American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Summer 2022-2023

Section: F

Software Quality and Testing

PROJECT TITLE:

Ghore Ghore an E-commerce Site

A Report Submitted

By

SN	Student Name	ID
1	MD. SADMANUR ISLAM SHISHIR	20-42871-1
2	ROBIUL ISLAM	19-41115-2
3	MD. RAZIB AHMED RAZU	19-41183-2
4	MD. PEYAL SARKER	20-43357-1

Under the supervision of

MD. MOMAN UL HAQUE KHAN

Software Test Plan

For

Ghore Ghore an E-commerce Site

Version 1.0 approved.

Prepared by

MD. SADMANUR ISLAM SHISHIR
MD. RAZIB AHMED RAZU
ROBIUL ISLAM
MD. PEYAL SARKER

American International University-Bangladesh (AIUB)

August 26, 2023

Checked By Industry Personnel

Name:

Designation:

Sign:

Date:

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.8.10	MD. SADMANUR ISLAM SHISHIR	First Draft
0.2	2023.8.12	MD. RAZIB AHMED RAZU	Second Draft
0.3	2023.8.15	ROBIUL ISLAM	Third Draft
0.4	2023.8.18	MD. PEYAL SARKER	Fourth Draft
0.5	2023.8.20	MD. RAZIB AHMED RAZU	Fifth Draft
0.6	2023.8.25	MD. SADMANUR ISLAM SHISHIR	Final

1. TESTING PLAN IDENTIFIER: RS-MTP01.3

2. REFERENCES

1. Website Link: https://ghoreghore.000webhostapp.com/
2. https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-automation-testing
3. https://learn.microsoft.com/en-us/dotnet/core/testing/

3. INTRODUCTION

3.1 Background to the Problem

In the dynamic landscape of e-commerce, technology's rapid growth has reshaped shopping paradigms, offering convenience and choice. However, in regions like Bangladesh, non-uniform technology access poses challenges. GHORE GHORE, an innovative e-commerce platform, aims to bridge this gap and make online shopping inclusive for all Bangladeshi citizens. In Bangladesh, urban tech adoption contrasts rural infrastructural limitations, hindering broad e-commerce integration. GHORE GHORE's vision arose from the belief that e-commerce benefits - convenience and variety - should transcend urban boundaries. GHORE GHORE's inception recognized the transformative potential of e-commerce for all. Yet, disparities in internet access, digital literacy, and reliable connectivity persist. Rural populations face barriers to accessing products available to urban peers instantly. GHORE GHORE's journey confronted multifaceted challenges. It developed a user-friendly platform, navigated complex logistics for nationwide

delivery, and promoted digital literacy in underserved areas. In essence, GHORE GHORE aims to democratize e-commerce, bridging urban-rural consumer gaps and fostering digital inclusion. This pursuit not only redefines online shopping but also propels Bangladesh's socio-economic advancement, ensuring equity in the digital era.

3.2 Solution of the Problem

GHORE GHORE, an innovative e-commerce platform, responds to the digital accessibility gap in Bangladesh by offering a holistic solution. The platform's strategy centers on equitable access to online shopping for all citizens.

1. **Enhanced Infrastructure:** GHORE GHORE collaborates with local authorities and tech providers to improve rural internet connectivity and digital infrastructure, minimizing access barriers.

2. **User-Centric Design:** The platform ensures inclusivity through user-friendly interfaces, multiple languages, and intuitive navigation, catering to various digital literacy levels.

3. **Innovative Logistics:** GHORE GHORE employs creative delivery models, partnering with local logistics and optimizing last-mile strategies for timely product access, even in remote areas.

4. **Digital Literacy Campaigns:** Underserved regions benefit from GHORE GHORE's digital literacy initiatives, offering training on internet use, online shopping, and security.

5. **Localized Product Range:** Collaborating with local artisans, the platform offers diverse, region-specific products, celebrating cultural variety and attracting a broader customer base.

6. **Affordability:** GHORE GHORE introduces cost-effective pricing and discounts, ensuring economic barriers don't hinder online shopping.

7. **Collaborative Approach:** Partnerships with institutions, NGOs, and government bodies amplify impact, fostering digital literacy and holistic development.

8. **User Feedback Loop:** Continuous improvements are driven by customer feedback, gathered through surveys and reviews, adapting to evolving needs.

GHORE GHORE envisions an inclusive e-commerce landscape, transcending geographic and socioeconomic divides, and propelling Bangladesh into an equitable and advanced digital future.

4. REQUEIRMENT SPECIFICATION

4.1 System Features

1. **System Registration**

   Functional Requirements

   i) The website must allow users to register with the necessary information.

   ii) If the username is not unique, the system will prompt the user to try registration with a different username again.

   Priority Level: High

   Precondition: Not applicable.

2. **System Login**

   Functional Requirements

   i) Users must be able to login using their username and password.

   ii) If the username and password have been entered incorrectly more than three times, the random verification code will be generated and sent by email by the system to retry login.

   Priority Level: High

   Precondition: The user must have a valid username and password.

3. **System Forget Password**

   Functional Requirements

   i) Users must be able to recover their password if they forget their password.

   ii) If they forget their password, user can recover their password by using their valid email address which is used to create their account and the system send a mail to recover the password.

   Priority Level: High

   Precondition: The user must use a valid email address.

4. **Search Product**

   Functional Requirements

   i) User can search products by the product name.

   ii) If the search product is not found system gives a message that this product does not exist on this website.

Priority Level: Medium

Precondition: Search by product name only.

5. **Add to Cart**

Functional Requirements

i) Users can use add to cart button to save products for buying in the future.

ii) If the user adds any product to the cart before login, they can show this product after login also.

Priority Level: Medium

Precondition: You need to click on Add to Cart button to save the product.

6. **Buy Product**

Functional Requirements

i) Users can buy any product from this website.

ii) If anyone wants to buy any product then need to click on this product and click on the buy button.

Priority Level: Medium

Precondition: Must show the product details for buy.

7. **Make payment**

Functional Requirements

i) User needs to select a payment method to confirm the order.

ii) There are two options to make payment, one is Cash on Delivery and another is Bkash. If the user is not select any option, he or she can't buy the product.

Priority Level: High

Precondition: The user must need to select any payment method.

8. **Show Cart**

Functional requirements

i) User can see their saved product by clicking on the Cart option on this website.

ii) In this option user can delete the saved product or buy the product by selecting Delete or Buy option.

Priority Level: Medium

Precondition: Need to add the product to the cart.

9. **Chat Box**

Functional requirements

i) Using can find their solution using the chat box.

Priority Level: Low

Precondition: No precondition needed.

10. **Profile**

Functional requirements

i) User can see profile and change profile picture.

Priority Level: High

Precondition: The user must need login first.

11. **Edit profile**

Functional requirements

i) User can edit their information and update it.

Priority Level: High

Precondition: The user must need to login first.

12. **Change the password**

Functional requirements

i) User must update their password.

Priority Level: High

Precondition: The user must need to login and use the old password also.

13. **System Logout**

Functional requirements

i) User must log out from the website after use.

Priority Level: High

Precondition: The user must login first.

4.2 System Quality Attributes

1. **Usability:** The website must be user-friendly. The website should be intuitive and simple to navigate.

2. **Efficiency:** The website should maximize the capacity and memory of the processor. Any task should be completed with optimal efficacy.

3. **Security:** Website security should be sufficient to prevent unauthorized access to website functions to prevent information loss, protect data privacy, and safeguard the website against hackers.

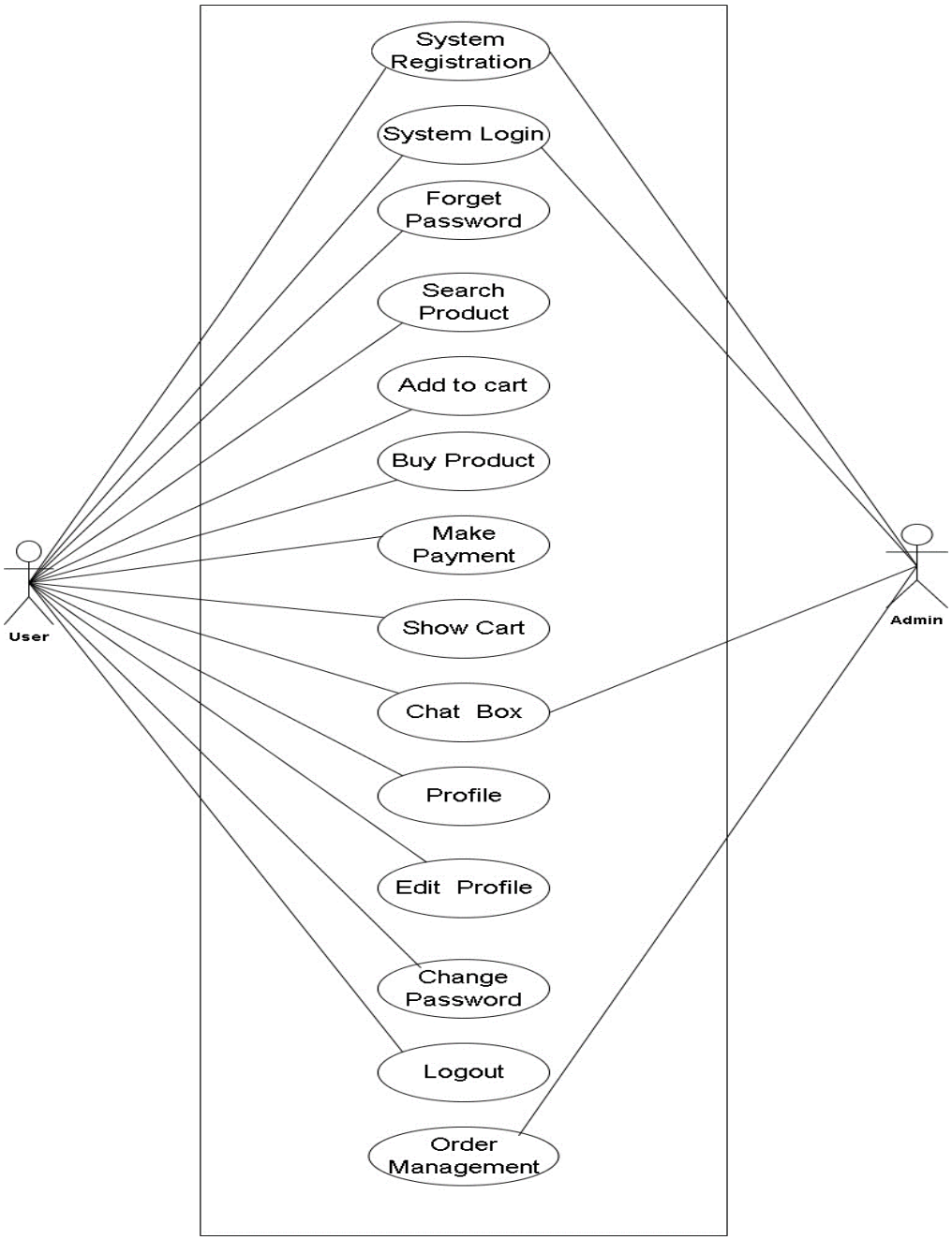
4. **Modularity:** The website's every block of code must be under separate and acceptable modules.

