BOSTA TESTING TASK

Task I:

Create a well-structured and comprehensive set of test cases to test the functionality of the following user stories:

User story 1: open a credit card account

Description**:**
As a customer I want to open a credit card account

Acceptance criteria:

1. first if you are a new customer, you will get a 15% discount on all your purchases today.
2. second if you are an existing customer and you hold a loyalty card, you get a 10% discount.
3. third if you have a coupon, you can get 20% off today (but it cannot be used with the 'new customer' discount). Discount amounts are added, if applicable.

User story 2: Add product to cart.

Description:
As a customer,
I want to add a product to my cart.
So that I buy this product

Acceptance criteria:

1. product should have at least one item to be able to be added to the cart.
2. If the product is out of stock the user cannot add it to his cart.
3. The user clicks on add to cart button to add the product to his cart.
4. A message is displayed to the user to inform him the product is added to his cart.
5. There is a link in the notification message to redirect the user to his cart.
6. The user should be logged in to be able to add to his cart

User story 3: Contact us.

Description:

As a customer,
I want to contact the website about a problem in my product

Acceptance criteria:

1. The Email address should be valid with the following structure (name@domain.xyz)
2. The Email is a mandatory field
3. Contact name is a mandatory field
4. Message shouldn't be more than 500 characters
5. Message is a mandatory field

Task II:

Go to this website and do the following:

1. Create an automated test script using one of the following tools (Selenium, Cypress)
   To Automate the following scenario:
   1. Login to the website
   2. Navigate through the categories list
   3. Pick one of the products and open the product details page
   4. Add the product to the cart.
   5. Go to Cart page and place an order
   6. Click on purchase button without filling the order form
   7. Assert that an Error message is displayed
   8. Fill the form then click on purchase button
   9. Assert the display of the success message

The tests should utilize the best practice coding standards for maintainability, be aware of design patterns in your solution.
An advantage when using data driven scenarios.

Task III:

Write as many test cases as possible for this API:

Description

Get the full history of a specific delivery.

Test Data: tracking number to be used: 6636234, 7234258, 9442984,1094442

Path GET https://tracking.bosta.co/shipments/track/:trackingNumber

URL Params

Param	Type	Mandatory	Description
`trackingNumber`	string	Yes	The delivery tracking number.

Header

Field	Type	Description
`Content-Type`	string	Sets the format of payload you are sending. Can be application/json or text/xml Default value: `application/json`
`X-Requested-By`	string	Landmark
`Accept`	string	Sets what format the response body is returned in. Can be application/json or application/xml Default value: `application/json`

Success 200

Field	Type	Description
`CurrentStatus`	TrackingCurrentStatus Object	The current status of the delivery.
`TrackingNumber`	String	The tracking number of the delivery.

Field	Type	Description
`TrackingURL`	String	Array of customer support numbers the business can call in case of any questions.
`SupportPhoneNumbers`	Array of String	Lastname for the guest who made the booking
`CreateDate`	String	The creation date of the delivery
`TransitEvents`	Array of TrackingTransitEvent Object	Array of objects contains all the transit history for that delivery.

Errors

Error Code	Description
`404 (Not Found)`	if no delivery matched the given tracking number.
`500 (Internal Server Error)`	Something has gone wrong, and the server encountered a problem that prevented it from fulfilling the request.

Bonus: use postman tool to do the following:

1. Create new environment with the URL (https://tracking.bosta.co) and use it in the request
2. make the tracking number as a variable.
3. Add an assertion to make the response code is 200
4. Add an assertion to make sure the response time is less than 100ms.

Thanks!