalid stock amount	{"productName": "testing827", "productDescription": "test", "productPrice": 1, "productStock": -1}	400: Stock cannot be less than 0	"status": 400, "errors": { "productStock": ["Stock cannot be less than 0"] },

PUT/api/Product/Update-Product/{id}

Test Case ID	Description	Input	Expected Result	Comment
27	Successfully updating a product	Id: 54 {"productName": "testingUpdated", "productDescription": "added description",	200: return the product updated details.	{"productName": "testingUpdated", "productDescription": "added description", "productPrice": 55, "productStock": 30, "productRating":

		"productPrice": 55, "productStock": 30}		null}
28	Updating with a product name of an existing product.	{"productName": "testing123", "productDescription": "added description", "productPrice": 55, "productStock": 30}	422: a product with this name already exists.	Informing the user that a product with this name already exists.
29	Updating a product using an incorrect id	Id: 555 {"productName": "testingNew", "productDescription": "added description", "productPrice": 55, "productStock": 30}	422: Product not found.	Informing the user that the product was not found.
30	Updating a product as a	Id: 54 {"productNam	403: unauthorized	Only admins are allowed to

	normal user	e": "testingNew", "productDescription": "added description", "productPrice": 55, "productStock": 30}	admin	update product details.

GET/api/Product/GetAllProducts

Test Case ID	Description	Input	Expected Result	Comment

31	Using the endpoint without any inputs	Default:
		pageNumber = 1
		pageSize = 1
		searchName = null
		minPrice = 0
		maxPrice = 2147483647
			200: display 1 item per page	Displays 1 item per page due to pageSize default.

32	Change page number.	pageNumber = 2	200: displays the next item in the list	Displays 1 item per page due to pageSize default.
33	Change	pageSize = 10	200: display 10	Determines

	pageSize		items per page	the number of items displayed per page.
34	Add search name	searchName = xbox	200: displays items that contain the term “xbox”	Filters items from the list that contain the search name.
35	Add min price	minPrice = 100	200: display items that have a pice equal to or above 100	Filters items by minimum price.
36	Add max price	maxPrice = 500	200: display items that have a price equal to or below 500	Filters items by maximum price.

GET/api/Product/GetProductById/{id}

Test Case ID	Description	Input	Expected Result	Comment
37	Entering a valid product id	Id = 1	200: display product details	Displays the details of the product with that id.
38	Interesting an invalid product id	Id = 555	404: product not found	Return a not found status code.

Order Controller

POST/api/Order/AddItemToCart/{name}/{quantity}

Test Case ID	Description	Input	Expected Result	Comment
39	Successfully adding a product to cart	Name: testing123 Quantity: 2	201: display item details	Product name: testing123 Quantity: 2 Total Cost: 46.00
40	Adding quantity more than available stock	Name: testing123 Quantity: 55	404: insufficient stock.	Product stock insufficient
41	Entering invalid quantity	Name: testing123 Quantity: -5	404: out of range	Specified argument was out of the range of valid values. (Parameter 'quantity must be 1 or above')
42	Entering an invalid item name	Name: InvalidItem Quantity: 1	404: not found	Product not found

POST/api/Order/PlaceOrder

Test Case ID	Description	Input	Expected Result	Comment
43	Successfully placing order	Execute method	201: display order details	Displays order details

				along with order items.
44	Placing an order with an empty cart	Execute method without adding items to cart	404: cart is empty	Cart is empty
45	Placing an order with an item that exceeds stock	Execute method	404: item quantity exceeds available stock	testing123 quantity exceeds available stock

DELETE/api/Order/RemoveItemFromCart/{productName}

Test Case ID	Description	Input	Expected Result	Comment
46	Successfully removing an item from cart	Name: testing123	200: display message of successful removal	Product "testing123" removed from cart
47	Removing an item that is not in cart	Name: abc	404	Product not found

GET/api/Order/GetUserOrders

Test Case ID	Description	Input	Expected Result	Comment
48	Getting user orders for user that has	Execute method	200: show orders	Displays each order with order info

	orders			
49	Getting user orders for user that made no orders.	Execute method	404: no orders	No orders made by this user

GET/api/Order/GetOrderDetails/{orderId}

Test Case ID	Description	Input	Expected Result	Comment
50	Successfully getting an order's details	orderId: 6	200: display order details	Displays the products of the order along with the order details.
51	Entering an incorrect order id	orderId: 55	404: order not found	Order not found

ReviewController

POST/api/Review/ReviewProduct

Test Case ID	Description	Input	Expected Result	Comment
52	Successfully creating a review	{"productName": "testing123", "rating": 5, "comment": "Good	201: return reviewed product id	Returns the id of the reviewed product

		Product"}		
53	Reviewing a product already reviewed before	{"productName": "testing123", "rating": 5, "comment": "Good Product"}	522: user has already reviewed this product	User has aleady reviewed this product
54	Entering a rating out of range	{"productName": "testing123", "rating": 0, "comment": "Good Product"}	400: rating must be between 1 and 5	"status": 400, "errors": { "rating": ["Rating must be between 1 and 5"]},
55	Reviewing a product not purchased by user	{"productName": "HP Spectre x360", "rating": 4, "comment": ""}	422: cannot review products not purchased yet	Cannot review products not purchased yet.
56	Not entering a comment	{"productName": "Canon EOS Rebel T7", "rating": 4, "comment": ""}	201: returns product id	Not entering comments is allowed, will be replaced by "No Comment"
57	Entering an invalid product name	{"productName": "jkbk", "rating": 4, "comment": ""}	422: product not found	Product not found

GET/api/Review/GetProductReviews/{productName}

Test Case ID	Description	Input	Expected Result	Comment

58	Successfully getting a product's reviews	Name: testing123
		pageNumber = 1
		pageSize = 1
			200: display review details	[{"rating": 5, "comment": "Good Product", "reviewDate": "2024-12-22T13:21:57.3698869", "userName": "Bob Smith", "productName": "testing123"}]

59	Getting reviews of a product with no reviews	Name: Keurig K-Elite	200: displays empty list	Displays a list with no content
60	Entering an invalid product name	Name: abc	404: product not found	Product not found

GET/api/Review/GetReviewsByUser

Test Case ID	Description	Input	Expected Result	Comment
61	Successfully getting user reviews	Execute method	200: display user reviews	Displays reviews made by user
62	Getting reviews of a user that made no reviews	Execute method	404: user has not made any reviews	User has not made any reviews

PUT/api/Review/UpdateReview

Test Case ID	Description	Input	Expected Result	Comment
63	Successfully updating a product review	{"productName": "testing123", "rating": 5, "comment": "very good"}	200: returns id of the reviewed product	Returns the id of the product in the review
64	Entering the wrong product name	{"productName": "qwe", "rating": 5, "comment": "very good"}	422: product not found	Product not found
65	Entering an out of range rating	{"productName": "testing123", "rating": 0, "comment": "very good"}	400: rating must be between 1 and 5	"status": 400, "errors": { "rating": ["Rating must be between 1 and 5"]},

DELETE/api/Review/DeleteReview/{productName}

Test Case ID	Description	Input	Expected Result	Comment
66	Successfully deleting a review	productName: testing123	200: returns the product id of the product reviewed by the deleted review	returns the product id of the product reviewed by the deleted review.

67	Entering a name of a product that was not reviewed by user	productName: testing123	422: review not found	Review not found