5. **Testability:** The website should be simple to test and identify flaws.
6. **Flexibility:** The website should be flexible enough to be modified.

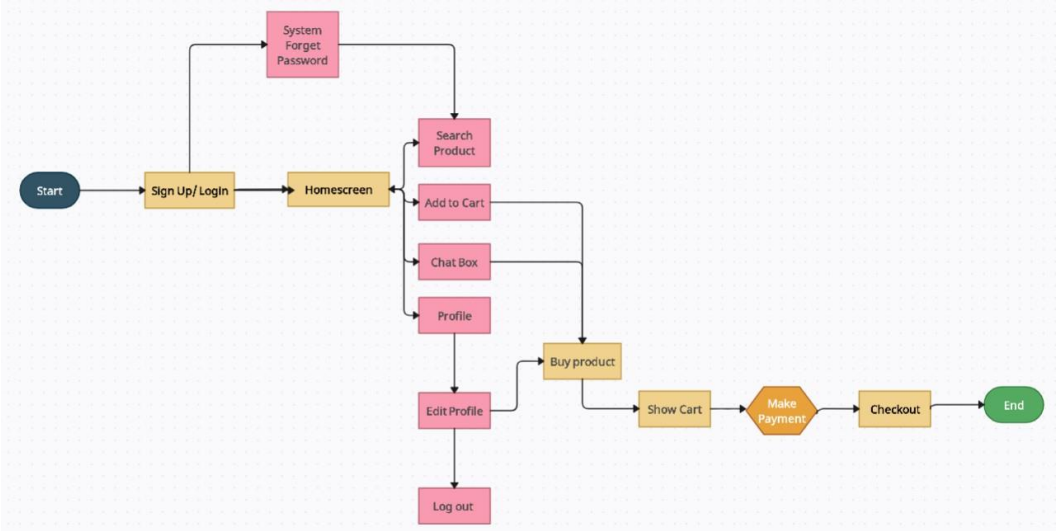
7. **Reusability:** Code library classes should be general enough to be utilized on multiple versions of a website or new projects.

4.3 System Interface

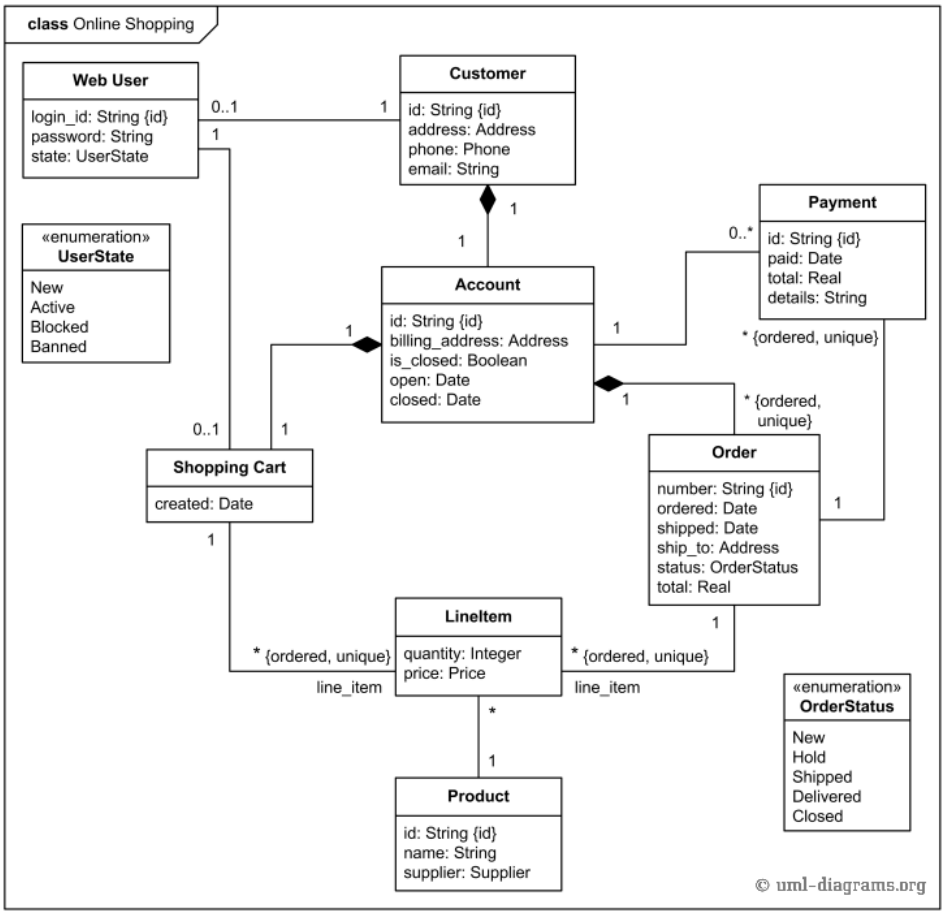
Use Case Diagram:



UML Diagram:



Class Diagram:



