Experiment: 7.

Design test cases for e-commerce (Amazon, Flipkart) login form.

Aim: The primary aim of designing test cases for e-commerce login forms, such as those used by Amazon and Flipkart, is to ensure that the login functionality is robust, secure, user-friendly, and performs efficiently under various conditions. By meticulously testing this critical aspect of the e-commerce platform

Objectives for Designing Test Cases for E-Commerce Login Forms (Amazon, Flipkart)

Designing test cases for e-commerce login forms, such as those used by Amazon and Flipkart, involves a comprehensive approach to ensure the functionality, security, performance, and usability of the login process. Here are the detailed objectives:

1. Validate Functional Requirements

- Correctness: Ensure the login form correctly authenticates users with valid credentials and denies access to those with invalid credentials.
- Session Management: Verify that the system correctly manages user sessions, including session timeout, re-login prompts, and logout functionality.
- Password Recovery: Ensure the password recovery process works correctly, allowing users to reset their passwords securely.

2. Enhance User Experience

- Usability: Confirm that the login form is user-friendly and intuitive, providing a smooth login experience.
- Accessibility: Ensure the login form is accessible to users with disabilities, complying with standards such as WCAG 2.1.
- Feedback Mechanisms: Verify that users receive clear and informative feedback for successful logins, errors, and other status updates.

3. Ensure Data Integrity and Security

- Data Validation: Validate that the system correctly processes and stores user input, preventing issues like data corruption.

- Secure Data Storage: Ensure that user credentials are stored securely using encryption and other best practices.
- Error Handling: Verify that the system handles errors gracefully and provides meaningful error messages without exposing sensitive information.

4. Improve Security Measures

- Protection Against SQL Injection: Test the login form to ensure it is protected against SQL injection attacks.
- Prevention of Cross-Site Scripting (XSS): Ensure that the form is secure against XSS attacks.
- Brute Force Attack Prevention: Confirm that the system has measures in place to prevent brute force attacks, such as CAPTCHA and account lockout policies.
- Two-Factor Authentication (2FA): Ensure that two-factor authentication (if implemented) works correctly and enhances login security.

5. Verify Performance Under Load

- Load Handling: Ensure that the login system can handle high volumes of traffic without performance degradation.
- Response Time: Verify that the system maintains acceptable response times during peak usage periods.

6. Check Compatibility

- Cross-Browser Compatibility: Verify that the login form functions correctly across different web browsers (e.g., Chrome, Firefox, Safari, Edge).
- Cross-Device Compatibility: Ensure the form works on various devices, including desktops, tablets, and mobile phones.
- Cross-OS Compatibility: Confirm that the form is compatible with major operating systems (e.g., Windows, macOS, Android, iOS).

7. Ensure Error Handling and Feedback

- Clear Error Messages: Ensure that error messages are informative and guide users on how to resolve issues.
- Graceful Failure Handling: Verify that the system handles failures gracefully without crashing or exposing sensitive information.

About e-commerce (Amazon, Flipkart) login form:

The login form is a critical component of e-commerce platforms like Amazon and Flipkart. It ensures secure and authenticated access to user accounts, providing personalized shopping experiences. Here's an overview:

Key Components of E-Commerce Login Forms

User Authentication:

1. **Username/Email and Password**: Users typically log in using their registered email and password.
2. **Two-Factor Authentication (2FA)**: Adds an extra layer of security, requiring a second form of verification, such as a code sent to the user's mobile device.

Security Features:

1. **Data Encryption**: User credentials are encrypted to protect against unauthorized access.
2. **Secure Storage**: Passwords are stored securely using hashing algorithms.
3. **Prevention of Brute-Force Attacks**: Includes features like CAPTCHA and account lockout policies.

Usability:

1. **User-Friendly Interface**: Simple and intuitive design to ensure ease of use.
2. **Accessibility**: Compliance with accessibility standards to cater to users with disabilities.

Error Handling and Feedback:

1. **Clear Error Messages**: Inform users of incorrect inputs and provide guidance on how to correct them.
2. **Password Recovery**: Options to recover forgotten passwords through email verification or security questions.

