**Shopping Cart Code Challenge**

* **Description of the problem and solution.**

**Problem:**

1. We have a products catalogue that has some products with their prices
2. User can select some products to buy by adding them in the cart
3. Some products have discounts
4. Taxes are added to the products purchased
5. User can choose to pay by which currency
6. What we need is to calculate the bill of the added items in the cart by taking in consideration the discounts, the taxes and currency that user will pay with

**Solution:**

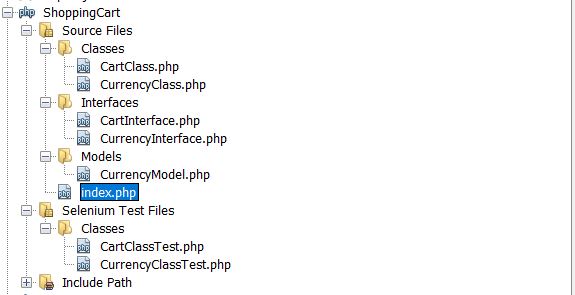
A program that defines the product catalogue and allow the user to choose the items to buy and choose the currency, then program will calculate the bill applying taxes and discounts and then display the total bill with the details of the total, taxes added and discounts.

* **Whether the solution focuses on back-end, front-end or if it's full stack.**

The solution focuses on backend using classes, interfaces, models and only 1 index file is used to display output by using html

* **Reasoning behind your technical choices, including architectural.**

**Architectural is as the below, image :**



1. Project architectural includes classes, models, interfaces, selenium test files test and view form for the display (Index file)
2. Interfaces folder, contains 2 interfaces:

|  |  |
| --- | --- |
| Currency Interface | * It is used to be implemented by currency class * It contains only 1 method : GetCurrencyDetails |
| Cart Interface | * It is used to be implemented by cart class * It contains 4 methods : CreateProductCatalogArray, GetCartProducts, CalculateCart |

1. Models folder, contains 1 model

|  |  |
| --- | --- |
| Currency Model | * It is used to create currency object that is used in currency class * It defines 3 properties for the currency: currency sign, currency convert and currency sign position |

1. Classes folder, contains 2 classes :

|  |  |
| --- | --- |
| Currency Class | * Class is used to define the currency details, I defined 3 currencies that the user can choose from, currencies are : USD, EGP and Euro , if the user entered any other currency program will exit * Class contains only 1 method : * GetCurrencyDetails: It takes currency as input and return the currency details |
| Cart Class | * Class is used to calculate the bill details price, taxes and discount * Class display the bill based on the calculation * Class contains 5 methods: * CreateProductCatalogArray: It return the product catalogue (product with its price) * GetCartProducts : It takes products added in the cart and catalogue and return the products that matches the catalogue after removing the products that not found * CalculateDiscounts: It takes product added in the cart and products catalogue and calculate both jackets and shoes discount * DisplayCartBill : It takes price, taxes , shoes discount, jacket discount and currency details as parameters and display the cart bill * CalculateCart: It takes product added in the cart and product catalogue array and calculate cart |

1. Selenium test files, contains 2 classes

|  |  |
| --- | --- |
| Currency Class Test | * Class is used to test currency class * It contains 5 methods: * testGetCurrencyDetails : Method that tests the currency details entered ; it asserts true if the return of currency details method in equal to currency model * testGetCurrencyDetailsForUSD: Method that tests the usd currency ,it asserts same if the return of currency details method in equal to details of usd currency model * testGetCurrencyDetailsForEGP: Method that tests the egp currency, it asserts same if the return of currency details method in equal to details of egp currency model * testGetCurrencyDetailsForEuro: Method that tests the euro currency, it asserts same if the return of currency details method in equal to details of euro currency model * testGetCurrencyDetailsForOtherCurrencies: Method that tests any other currencies not defined in currency class; it asserts same if the return of currency details method is equal to default object |
| Cart Class Test | * Class is used to test cart class * Class contains 5 methods: * testCreateCartArray: Method that tests create cart array; it asserts same if the return of the method is equal to the products catalogue array * testGetCartProducts: Method that test the product in the cart that match the product catalogue, it asserts true if the return of the method is not equal empty string * testCalculatePrice: Method that test the price calculation of the products in the cart, it asserts true if the return of the method is greater than 0 * testCalculateDiscounts: Method that test the discount calculation for both shoes and jacket, it asserts true if the return of the method is equal to array that contains both shoesDiscount and jacketDiscounts values * testDisplayCartBill: Method that test the display of the cart bill, it asserts instance of the exception if an exception is thrown * testCalculateCart: Method that test the cart calculation, it asserts instance of the exception if an exception is thrown |