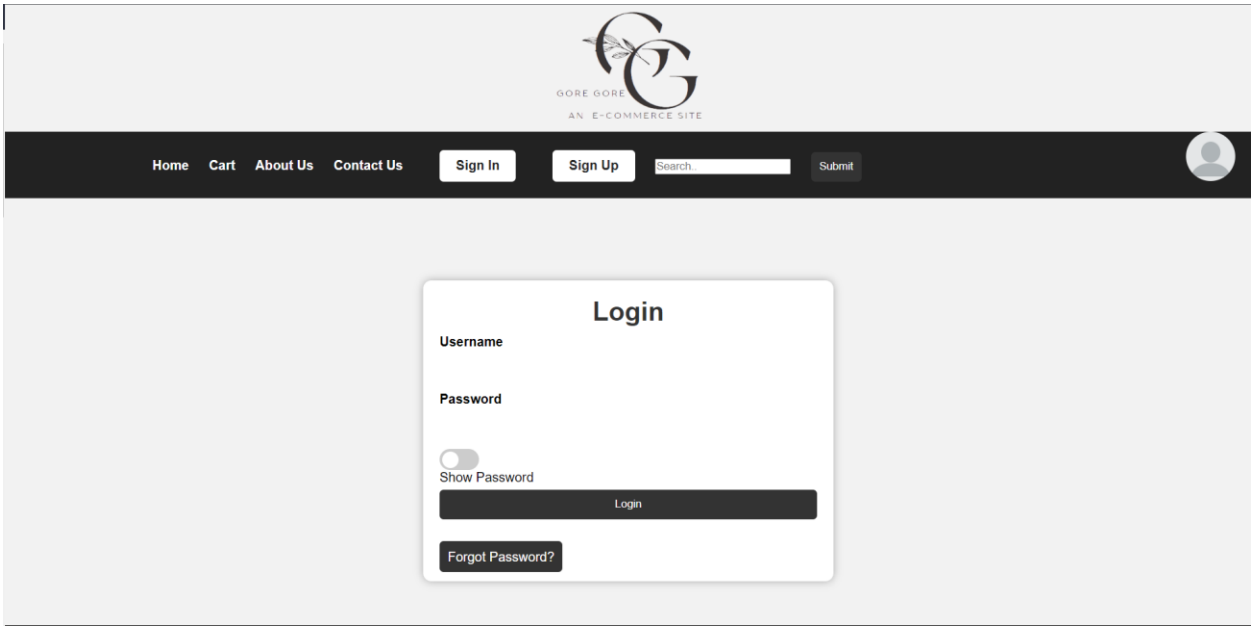


Fig: Login Page

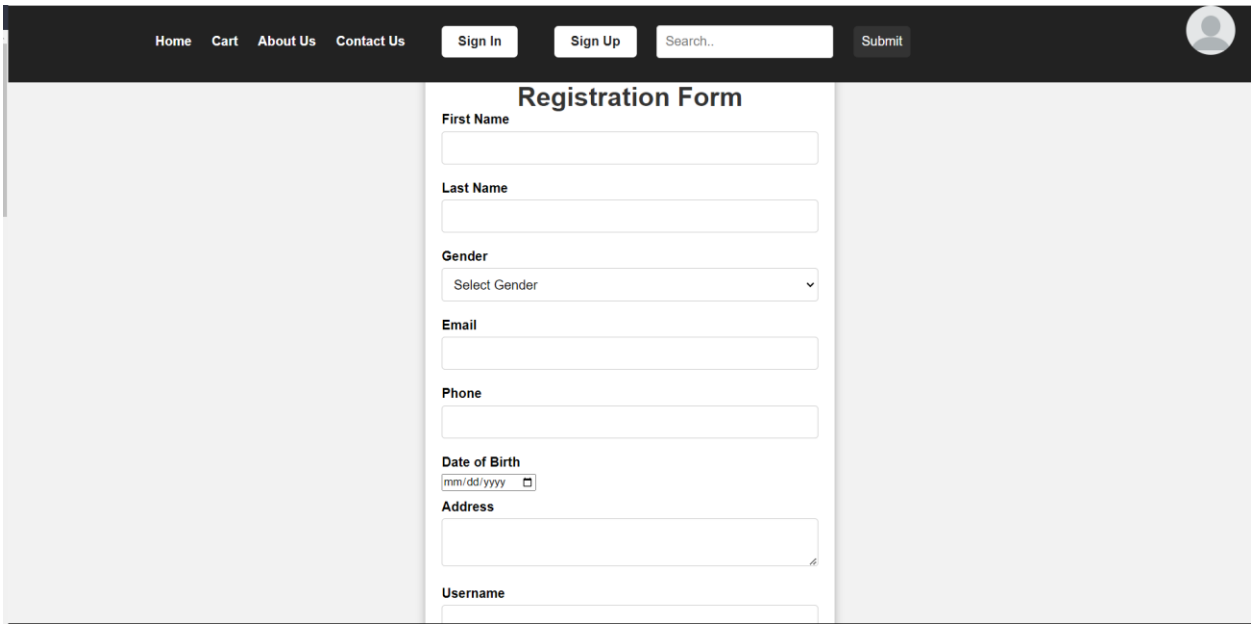


Fig: Registration page

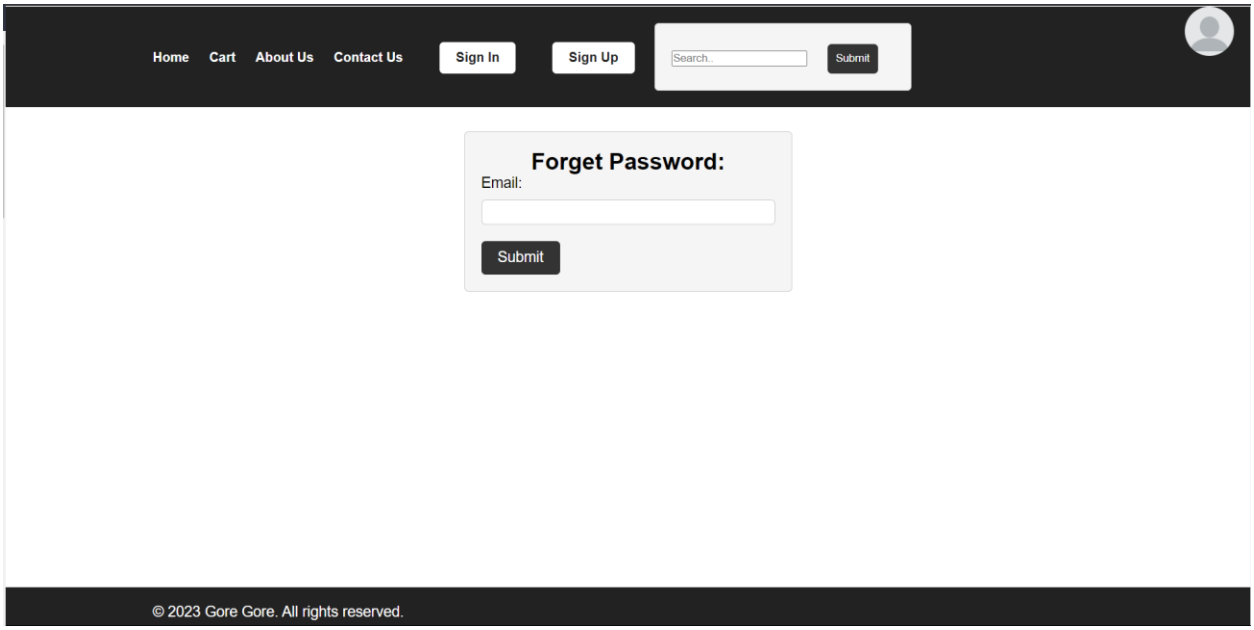


Fig: Forget Password

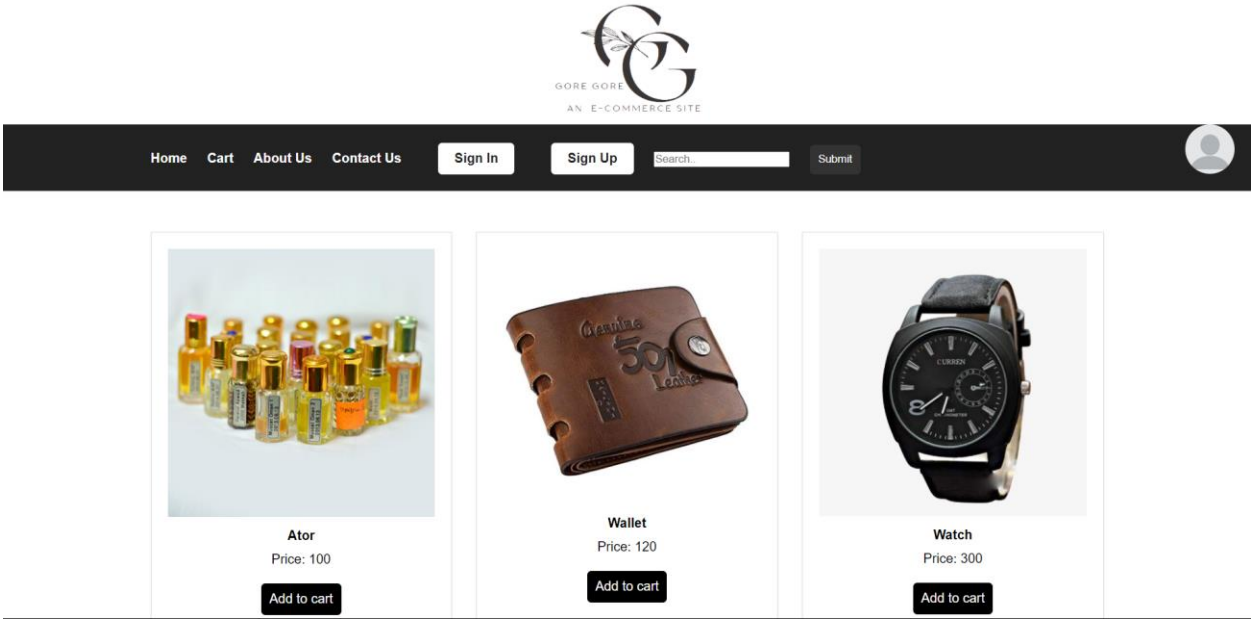


Fig: Home Page

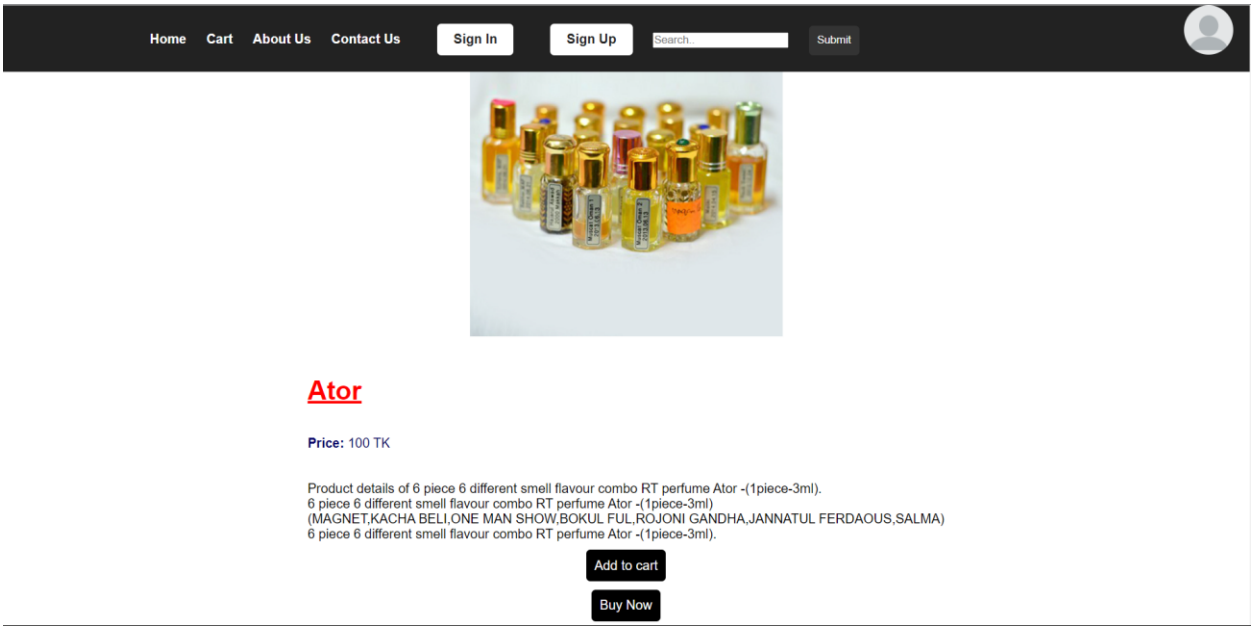


Fig: Product Info

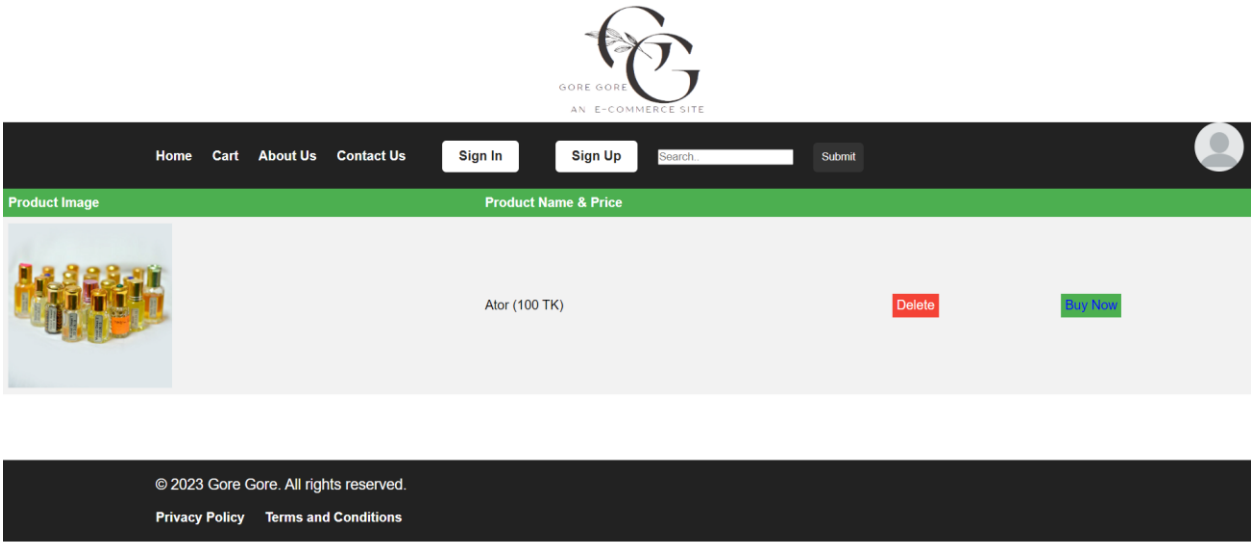


Fig: Add to Cart

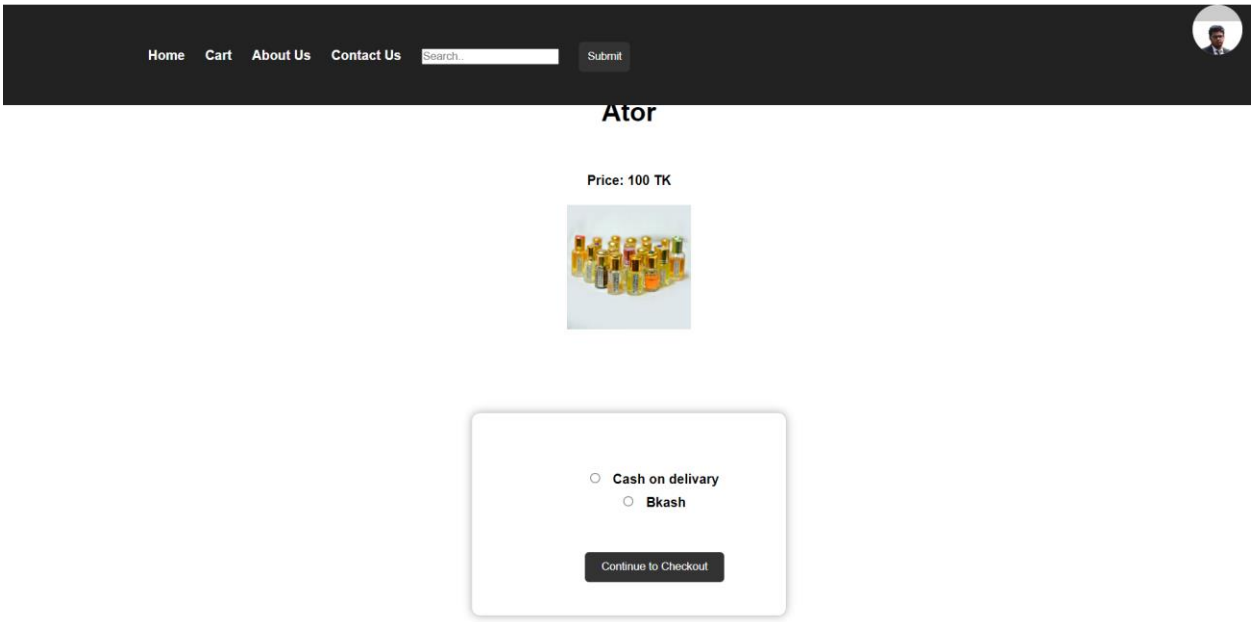


Fig: Buy Product

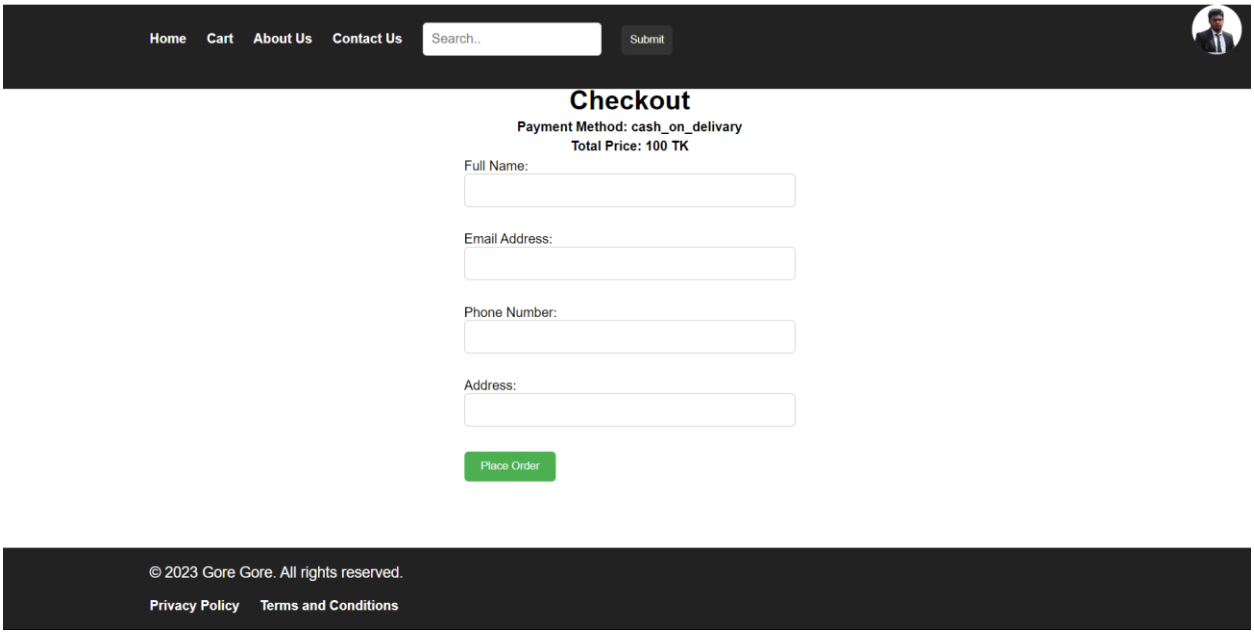


Fig: Place Order

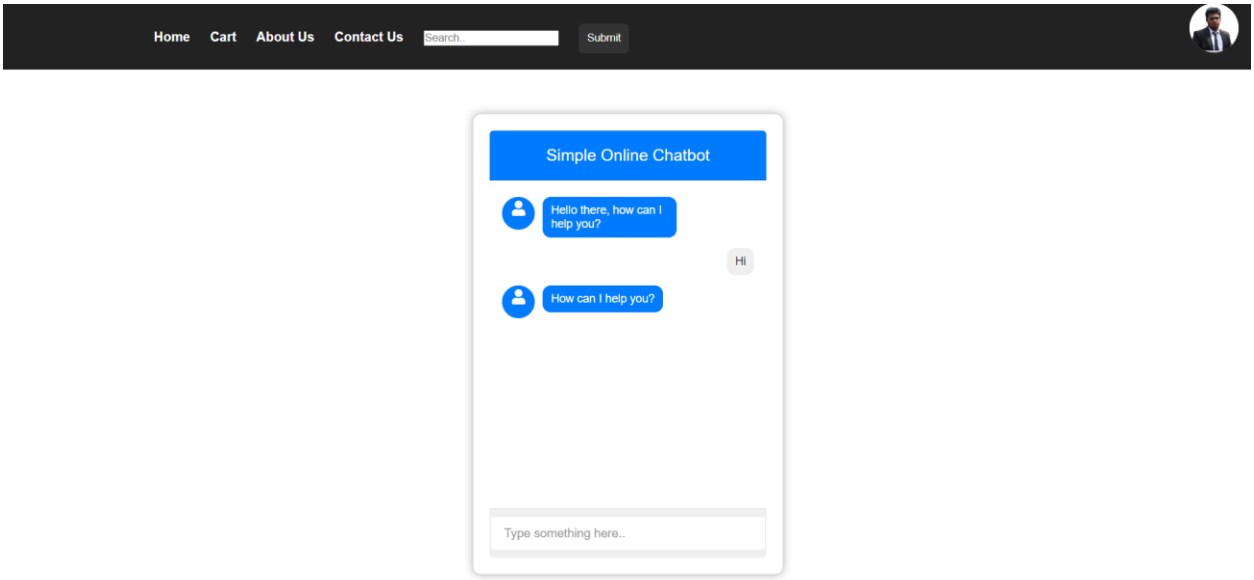


Fig: Chat Box

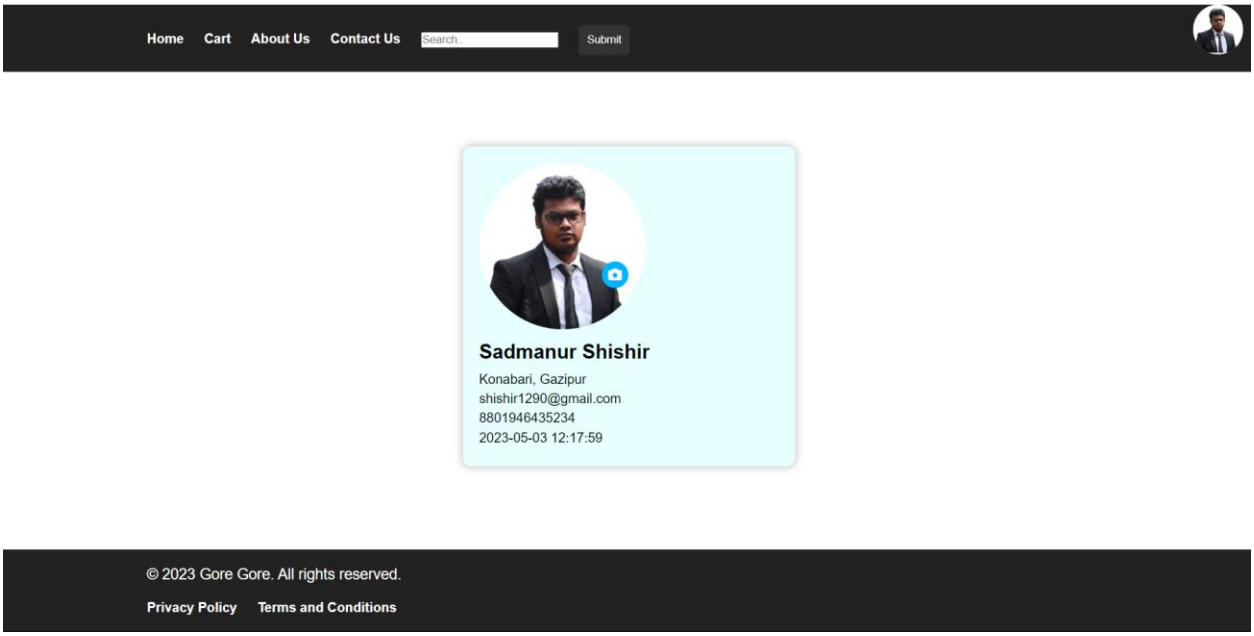


Fig: Profile

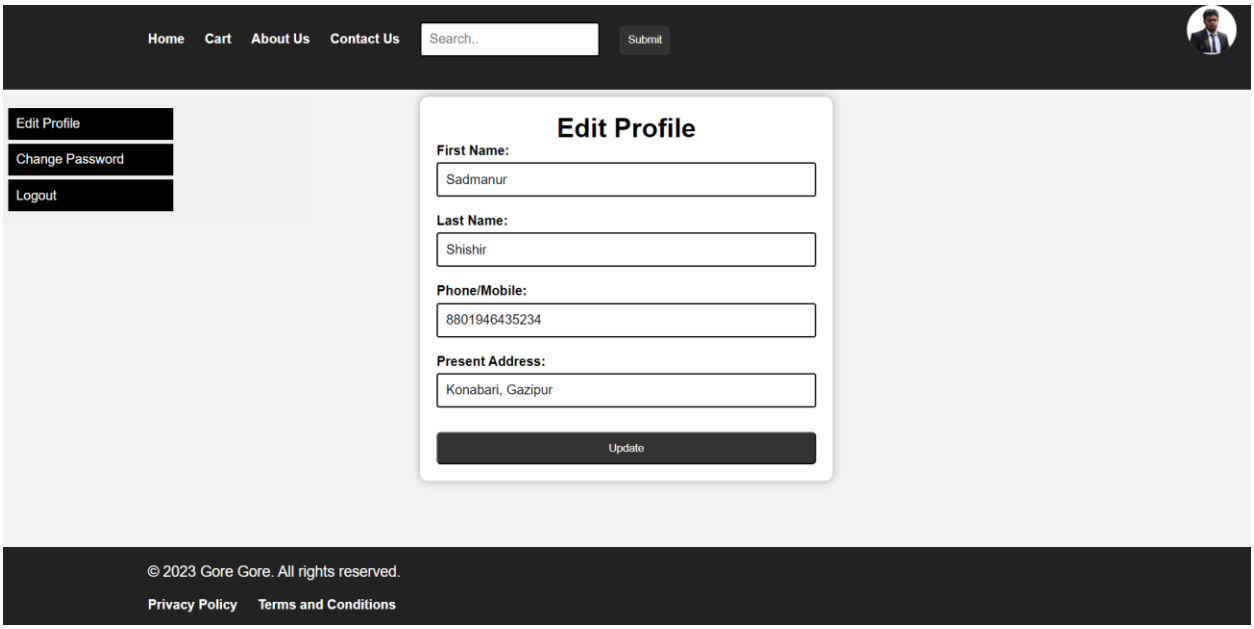


Fig: Edit Profile

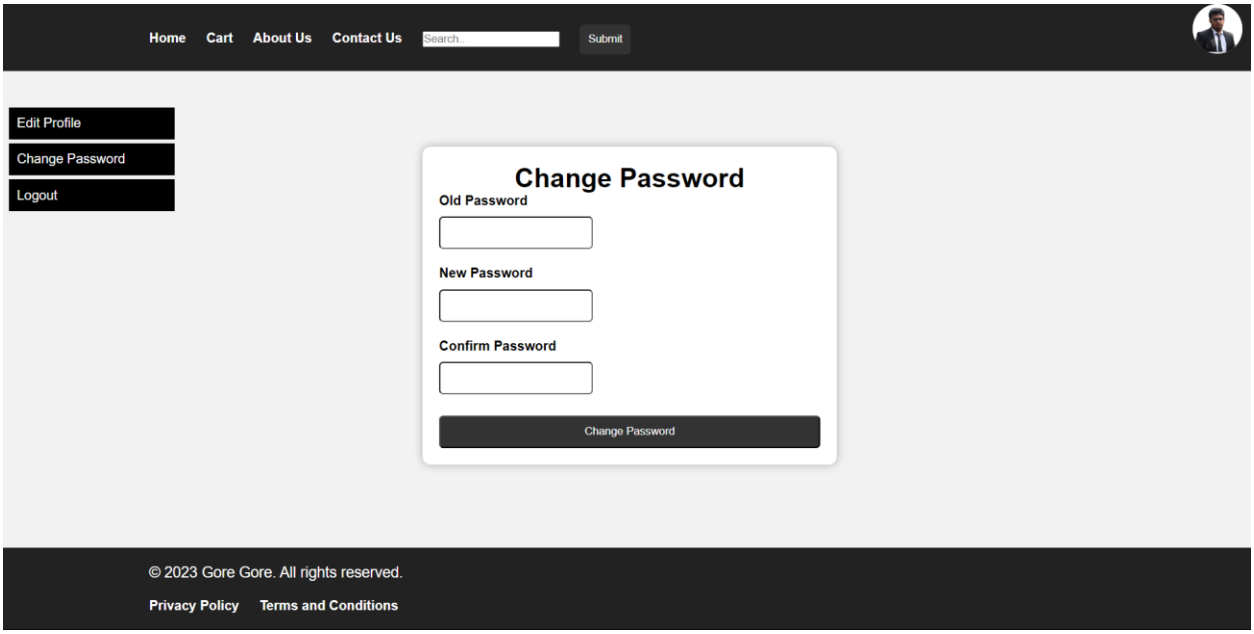


Fig: Change Password

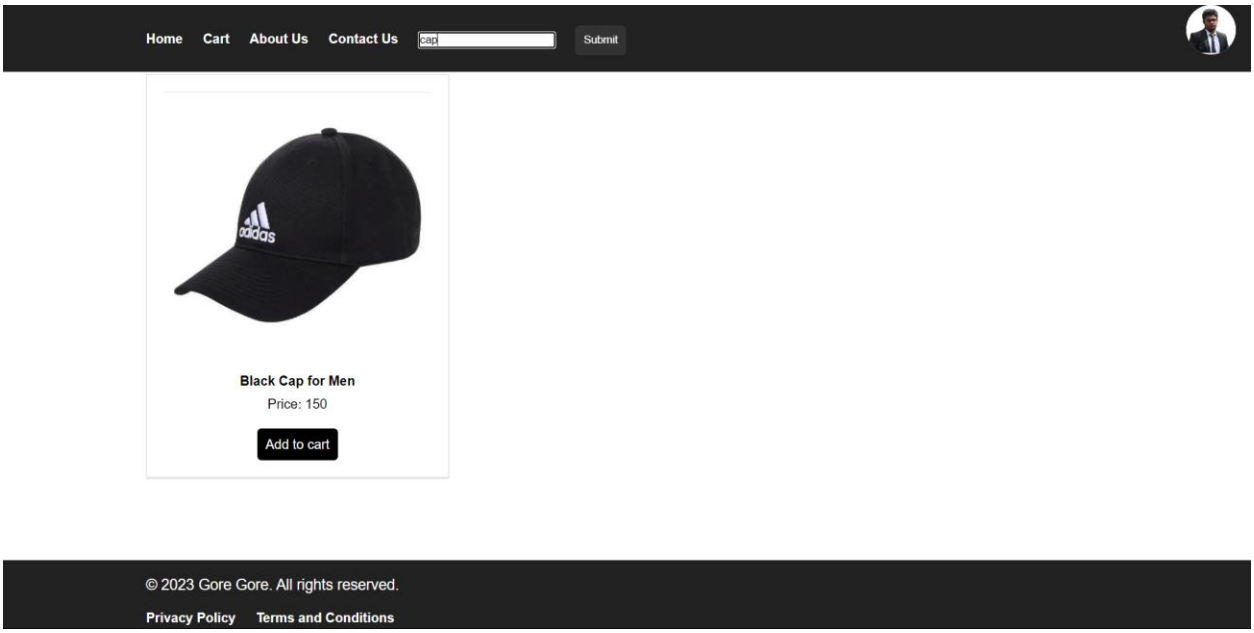


Fig: Search Product

4.4 Project Requirements

i) The source code must be in PHP, HTML, CSS, and JS language.

ii) For the website database, we use the MySQL database server, but other databases are also acceptable.

iii) For website development, we use visual studio.

iv) For testing, we use selenium automation.

v) The website is suitable for any device.

4.5 Constructive Cost Model

Software project type: Organic; [b = 1.05]
Coefficient < Effort Factor > = 2.4
Effort = PM
So, b = 1.05, c = 2.5 and d = 0.38
KLOC = 15000 Lines

PM = Coefficient < Effort Factor > * (KLOC/1000) ^b = 2.4*(15000/1000) ^1.05
= 41.22 Person-month

Development time, DM = c*(PM)^d = 2.5*(41.22) ^0.38 = 10.27 moths = 11 months
=1760 Working hours in total (Per week 40 hours)

Required number of people, ST = PM/DM
= 41.22/11
= 3.74 people
= 4 people

4.6 Budgeting

Developer/ Tester salary of 11 months**:**
= 400 Taka per hour, Total salary = 400 *1760
= **7, 04, 000 Taka**

Requirement analysis**:**
Required time = 1 month = 25 working days
= 200 working hour Requirement analysis
persons per hour salary = 250 Taka
Total requirement analysis salary = 250 * 200 = **50,000 Taka**

Transportation cost**: 15,000 Taka**
(Approximate) Hardware expense:
1, 20, 000 Taka

(Approximate) Rent expenses:
Total in 11 months = **1, 65, 000 Taka** [Per
month = 15,000 Taka]

Total utilities in 11 months: **15,000 Taka**
(Approximate)

Maintenance (Till 4 months after delivery):
Cost per hour = 1,200 Taka
Total estimated time needed for maintenance = 40 hours
Total estimated maintenance cost = 1,200 * 40 = **48,000 Taka**

Project manager's salary of 11 months:
Per month salary = 40,000 Taka
Total salary = 40,000 * 11 = **4, 40, 000 Taka**

Accountant's salary of 11 months:
Per month salary = 12,000 Taka
Total salary = 12, 000 * 11 = **1, 32, 000 Taka**

Total expense: 7, 04, 000 + 50,000 + 15,000 + 1, 20, 000 + 1, 65, 000 + 15,000 + 48,000 + 4, 40, 000 + 1, 32, 000 = **1, 689, 000 Taka**
Profit: 25% of total expense = 1, 689, 000 * 25% = **4, 22,250 Taka**

5. FEATURES NOT TO BE TESTED

1. Networks
2. Hardware
3. User's registration information (Name, Address, Email, Phone Number)

6. TESTING APPROACH

6.1 Testing Levels

The testing for the "GHORE GHORE an E-commerce site" project will consist of Unit, System test levels. It is hoped that there will be at least one full-time independent test person for system testing.

Unit Testing:

Unit testing involves testing individual components, functions, or classes in isolation to ensure they work as expected. For an e-commerce site like "GHORE GHORE," you might consider the following unit tests:

1. **User Management:**
   Entry Criteria:
   - Functions/methods responsible for user registration, login, profile, update, and account deletion are implemented.

   Exit Criteria:

   - All test cases for user registration, login, profile update, and account deletion pass successfully.
   - Test coverage includes various scenarios like valid inputs, invalid inputs, edge cases, etc.
   - The user management functions have been reviewed and approved by the development team.

2. **Cart Functionality:**
   Entry Criteria:
   - Functions/methods for adding items, updating quantities, and removing items from the cart are implemented.

   Exit Criteria:

   - All test cases for adding, updating, and removing items from the cart are passing.
   - Different scenarios like adding duplicates, updating with invalid quantities, and removing non-existent items are tested.
   - Code reviews have been conducted for cart functionality.

3. **Order Processing:**
   Entry Criteria:
   - Functions/methods for creating orders, calculating order totals, and applying discounts are implemented.

   Exit Criteria:

   - Successful execution of test cases related to order creation, total calculation, and discount application.
   - Tests cover scenarios like applying multiple discounts, checking correct calculations, and handling various order scenarios.
   - Order processing code has been reviewed and validated.

4. **Payment Integration:**
   Entry Criteria:
   - Payment processing functions/methods are integrated and ready for testing.

   Exit Criteria:

   - Successful execution of test cases for processing valid payments and handling invalid payment scenarios.
   - Refund functionality is tested and verified to work as expected.
   - Payment integration code has been reviewed and verified.

5. **Search and Filtering:**
   Entry Criteria:
   - Product search and filtering functions/methods are implemented.

   Exit Criteria:

   - All test cases related to product search and filtering pass successfully.

- Testing includes scenarios like searching for products with different keywords, applying filters, and testing boundary conditions.
- Code for search and filtering has been reviewed and tested.

6. **Reviews and Ratings:**
   Entry Criteria:
   - Functions/methods for submitting reviews and calculating/displaying ratings are implemented.

   Exit Criteria:

   - Successful execution of test cases for submitting reviews and calculating average ratings.
   - Testing includes cases like submitting reviews with various lengths, calculating accurate average ratings, and displaying ratings correctly.
   - Code for reviews and ratings has been reviewed and validated.

System Testing:

System testing involves testing the integrated system as a whole to ensure all the components work together seamlessly. For "GHORE GHORE," consider the following system tests:

1. **End-to-End Order Flow:**
   Entry Criteria:
   - All individual components and functionalities are unit tested and integrated.
   - Basic usability testing has been conducted to ensure a functional website.

   Exit Criteria:

   - Successful execution of end-to-end tests that cover the entire order flow, from product browsing to payment.
   - Test cases include various scenarios like successful orders, failed payments, abandoned carts, etc.
   - The order flow is reviewed and approved by relevant stakeholders.

2. **User Account Management:**
   Entry Criteria:
   - User registration, login, and profile management functions are implemented and unit tested.
   - Basic role-based access control is in place.

Exit Criteria:

- Successful execution of tests covering user registration, login, and profile management for different user roles.
- Tests include scenarios like successful logins, failed logins, profile updates, etc.
- Role-based access control is thoroughly tested and validated.