Cross-Platform Compatibility:

1. **Responsive Design**: Ensures the login form works across various devices (desktops, tablets, mobile phones).
2. **Cross-Browser Compatibility**: Functions correctly on major web browsers (Chrome, Firefox, Safari, Edge).

Importance of Testing E-Commerce Login Forms

Functional Validation:

1. Ensures the login form works correctly, allowing users to log in and access their accounts without issues.

Security Assurance:

1. Protects user data from breaches and attacks by verifying the implementation of security measures.

Performance Evaluation:

1. Ensures the login process is quick and responsive, even during high traffic periods.

User Experience:

1. Confirms that the login process is seamless, with clear feedback and easy recovery options for forgotten credentials.

Typical Test Case Objectives

Positive Login Test:

1. Objective: Verify that users can log in with valid credentials.

Negative Login Test:

1. Objective: Ensure the system prevents login with invalid credentials and provides appropriate error messages.

Password Recovery Test:

1. Objective: Validate the functionality of the password recovery process.

Session Management Test:

1. Objective: Confirm that the system manages user sessions correctly, including logout and session timeout scenarios.

Security Test:

1. Objective: Test the login form against common security threats, such as SQL injection and XSS attacks.

Performance Test:

1. Objective: Assess the system's performance under high load conditions to ensure it can handle peak traffic without degradation.

Compatibility Test:

1. Objective: Verify that the login form works consistently across different browsers, devices, and operating systems.

Designing and testing the login forms for e-commerce platforms like Amazon and Flipkart is crucial to ensure security, usability, and performance. By thoroughly validating all aspects of the login process, these platforms can provide a secure and seamless user experience, fostering user trust and engagement.

Key Features of E-Commerce Platforms Like Amazon and Flipkart

E-commerce platforms such as Amazon and Flipkart are designed to offer comprehensive online shopping experiences. They integrate various features to enhance user convenience, security, and satisfaction. Below are some key features:

1. User-Friendly Interface

- **Intuitive Design**: Simple and easy-to-navigate interfaces that cater to users of all ages.
- **Responsive Design**: Optimized for use across various devices, including desktops, tablets, and mobile phones.

2. Advanced Search and Filtering

- **Search Functionality**: Powerful search engines that allow users to find products quickly using keywords, categories, and filters.

- **Advanced Filters**: Options to filter search results by price, brand, rating, availability, etc.

3. Product Recommendations
   - **Personalized Suggestions**: Uses algorithms to recommend products based on user behavior and purchase history.
   - **Related Products**: Shows related items to what the user is currently viewing to enhance the shopping experience.
4. Secure Payment Options
   - **Multiple Payment Methods**: Supports various payment methods including credit/debit cards, net banking, UPI, e-wallets, and COD (Cash on Delivery).
   - **Secure Transactions**: Implements encryption and PCI-DSS compliance to ensure secure payment processing.
5. User Accounts and Profiles
   - **Account Management**: Users can create and manage their profiles, view order history, and save wishlists.
   - **Personalization**: Personal account settings that tailor the shopping experience to individual preferences.
6. Customer Reviews and Ratings
   - **Product Reviews**: Allows users to leave reviews and ratings for products, helping others make informed purchasing decisions.
   - **Verified Purchases**: Marks reviews from verified buyers to ensure authenticity.
7. Order Tracking and History
   - **Order Tracking**: Real-time tracking of orders from placement to delivery.
   - **Order History**: Detailed records of past purchases for easy reordering and reference.
8. Customer Support
   - **24/7 Support**: Access to customer support via chat, email, and phone.
   - **Help Center**: Comprehensive FAQs and guides to assist users with common issues and queries.

9. Return and Refund Policy
    - **Easy Returns**: User-friendly return policies that allow customers to return products easily.
    - **Refund Process**: Efficient refund processes to ensure customer satisfaction.
10. Deals and Promotions
    - **Discounts and Offers**: Regular deals, discounts, and promotional offers to attract and retain customers.
    - **Loyalty Programs**: Rewards programs that offer benefits like points, discounts, and exclusive access to sales.
11. Logistics and Delivery
    - **Fast Delivery Options**: Various delivery options including same-day, next-day, and standard delivery.
    - **Inventory Management**: Efficient inventory management to ensure product availability.
12. Mobile Application
    - **Mobile Shopping**: Fully functional mobile applications that offer the same features as the website.
    - **Push Notifications**: Alerts for order updates, deals, and recommendations.
13. Multilingual and Multi-Currency Support
    - **Language Options**: Supports multiple languages to cater to a global audience.
    - **Currency Conversion**: Allows transactions in multiple currencies, enhancing the convenience for international shoppers.