1. Index file: it is used to call methods, display the product catalogue in html table, display the items user has chosen, the currency and the details of the bill

**Reason behind this technical choice:**

1. It fulfils the OOP, as it includes classes and objects
2. It sticks to SOLID principle , it follows the open closed principle as it’s open for extension and closed for modification as we could use currency class for any currencies without even changing the method and same for the cart class, we can use it to calculate any cart bill, we can also add any extra methods we want. It also follows Interface segregation principle, as for each class there is an interface defined for it
3. Supports error handling by using try and catch

* **Trade-offs you might have made, anything you left out, or what you might do differently if you were to spend additional time on the project.**

**Trade-offs I might have made :**

1. I will make this solution as a real shopping cart
2. I will use more frontend techniques for a better view

**Something I left out:**

1. Applying unit test cases, I setup the IDE to run the test cases, I wrote the test methods for each class but I couldn’t run the file, it’s giving me this error : ‘PHP Fatal error(s) occurred, test results can be incomplete!’

**What I would do differently:**

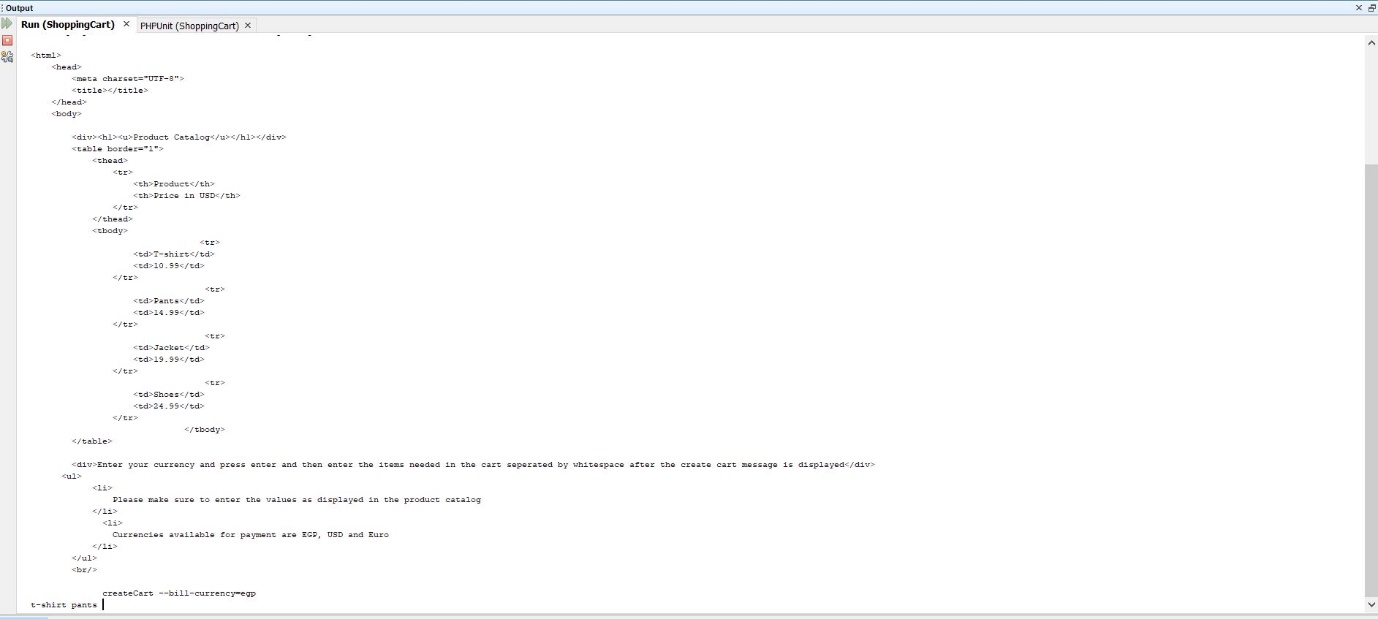
1. I would try other ways to create and run test cases