3. **Concurrency and Performance:**
   Entry Criteria:
   - Core website functionalities are stable and integrated.
   - Test environment is set up to simulate heavy user loads.

   Exit Criteria:

   - Successful execution of performance tests under various user load scenarios.
   - Website response times and resource usage are within acceptable limits.
   - Concurrent order processing and cart management are tested successfully.

4. **Security and Privacy:**
   Entry Criteria:
   - All security measures and encryption mechanisms are implemented.

   Exit Criteria:

   - Successful execution of security tests to verify data encryption and protection against common vulnerabilities.
   - Sensitive user information and payment details are properly secured.
   - Security vulnerabilities are identified and addressed.

5. **Cross-Browser and Cross-Device Testing:**
   Entry Criteria:
   - Core website functionalities are implemented and stable.

   Exit Criteria:

   - Successful execution of tests on various browsers and devices.
   - The website functions correctly and is visually appealing on different platforms.
   - Cross-browser and cross-device issues are identified and resolved.

6. **Integration with Third-Party Services:**
   Entry Criteria:
   - Integration with third-party services like payment gateways and shipping providers is complete.

   Exit Criteria:

   - Successful execution of tests that validate the integration with external services.
   - Communication between the system and third-party services works as expected.
   - Data exchange and synchronization are verified.

7. **Error Handling and Logging:**
   Entry Criteria:
   - Core functionalities are implemented and integrated.

   Exit Criteria:

   - Successful execution of tests that cover different error scenarios and edge cases.
   - Error messages are clear and helpful to users.
   - Proper logging of errors and exceptions is validated.

8. **Accessibility Testing:**
   Entry Criteria:
   - Core website functionalities are implemented and stable.

   Exit Criteria:

   - Successful execution of accessibility tests using tools and guidelines.
   - The website is accessible to users with disabilities, including screen readers and keyboard navigation.
   - Accessibility issues are identified and remediated.

6.2 Test Tool

1. The Selenium web driver tool will be used for automated testing. We utilize this instrument to discover errors and verify that our website is high quality, responsive, progressive, and consistent.

2. Jira project management tool will be used to share documents, communicate with team members, keep track of schedule and planning, the progress of the testing project, and so on.

6.3 Meeting

The test team will once every week evaluate progress to date and identify error trends and problems as early as possible. The test team leader will meet with the development and the project manager once every two weeks as well. These two meetings will be scheduled in different weeks. Additional sessions can be called as required for emergencies.

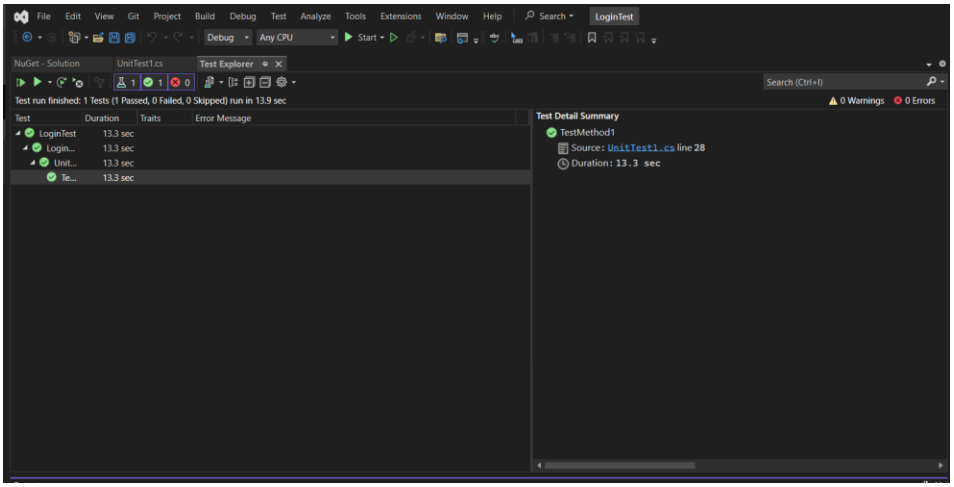
Date	Meeting With	Meeting Title
2023-03-12	Client	Project Kickoff
2023-03-20	Client	Defining Website Goals
2023-04-05	Project Manager	Project Roadmap Review
2023-04-15	Team Leader	Design and UI Discussion
2023-05-02	Developer	Technology Stack Discussion
2023-05-18	Client	Progress Review
2023-06-02	Test Team Leader	QA Strategy
2023-06-12	Team Leader	Design and UI/UX Review
2023-06-24	Client	Marketing and Promotion Plans
2023-07-01	Developer	Feature Implementation
2023-07-09	Client	Feedback and Iteration
2023-07-17	Team Leader	Performance Optimization
2023-07-24	Test Team Leader	User Acceptance testing
2023-08-07	Client	Final Product Review
2023-08-12	Project Manager	Project Conclusion

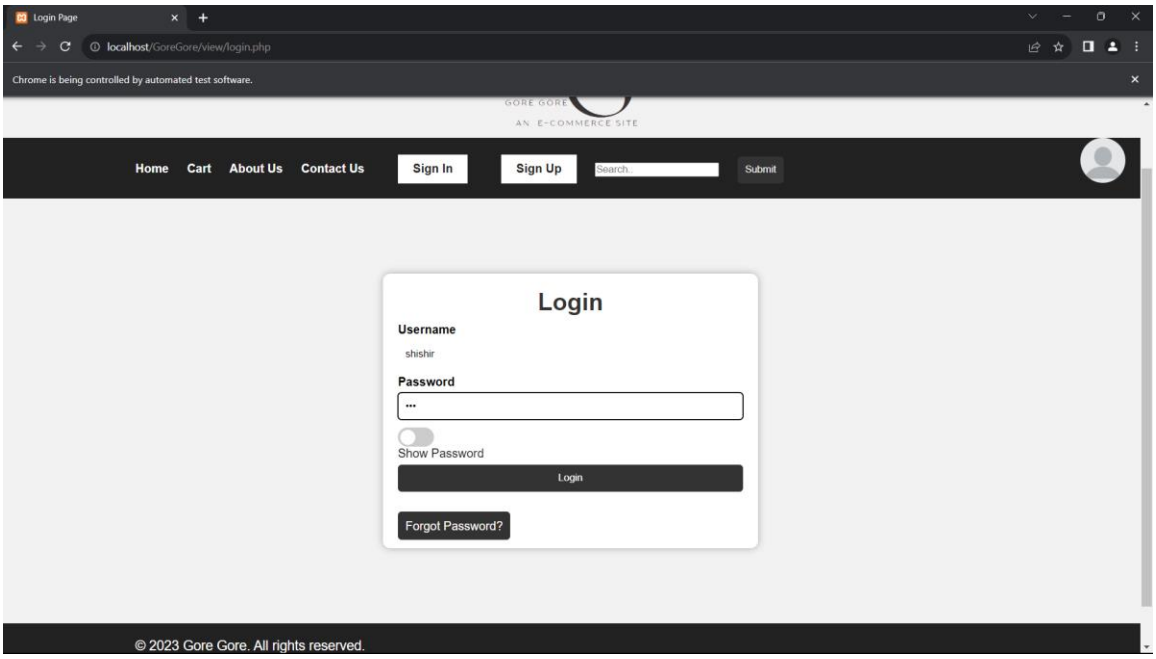
7. TEST CASES / TEST ITEMS

1.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 1</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High.</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: System Registration Session.</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: User Registration with Non-Unique Username</td></tr>
<tr><td colspan="5">Description: Test To verify that the system handles non-unique usernames during user registration.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user is on the registration page.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to the registration page of the website.
2. Enter the required information in the registration form, including a non-unique username that is already in use by another user.
3. Click the "Register" or "Sign Up" button to submit the registration form.</td><td>Username: "shishir" (assuming this username is already taken by another user)
Password: “123”</td><td>The system should detect the non-unique username and display an error message to the user. The user should not be successfully registered with a non-unique username.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition:
1. The user's registration is not completed.
2. The registration form fields remain populated with the entered information except for the username.
3. The error message indicating the non-unique username is displayed on the page.</td></tr>
</table>

```
15          [TestInitialize]
            0 references
16          public void Setup()
17          {
18              driver = new ChromeDriver(); // You can use other drivers like FirefoxDriver, EdgeDriver, etc.
19              driver.Navigate().GoToUrl("http://localhost/GoreGore/view/login.php");
20          }
21          [TestCleanup]
            0 references
22          public void Cleanup()
23          {
24              driver.Quit();
25          }
26
27          [TestMethod]
            0 references
28          public void TestMethod1()
29          {
30              IWebElement usernameField = driver.FindElement(By.Id("username"));
31              IWebElement passwordField = driver.FindElement(By.Id("password"));
32              IWebElement loginButton = driver.FindElement(By.Id("login"));
33
34              // Enter login credentials
35              usernameField.SendKeys("shishir");
36              Thread.Sleep(1000);
37              passwordField.SendKeys("123");
38              Thread.Sleep(5000);
39
40              // Click the login button
41              loginButton.Click();
42              Thread.Sleep(5000);
43          }
44      }
45  }
```





2.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 2</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High.</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: System login session.</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: User Login with Incorrect Credentials and Verification Code Generation</td></tr>
<tr><td colspan="5">Description: Test to verify that the system handles user login with incorrect credentials and generates a verification code after three unsuccessful login attempts.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user must have a valid username and password for the login.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to the login page of the website.
2. Enter the user's valid username and incorrect password.
3. Click the "Login" button.
4. Repeat steps 2 and 3 two more times, entering incorrect credentials each time.
5. On the fourth attempt, enter incorrect credentials again.</td><td>Valid Username: "razu21"

Incorrect Password: "incorrect123"