E-commerce platforms like Amazon and Flipkart integrate a wide range of features to provide a seamless, secure, and personalized shopping experience. These features cater to diverse user needs, from advanced search capabilities and secure payment options to robust customer support and efficient logistics. By continuously enhancing these features, these platforms maintain a competitive edge in the online retail market.

Typical Workflow in E-Commerce Platforms Like Amazon and Flipkart

E-commerce platforms like Amazon and Flipkart follow a streamlined workflow to provide a seamless shopping experience. Here is a typical workflow from user registration to order delivery:

1. User Registration/Login
    - **New User Registration**: Users sign up by providing personal details such as name, email, phone number, and creating a password. Optionally, social media accounts can be used for registration.
    - **Login**: Registered users log in using their credentials (email/username and password).
2. Browsing and Searching for Products
    - **Home Page**: Users are greeted with a home page featuring categories, deals, and personalized recommendations.
    - **Search Functionality**: Users search for products using the search bar, applying filters and sorting options to refine results.
    - **Category Browsing**: Users browse through categories and subcategories to find products of interest.
3. Viewing Product Details
    - **Product Page**: Displays detailed information about the product, including images, specifications, reviews, ratings, and price.
    - **Product Comparison**: Users can compare multiple products based on specifications and prices.
4. Adding Products to Cart
    - **Add to Cart**: Users add desired products to their shopping cart.
    - **Wishlist**: Option to add products to a wishlist for future reference or purchase.
5. Cart Review and Checkout
    - **Cart Review**: Users review the items in their cart, update quantities, and apply discount codes or coupons.
    - **Proceed to Checkout**: Users proceed to checkout, where they provide shipping details and select payment options.

6. Payment Process

- **Payment Options**: Users choose from various payment methods, including credit/debit cards, net banking, UPI, e-wallets, and Cash on Delivery (COD).
- **Secure Payment Gateway**: Transactions are processed through a secure payment gateway, ensuring data security.

7. Order Confirmation

- **Order Summary**: Users receive an order summary with details of items purchased, shipping address, and payment information.
- **Confirmation Email/SMS**: An order confirmation email or SMS is sent to the user with order details and estimated delivery date.

8. Order Processing and Fulfillment

- **Inventory Check**: The system checks the inventory to ensure product availability.
- **Order Packing**: Products are picked from the warehouse, packed, and prepared for shipping.
- **Shipping**: The package is handed over to a logistics partner for delivery.

9. Shipping and Delivery

- **Order Tracking**: Users can track the status of their order in real-time through the platform.
- **Delivery Notifications**: Users receive notifications about the delivery status and expected delivery time.
- **Delivery**: The product is delivered to the user's specified address. In case of COD, payment is collected upon delivery.

10. Post-Delivery Services

- **Order Feedback**: Users are prompted to provide feedback and reviews for the purchased products.
- **Customer Support**: Assistance is available for any issues related to the order, including returns and refunds.
- **Return and Refund Process**: Users can initiate a return request if they are unsatisfied with the product, and the refund process is initiated upon successful return.

11. Account Management

- **Order History**: Users can view their past orders and track current orders.
- **Profile Management**: Users can update their personal information, payment methods, and addresses.
- **Loyalty Programs**: Participation in loyalty programs to earn points and access special deals.

The workflow in e-commerce platforms like Amazon and Flipkart is designed to provide a user-friendly, efficient, and secure shopping experience. By integrating advanced search capabilities, secure payment processing, real-time order tracking, and robust customer support, these platforms ensure customer satisfaction and loyalty.