**How to run the solution and display input:**

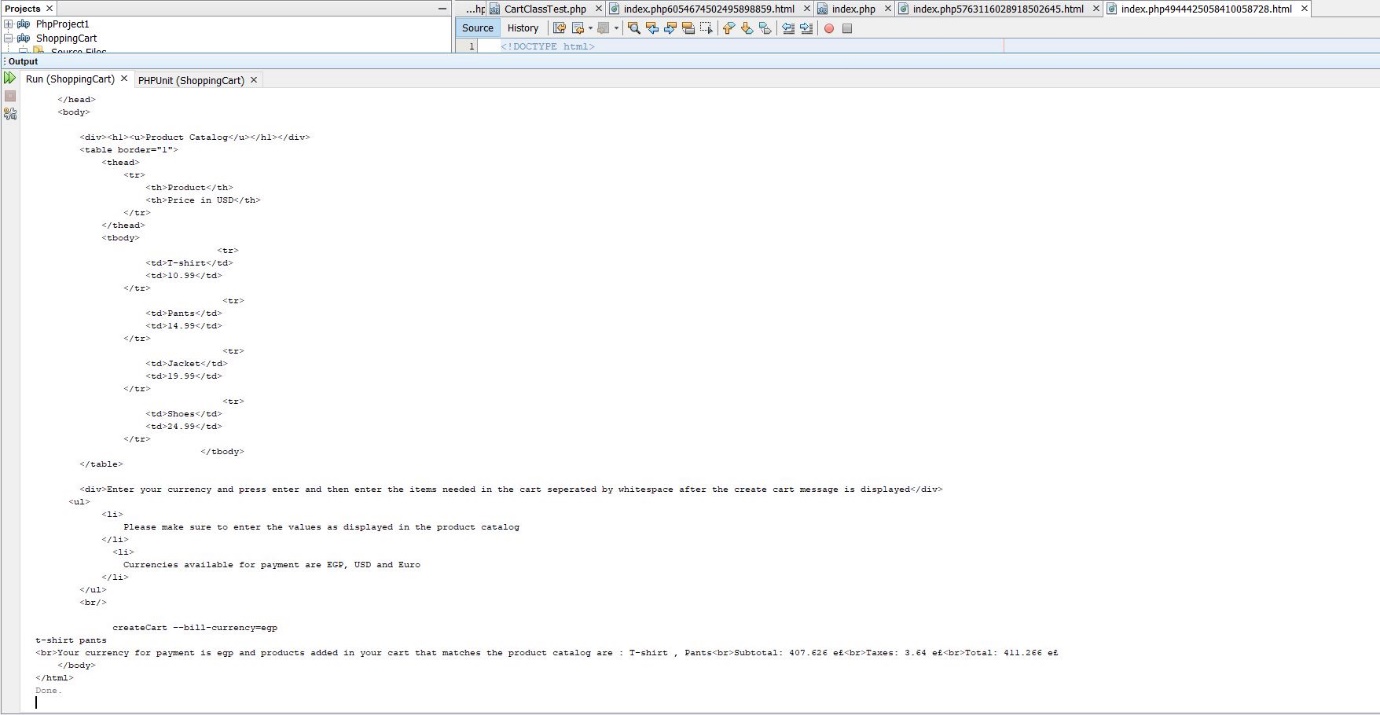
1. When solution runs, the command line appears and asking for the input as the below :