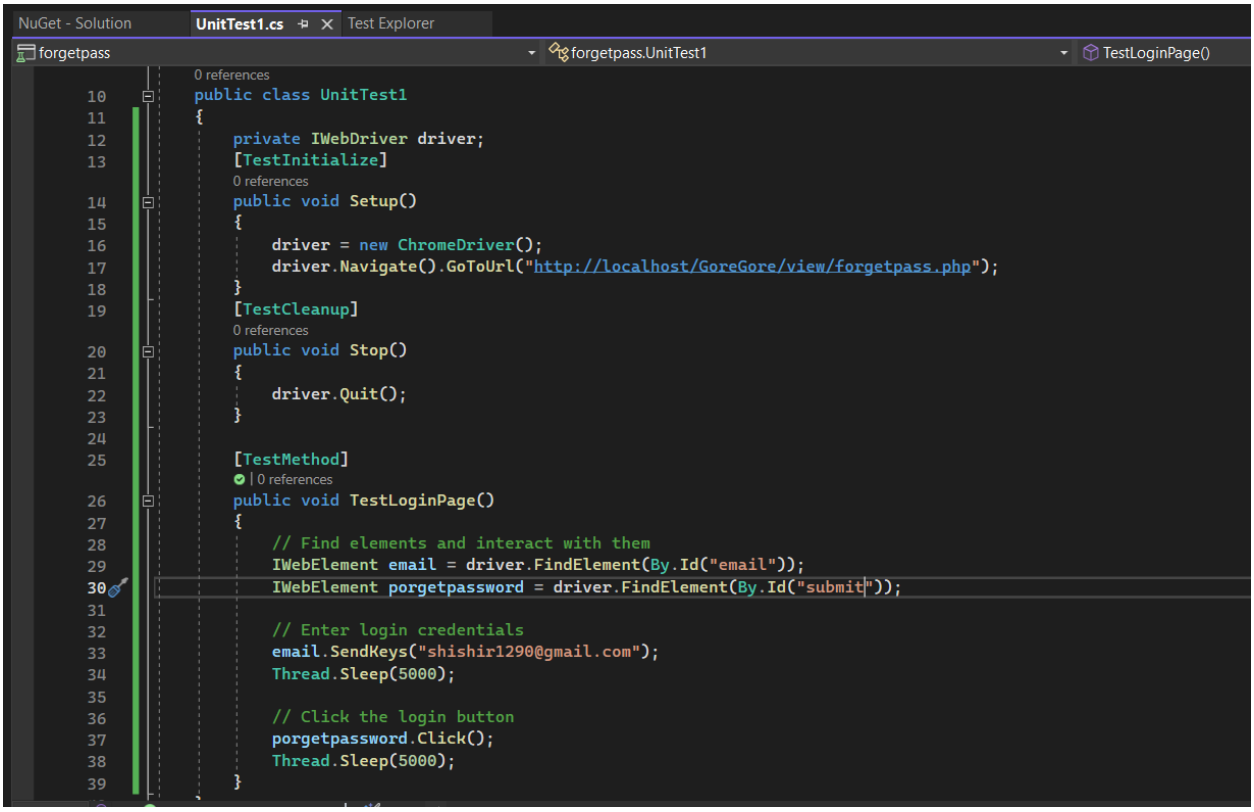
Email Address: ahmed.razu1426@gmail.com</td><td>After three unsuccessful login attempts with incorrect credentials, the system should recognize the pattern and generate a random verification code.
The system should send the verification code to the user's registered email address.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition:
1. The user is not logged in.
2. The user is prompted to check their email for the verification code.
3. The login form fields remain populated with the entered username and password, except for the password.
4. A message indicating the generation and email of the verification code is displayed on the page.</td></tr>
</table>

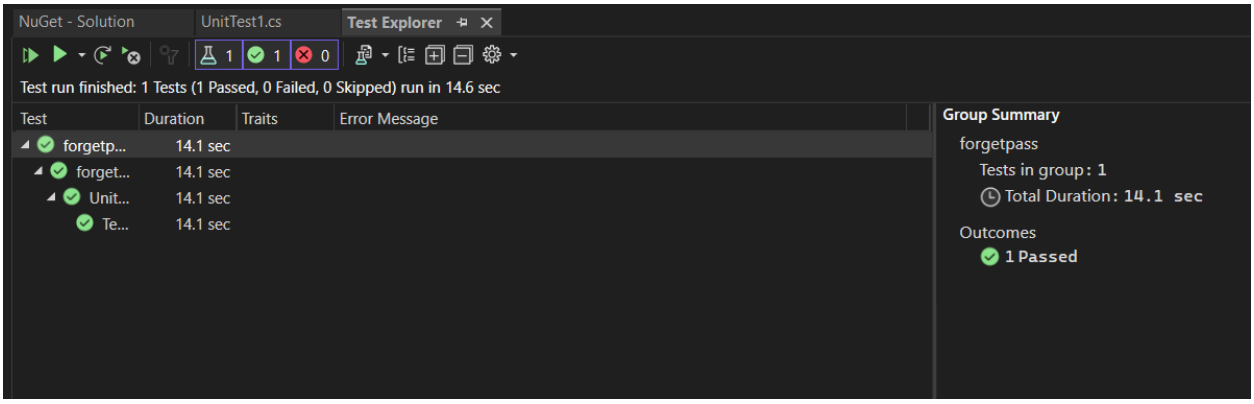
3.

Project Name: Ghore Ghore an E-commerce site	Test Designed by: Razib
Test Case ID: 3	Test Designed date: 20/8/23
Test Priority (Low, Medium, High): High.	Test Executed by: Razib
Module Name: System Forget Password	Test Execution date: 23/8/23
Test Case: Password Recovery via Email	
Description: Test to verify that the system allows users to recover their forgotten password using their valid email address.	
Precondition (If any): 1. The system is accessible and functional. 2. The user has a valid account with a registered email address.	

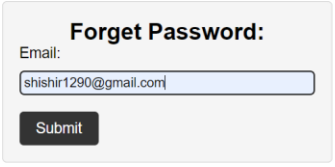
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the password recovery or "Forgot Password" page of the website. 2. Enter the user's valid email address associated with the account. 3. Click the "Recover Password" or "Submit" button.	Valid Email Address: shishir1290@gmail.com	The system should validate the entered email address and check if it corresponds to a registered user account. If the email address is valid and associated with a user account, the system should send a password recovery email to that address.	As expected	Pass

Post Condition: 1. The user receives a password recovery email at the provided email address. 2. The password recovery page displays a success message.


NuGet - Solution
UnitTest1.cs
Test Explorer
forgetpass
forgetpass.UnitTest1
TestLoginPage()
0 references
public class UnitTest1
{
private IWebDriver driver;
[TestInitialize]
0 references
public void Setup()
{
driver = new ChromeDriver();
driver.Navigate().GoToUrl("http://localhost/GoreGore/view/forgetpass.php");
}
[TestCleanup]
0 references
public void Stop()
{
driver.Quit();
}
[TestMethod]
0 references
public void TestLoginPage()
{
// Find elements and interact with them
IWebElement email = driver.FindElement(By.Id("email"));
IWebElement porgetpassword = driver.FindElement(By.Id("submit"));
// Enter login credentials
email.SendKeys("shishir1290@gmail.com");
Thread.Sleep(5000);
// Click the login button
porgetpassword.Click();
Thread.Sleep(5000);
}


NuGet - Solution
UnitTest1.cs
Test Explorer
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 14.6 sec
Test
Duration
Traits
Error Message
forgetp...
14.1 sec
forget...
14.1 sec
Unit...
14.1 sec
Te...
14.1 sec
Group Summary
forgetpass
Tests in group: 1
Total Duration: 14.1 sec
Outcomes
1 Passed


Home
Cart
About Us
Contact Us
Sign In
Sign Up
Search..
Submit


Forget Password:
Email:
shishir1290@gmail.com
Submit

4.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 4</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): Medium</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: Search Product.</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Searching for a Product-by-Product Name</td></tr>
<tr><td colspan="5">Description: Test to verify that the system allows users to search for products by product name and handles cases where the searched product is not found.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. Products are available in the system's database.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Navigate to the website's search functionality or search bar.
2. Enter the name of an existing product that is available in the system's database.
3. Click the "Search" button or press Enter.</td><td>Existing Product Name: "Wallet"</td><td>The system should perform a search based on the entered product name.
If the product with the entered name exists, the search results should display the details of the product.
If the product with the entered name does not exist, the system should display a message stating that the product does not exist on the website.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The search results page is displayed with either the product details or the "Product not found" message.</td></tr>
</table>

5.

Project Name: Ghore Ghore an E-commerce site			Test Designed by: Razib	
Test Case ID: 5			Test Designed date: 20/8/23	
Test Priority (Low, Medium, High): Medium			Test Executed by: Razib	
Module Name: Add to Cart.			Test Execution date: 23/8/23	
Test Case: Adding a Product to the Cart and Persistence After Login.				
Description: Test to verify that the system allows users to add products to their cart and ensures the persistence of cart items even after the user logs in.				
Precondition (If any): 1. The system is accessible and functional. 2. The user has products available for selection on the website.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to a product page on the website. 2. Click the "Add to Cart" button associated with a specific product. 3. Log out from the current session (if logged in). 4. Log in to the website using valid credentials. 5. Navigate to the user's cart or shopping bag.	Product Selected: "Wallet"	After step 2, the system should add the selected product to the user's cart. After step 4, the system should recognize the user and link the cart items to their account. After step 5, the cart should display the previously added product, which was added before logging in.	As expected	Pass
Post Condition: The cart should display the selected product even after logging in.				

6.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 6</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): Medium.</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: Buy Product.</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Purchasing a Product from the Website</td></tr>
<tr><td colspan="5">Description: Test to verify that the system allows users to purchase a product from the website by clicking on the product and proceeding with the buying process.</td></tr>
<tr><td colspan="5">Preconditions (If any):
1. The system is accessible and functional.
2. The user is logged in to their account.
3. Product details are available and shown on the website.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in to the website using valid credentials.
2. Navigate to a product page on the website.
3. Review the product details and ensure they are accurate.
4. Click the "Buy" or "Add to Cart" button on the product page.
5. Proceed to the checkout process.</td><td>Product Selected: "Wallet"</td><td>After step 4, the system should add the selected product to the user's cart or initiate the purchase process.
After step 5, the user should be directed to the checkout page or a confirmation page to complete the purchase.
The user should see accurate product details and pricing during the checkout process.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition:
1. The user should receive a confirmation of the purchase.
2. The purchased product should be recorded in the user's order history.</td></tr>
</table>

7.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 7</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High.</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: Make payment.</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Selecting Payment Method to Confirm Order.</td></tr>
<tr><td colspan="5">Description: Test to verify that the system requires users to select a payment method to confirm their order during the checkout process.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user has added at least one product to their cart and is in the checkout process.
3. The user has logged in to their account.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in to the website using valid credentials.
2. Add a product to the cart.
3. Proceed to the checkout process.
4. Reach the payment method selection step during checkout.
5. Attempt to proceed without selecting any payment method.
6. Select a payment method (either Cash on Delivery or Bkash).
7. Proceed with the payment.</td><td>Selected Payment Method: "Cash on Delivery"</td><td>After step 5, the system should not allow the user to proceed to payment unless a payment method is selected.
After step 6, the system should accept the selected payment method and proceed with the payment process.
The selected payment method should be displayed during the checkout process.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition:
1. The user receives a confirmation of the order and payment.
2. The selected payment method is recorded with the order details.</td></tr>
</table>

8.