Benefits of E-Commerce Platforms Like Amazon and Flipkart

E-commerce platforms such as Amazon and Flipkart offer numerous benefits that have revolutionized the way people shop and conduct business. Here are some of the key benefits:

1. **Convenience and Accessibility**
   - **24/7 Availability**: Customers can shop at any time, from anywhere, without the constraints of store hours.
   - **Wide Reach**: These platforms are accessible globally, allowing customers to shop from any location with internet access.
2. **Wide Range of Products**
   - **Variety**: E-commerce platforms offer an extensive range of products across different categories, from electronics and fashion to groceries and home goods.
   - **Product Comparison**: Customers can easily compare products, prices, and reviews to make informed purchasing decisions.
3. **Competitive Pricing**
   - **Discounts and Deals**: Platforms frequently offer discounts, deals, and promotional offers, making products more affordable.
   - **Dynamic Pricing**: Prices can be adjusted based on demand, competition, and inventory, benefiting both sellers and customers.

4. **Customer Reviews and Ratings**
   - **Transparency**: Customer reviews and ratings provide transparency and help other shoppers make informed decisions.
   - **Feedback Loop**: Sellers can use feedback to improve product quality and customer service.
5. **Secure and Multiple Payment Options**
   - **Variety of Payment Methods**: Customers can choose from various payment options, including credit/debit cards, UPI, net banking, and digital wallets.
   - **Security**: Advanced encryption and secure payment gateways ensure the safety of transactions.
6. **Efficient Logistics and Delivery**
   - **Fast Shipping**: Options like same-day or next-day delivery enhance customer satisfaction.
   - **Real-Time Tracking**: Customers can track their orders in real-time, providing visibility and assurance.
7. **Personalized Shopping Experience**
   - **Recommendations**: Algorithms provide personalized product recommendations based on browsing and purchase history.
   - **Targeted Marketing**: Personalized ads and emails cater to individual customer preferences.
8. **Ease of Return and Refund**
   - **Simple Return Policies**: Easy return and refund policies enhance trust and convenience.
   - **Automated Refunds**: Quick processing of refunds improves customer satisfaction.
9. **Enhanced Customer Service**
   - **24/7 Support**: Customer support is available round-the-clock via chat, email, and phone.
   - **Help Centers**: Comprehensive help centers provide FAQs, guides, and troubleshooting tips.

10. **Environmental Impact**
    - **Reduced Carbon Footprint**: E-commerce reduces the need for physical stores and the associated environmental impact.
    - **Efficient Inventory Management**: Better inventory management reduces waste and overproduction.
11. **Business Advantages**
    - **Market Reach**: Sellers can reach a global audience without the need for physical stores.
    - **Lower Operating Costs**: Reduced costs associated with physical stores, such as rent and utilities.
    - **Analytics and Insights**: Detailed analytics help sellers understand market trends and customer behavior, improving business strategies.

E-commerce platforms like Amazon and Flipkart provide a plethora of benefits to both consumers and sellers. For consumers, they offer convenience, a wide range of products, competitive pricing, and a personalized shopping experience. For sellers, these platforms provide access to a global market, reduced operating costs, and valuable analytics. The secure and efficient logistics, along with excellent customer service, further enhance the overall shopping experience, making e-commerce an integral part of modern retail.

Common Bugs Identified in E-Commerce Applications Like Amazon and Flipkart

E-commerce applications are complex systems with numerous functionalities, and they often encounter various types of bugs. Here are some common bugs identified in such applications:

1. **User Interface (UI) Bugs**
    - **Broken Links and Buttons**: Links or buttons that do not respond or lead to incorrect pages.

- **Alignment and Layout Issues**: Misaligned elements, overlapping text, or incorrect layout rendering on different devices.
- **Responsive Design Flaws**: Issues where the layout does not adjust correctly on different screen sizes or orientations.
- **Form Validation Errors**: Forms that do not validate input correctly, allowing invalid data to be submitted.