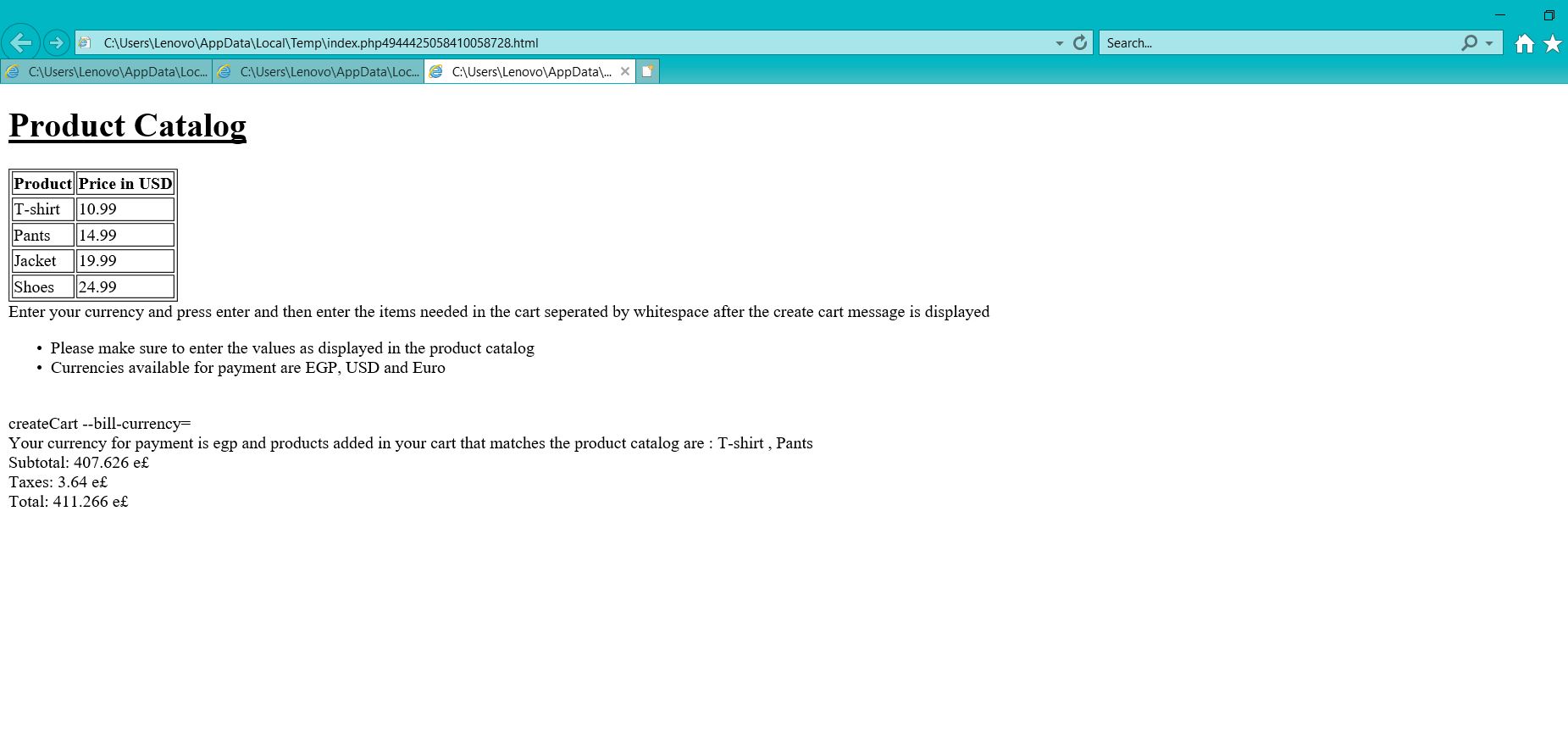


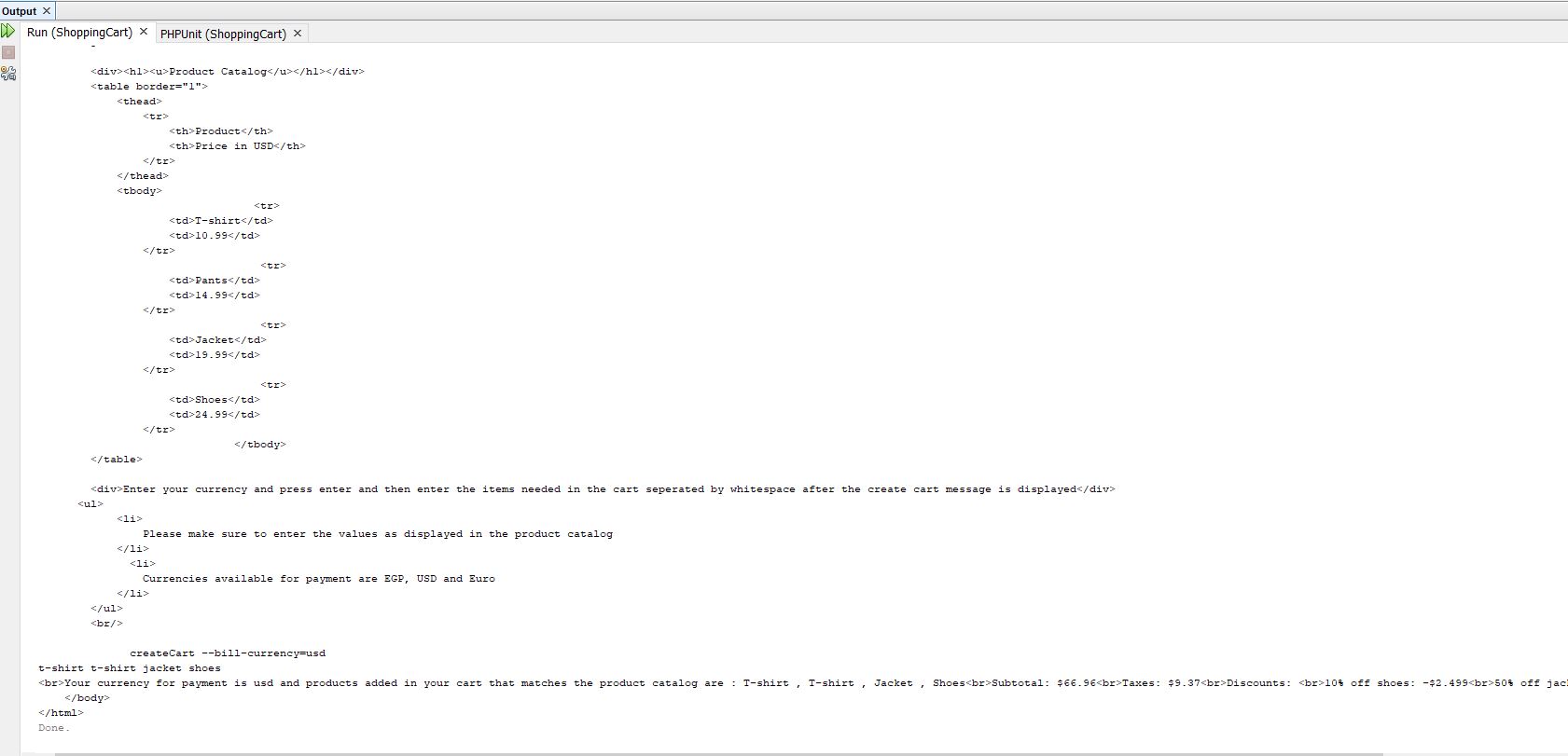
1. User shall enter the currency and press enter then enter the products separated by whitespace as the below and then press enter:

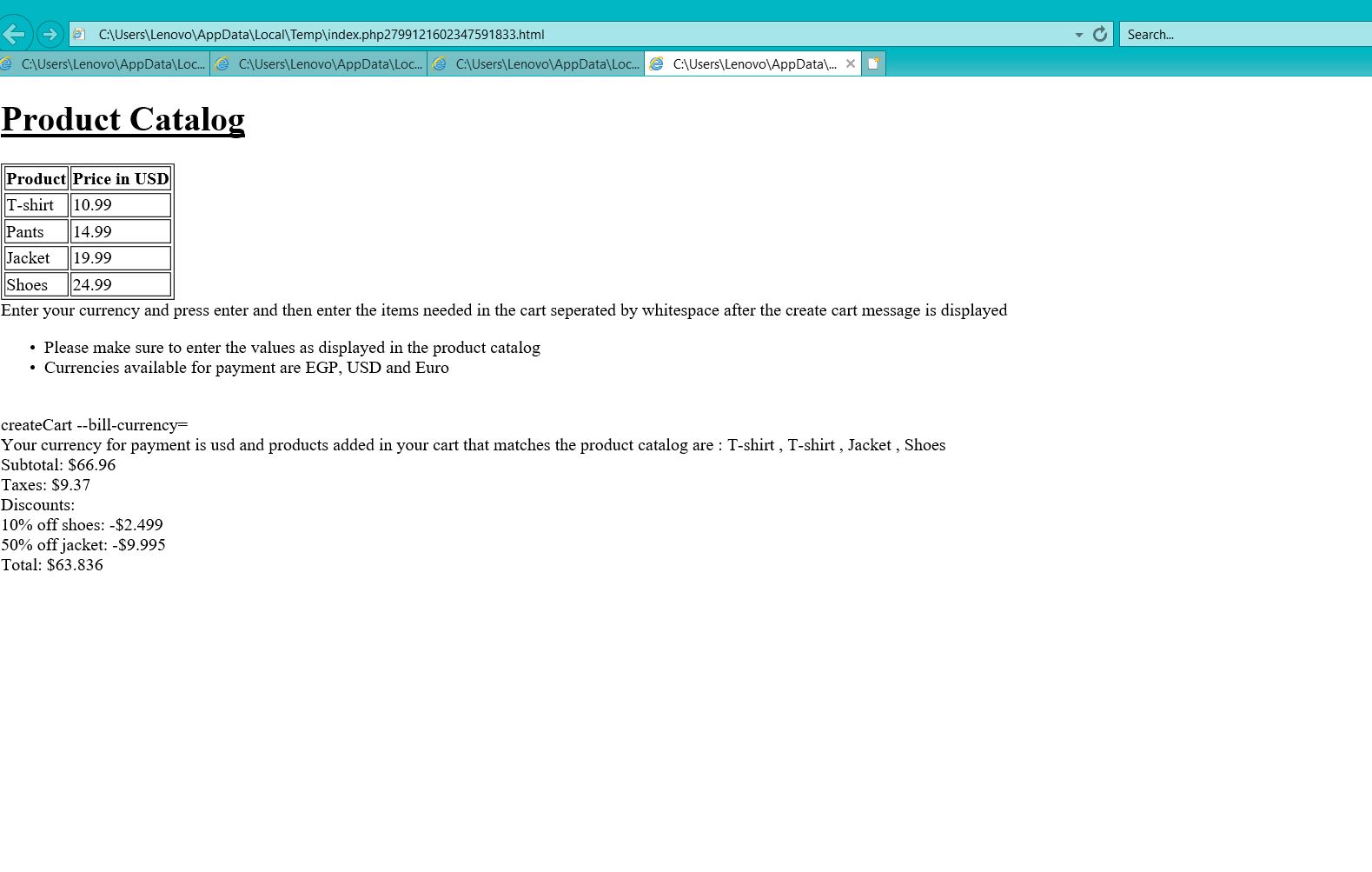


1. Bill will be calculated directly and results will appear in the command line and web as the below:









**IDE used:**

1. Netbeans
2. Xampp to run the projects and configure unit testing