<table>
<tr><td colspan="4">Project Name: Ghore Ghore an E-commerce site</td><td>Test Designed by: Razib</td></tr>
<tr><td colspan="4">Test Case ID: 8</td><td>Test Designed date: 20/8/23</td></tr>
<tr><td colspan="4">Test Priority (Low, Medium, High): Medium.</td><td>Test Executed by: Razib</td></tr>
<tr><td colspan="4">Module Name: Show Cart.</td><td>Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Viewing and Interacting with the Cart.</td></tr>
<tr><td colspan="5">Description: Test to verify that the system allows users to view the products in their cart, delete saved products, and proceed to buy products from the cart page.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user is logged in to their account.
3. Products have been added to the user's cart.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in to the website using valid credentials.
2. Navigate to the cart page by clicking on the "Cart" option.
3. Review the products listed in the cart.
4. Select one of the products for deletion by clicking the "Delete" option.
5. Confirm the deletion of the selected product.
6. Select one of the products for purchase by clicking the "Buy" option.
7. Proceed to the checkout process from the cart page.</td><td>Product Selected for Deletion: "Watch"
Product Selected for Purchase: "Wallet"</td><td>After step 3, the cart page should display the products added by the user, along with their details and options for interaction.
After step 4, the selected product should be removed from the cart.
After step 6, the system should proceed to the checkout process with the selected product.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition:
1. The user's cart should reflect the changes made during the test (deletion or purchase).
2. The user should receive a confirmation of the purchase if they proceed to the checkout process.</td></tr>
</table>

9.

Project Name: Ghore Ghore an E-commerce site			Test Designed by: Razib	
Test Case ID: 9			Test Designed date: 20/8/23	
Test Priority (Low, Medium, High): Low.			Test Executed by: Razib	
Module Name: Chat Box.			Test Execution date: 23/8/23	
Test Case: Using the Chat Box to Find Solutions.				
Description: Test to verify that the system's chat box allows users to find solutions by interacting with it.				
Precondition (If any): 1. The system is accessible and functional. 2. The chat box feature is enabled and available on the website.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Open the website where the chat box is available. 2. Initiate a conversation with the chat box by typing a question or inquiry. 3. Observe the chat box's response.	User Inquiry: "How do I reset my password?"	After step 2, the chat box should respond to the user's query with relevant information, solutions, or guidance. The chat box's response should be coherent, accurate, and aligned with the user's query.	As expected	Pass
Post Condition: The user should receive the chat box's response and any relevant solutions or guidance.				

10.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 10</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: Profile</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Viewing and Changing Profile Information and Picture.</td></tr>
<tr><td colspan="5">Description: Test to verify that the system allows users to view their profile information and change their profile picture.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user is logged in to their account.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in to the website using valid credentials.
2. Navigate to the user's profile page.
3. Review the user's profile information, including name, email, and other relevant details.
4. Attempt to change the user's profile picture by uploading a new image.
5. Save the changes (if applicable).</td><td>New Profile Picture: [Choose a suitable image for testing]</td><td>After step 3, the user's profile information should be displayed accurately.
After step 4, the user should be able to upload a new profile picture.
After step 5, if the user chooses to save the changes, the new profile picture should replace the previous one.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The user's profile should reflect any changes made during the test, including the updated profile picture.</td></tr>
</table>

11.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 11</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: Edit profile.</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Editing and Updating User Profile Information.</td></tr>
<tr><td colspan="5">Description: Test to verify that the system allows users to edit their profile information and successfully update it.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user is logged in to their account.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in to the website using valid credentials.
2. Navigate to the user's profile page.
3. Click on the "Edit Profile" or similar option to enter edit mode.
4. Modify the user's profile information, such as name, email, contact details, etc.
5. Save the changes.</td><td>New Email: "md.razib215@example.com"
New Contact Number: "01302221426"</td><td>After step 3, the user should be able to enter edit mode for their profile information.
After step 4, the user should be able to modify the profile information.
After step 5, the changes made to the profile information should be saved and reflected in the user's profile.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The user's profile should reflect any changes made during the test.</td></tr>
</table>

12.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 12</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: Change password</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Changing User Password</td></tr>
<tr><td colspan="5">Description: Test to verify that the system allows users to change their password using their old password for authentication.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user is logged in to their account.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in to the website using valid credentials.
2. Navigate to the user's profile page or account settings.
3. Locate the option to change the password and click on it.
4. Enter the old password, the new password, and confirm the new password.
5. Click the "Change Password" or similar button to submit the changes.</td><td>Old Password: 123

New Password: "NewPassword123"

Confirm New Password: "NewPassword123"</td><td>After step 3, the user should be able to access the password change functionality.
After step 4, the user should be able to enter the old password and set a new password.
After step 5, the password change should be successful, and the user should receive a confirmation message.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The user should be able to log in using the new password for subsequent sessions.</td></tr>
</table>

13.

<table>
<tr><td colspan="3">Project Name: Ghore Ghore an E-commerce site</td><td colspan="2">Test Designed by: Razib</td></tr>
<tr><td colspan="3">Test Case ID: 13</td><td colspan="2">Test Designed date: 20/8/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Razib</td></tr>
<tr><td colspan="3">Module Name: System Log out.</td><td colspan="2">Test Execution date: 23/8/23</td></tr>
<tr><td colspan="5">Test Case: Logging Out from the System.</td></tr>
<tr><td colspan="5">Description: Test to verify that the system allows users to log out from the website.</td></tr>
<tr><td colspan="5">Precondition (If any):
1. The system is accessible and functional.
2. The user is logged in to their account.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in to the website using valid credentials.
2. Locate the "Logout" or similar option within the system.
3. Click the "Logout" option.</td><td></td><td>After step 3, the user should be successfully logged out of the system. The user should be redirected to the website's homepage or a designated logout confirmation page.</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The user is no longer logged in to their account.</td></tr>
</table>

NuGet - Solution | UnitTest1.cs | Test Explorer

Logout | UnitTestProject1.UnitTest1 | driver

```
{
    [TestClass]
    0 references
    public class UnitTest1
    {
        private IWebDriver driver;

        [TestInitialize]
        0 references
        public void Setup()
        {
            driver = new ChromeDriver(); // You can use other drivers like FirefoxDriver, EdgeDriver, etc.
            driver.Navigate().GoToUrl("http://localhost/GoreGore/view/setting.php");
        }
        [TestCleanup]
        0 references
        public void Stop()
        {
            driver.Quit();
        }

        [TestMethod]
        | 0 references
        public void TestLoginPage()
        {
            // Find elements and interact with them

            IWebElement logoutButton = driver.FindElement(By.Id("logout"));

            Thread.Sleep(3000);
            logoutButton.Click();
            Thread.Sleep(5000);
        }
```

NuGet - Solution | UnitTest1.cs | Test Explorer

1 | 1 | 0

Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 11.3 sec

Test	Duration	Traits	Error Message
Logout (..	10.8 sec		
Unit...	10.8 sec		

Group Summary

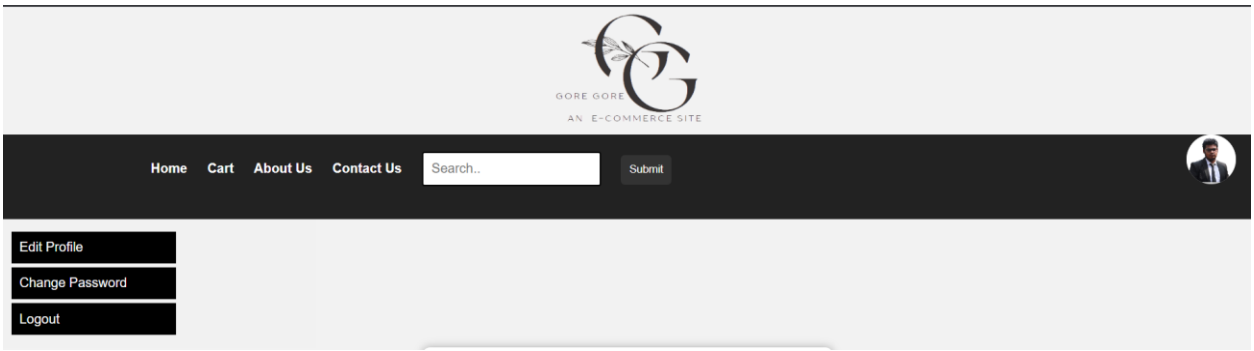
Logout

Tests in group: 1

Total Duration: 10.8 sec

Outcomes

1 Passed



8. ITEM PASS / FAIL CRITERIA

The entrance for each step of testing must be met before proceeding to the subsequent phase. The criteria for passing and failing are listed below.

1. By the stated scenario, the expected outcome must occur for the design to be deemed successful otherwise, this criterion must be failed.
2. If an item is tested ten times and functions correctly eight times, but fails once, it will be called a failure case.
3. Crashing of the website will be deemed a failure scenario.
4. After submitting a query to the system, if the desired page does not show, it will be considered a failure.

9. TEST DELIVERABLES

a) Test design specifications
b) Acceptance test plan
c) System test plan
d) Integration test plan
e) Unit test plan

10. STAFFING AND TRAINING NEEDS

This part covers personnel and test job preparation. At least one full-time tester is recommended for system/integration and acceptance testing. Most employees will embrace challenging tasks. Job descriptions follow:

1. Project Manager: Responsible for the overall project execution. This includes drafting requirements and managing the testing cycle, among other tasks. Therefore, project managers need training in these areas.

2. Test Manager: Responsible for creating expert test strategies, evaluating test deliverables, managing test cycles, and recommending testing completion. Test managers must be qualified to evaluate professional standard test designs.

3. Test Engineer: Responsible for designing tests, creating test methods, generating test data, executing tests, constructing automated test strategies, and providing the test administrator with measurement information. Test engineers should therefore be able to plan and execute any test case using automated technologies.

11. RESPONSIBILITIES

	TM	PM	DEV TEAM	TEST TEAM	CLIENT
Acceptance test Documentation & Execution.	X	X		X	X
System/Integration Test Documentation & Exec.	X		X	X	
Unit test documentation & Execution	X		X	X	
System Design Reviews	X	X	X	X	X
Detail Design Reviews	X	X	X	X	
Test procedures and rules	X	X	X	X	
Screen and Report prototype reviews			X	X	X
Change control and regression testing	X	X	X	X	X

12. TESTING SCHEDULE

1. Project Proposal
2. Requirement
3. Project Planning
4. System Design
5. Coding
6. Unit Testing
7. System, Integration, and testing
8. Acceptance testing
9. Project completion

13. PLANNING RISKS AND CONTINGENCIES

1. **Technical Glitches:**
    - **Risk:** Unforeseen technical issues or glitches could disrupt the website's functionality.
    - **Contingency:** Maintain a responsive technical support team to quickly identify and address technical problems. Regularly test the website's functionality across different devices and browsers.
2. **Cybersecurity Threats:**
    - **Risk:** The website may become a target for cyberattacks, data breaches, or hacking attempts.
    - **Contingency:** Implement robust cybersecurity measures, conduct regular security audits, and have a clear incident response plan in place to mitigate and respond to potential breaches.
3. **Traffic Surges:**
    - **Risk:** Sudden spikes in website traffic could lead to server overload and performance issues.
    - **Contingency:** Use scalable hosting solutions and content delivery networks to manage increased traffic. Monitor server performance and be prepared to scale up resources temporarily during anticipated traffic surges.

14. APPROVALS

Project Sponsor – Shishir	Approved
Development Management- Razu	Approved
EDI Project Manager- Piyal	Approved
RS Test Manager- Robiul	Approved
RS Development Team Manager- Shishir	Approved
Reassigned Sales- Razu	Approved
Order Entry EDI Team Manager- Robiul	Approved