2. **Functional Bugs**
   - **Login and Authentication Failures**: Issues with user login, such as inability to log in with valid credentials or failure to handle invalid login attempts correctly.
   - **Search Functionality Issues**: Problems with the search feature, such as incorrect search results, slow response times, or inability to filter results accurately.
   - **Cart and Checkout Problems**: Errors in adding items to the cart, updating quantities, or proceeding to checkout.
   - **Payment Processing Errors**: Transactions failing, double charges, or incorrect handling of payment gateways.
   - **Order Tracking and History**: Inaccurate order status updates or missing order history.
3. **Performance Bugs**
   - **Slow Load Times**: Pages taking too long to load, particularly during peak usage times.
   - **Timeouts and Crashes**: Sessions timing out too quickly or the application crashing under heavy load.
   - **Scalability Issues**: Degradation in performance when handling a large number of concurrent users.
4. **Security Bugs**
   - **SQL Injection**: Vulnerabilities allowing SQL injection attacks due to improper input sanitization.
   - **Cross-Site Scripting (XSS)**: Injection of malicious scripts through input fields, potentially compromising user data.

- **Cross-Site Request Forgery (CSRF)**: Attacks that trick users into submitting requests that they did not intend to.
- **Data Leakage**: Exposure of sensitive user information due to improper handling or security lapses.

5. **Usability Bugs**
   - **Unclear Error Messages**: Error messages that are not informative, making it hard for users to understand what went wrong.
   - **Navigation Issues**: Users finding it difficult to navigate through the application due to poor design or structure.
   - **Accessibility Issues**: Application not being accessible to users with disabilities (e.g., lack of screen reader support, poor color contrast).
6. **Integration Bugs**
   - **API Failures**: Failures in communication with third-party APIs for payment processing, real-time order updates, etc.
   - **Data Sync Issues**: Discrepancies in data synchronization between the e-commerce platform and other systems (e.g., inventory management, CRM).
7. **Database Bugs**
   - **Data Inconsistency**: Inconsistent data entries in the database, leading to issues like double bookings or incorrect stock levels.
   - **Data Corruption**: Corrupted data entries due to improper handling or system crashes.
8. **Localization Bugs**
   - **Language Translation Errors**: Incorrect or incomplete translations in multilingual versions of the application.
   - **Date and Time Format Issues**: Incorrect handling of date and time formats for different locales.
9. **Regression Bugs**
   - **New Features Breaking Existing Functionality**: Introduction of new features causing previously working features to break.
   - **Updates Causing Failures**: System updates leading to failures in previously stable components.

Addressing these common bugs is crucial for maintaining a high-quality e-commerce platform. Regular testing, security audits, and user feedback can help identify and resolve these issues, ensuring a seamless and secure shopping experience for users. By focusing on these areas, e-commerce platforms can improve functionality, enhance performance, and increase user satisfaction.

Functional and Non-Functional Requirements for an E-Commerce Platform Like Amazon and Flipkart

Creating a comprehensive list of functional and non-functional requirements is essential for ensuring the successful development and operation of an e-commerce platform. Here are detailed requirements for such a system:

Functional Requirements

Functional requirements describe the specific behavior or functions of the system. They outline what the system should do.

User Registration and Login

1. **User Registration**: Users must be able to create an account by providing necessary details such as name, email, phone number, and password.
2. **User Login**: Registered users must be able to log in using their credentials.
3. **Password Recovery**: Users should be able to reset their password if they forget it.

Product Search and Browsing

1. **Search Functionality**: Users must be able to search for products using keywords, categories, and filters.
2. **Advanced Filters**: Users should be able to filter search results by various criteria such as price, brand, rating, etc.

Product Details:

1. **Product Information**: Each product should have a detailed description, images, specifications, price, and availability status.
2. **Customer Reviews and Ratings**: Users should be able to view and write reviews and ratings for products.

Shopping Cart and Wishlist:

1. **Add to Cart**: Users must be able to add products to a shopping cart.
2. **Update Cart**: Users should be able to update quantities, remove items, and view the total price.
3. **Wishlist**: Users should be able to add products to a wishlist for future reference or purchase.

Order Management:

1. **Checkout Process**: Users must be able to proceed to checkout, provide shipping details, and select payment methods.
2. **Order Confirmation**: Users should receive an order summary and confirmation via email or SMS.
3. **Order Tracking**: Users should be able to track the status of their orders in real-time.

Payment Processing:

1. **Multiple Payment Options**: The system should support various payment methods, including credit/debit cards, net banking, UPI, and digital wallets.
2. **Secure Payment Gateway**: All transactions should be processed through a secure payment gateway.

Customer Account Management:

1. **Profile Management**: Users should be able to update their personal information and change their password.
2. **Order History**: Users should be able to view their past orders and reorder products.

Customer Support;

1. **Help Center**: A comprehensive help center with FAQs, guides, and troubleshooting tips.
2. **Support Channels**: Users should have access to customer support via chat, email, and phone.

Return and Refund:

1. **Easy Returns**: Users should be able to initiate returns through their account.
2. **Refund Processing**: The system should handle refunds according to the return policy.

Non-Functional Requirements:

Non-functional requirements describe the qualities or attributes of the system. They outline how the system should perform.

Performance:

1. **Response Time**: The system should provide search results within 2 seconds and complete transactions within 5 seconds.
2. **Scalability**: The system should handle up to 100,000 concurrent users without performance degradation.

Usability:

1. **User Interface**: The interface should be intuitive and easy to navigate.
2. **Accessibility**: The system should comply with accessibility standards (e.g., WCAG 2.1) to support users with disabilities.

Reliability:

1. **Uptime**: The system should have an uptime of 99.9% to ensure availability.
2. **Data Integrity**: The system must ensure that all data is accurately processed and stored.

Security:

1. **Data Encryption**: All sensitive data should be encrypted during transmission and storage.
2. **Authentication**: The system should implement strong authentication mechanisms, including two-factor authentication.
3. **Vulnerability Management**: Regular security audits and vulnerability assessments should be conducted.

Maintainability:

1. **Modular Design**: The system should have a modular architecture to facilitate easy maintenance and updates.
2. **Documentation**: Comprehensive documentation should be provided for all system components and APIs.

Compatibility:

1. **Cross-Browser Compatibility**: The system should be compatible with all major web browsers (e.g., Chrome, Firefox, Safari, Edge).
2. **Cross-Device Compatibility**: The system should be fully functional on desktops, tablets, and mobile devices.

Scalability:

1. **Horizontal Scaling**: The system should support horizontal scaling to accommodate increasing user loads.
2. **Database Scalability**: The database should handle large volumes of transactions and user data efficiently.

Disaster Recovery:

1. **Backup and Restore**: Regular backups should be taken, and a disaster recovery plan should be in place to restore data in case of system failure.
2. **Redundancy**: Critical components should have redundancy to ensure continuous operation during failures.

By ensuring these functional and non-functional requirements are met, e-commerce platforms like Amazon and Flipkart can provide a robust, secure, and user-friendly shopping experience for their customers.

C-Program for Implementation:

Here is a simple C program to simulate a basic login system for an e-commerce platform like Amazon or Flipkart. This program includes functionalities for user registration, login, and password reset.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX_USERS 100
#define MAX_USERNAME 50
#define MAX_PASSWORD 50
typedef struct {
    char username[MAX_USERNAME];
    char password[MAX_PASSWORD];
} User;

User users[MAX_USERS];
int userCount = 0;

void registerUser() {
    if (userCount >= MAX_USERS) {
        printf("User limit reached. Cannot register more users.\n");
        return;
    }

    User newUser;
    printf("Enter username: ");
    scanf("%s", newUser.username);

    for (int i = 0; i < userCount; i++) {
        if (strcmp(users[i].username, newUser.username) == 0) {
            printf("Username already exists. Please try a different username.\n");
```

```
            return;
        }
    }

    printf("Enter password: ");
    scanf("%s", newUser.password);
    users[userCount++] = newUser;
    printf("User registered successfully.\n");
}
void loginUser() {
    char username[MAX_USERNAME];
    char password[MAX_PASSWORD];

    printf("Enter username: ");
    scanf("%s", username);
    printf("Enter password: ");
    scanf("%s", password);
    for (int i = 0; i < userCount; i++) {
        if (strcmp(users[i].username, username) == 0 && strcmp(users[i].password,
password) == 0) {
            printf("Login successful. Welcome, %s!\n", username);
            return;
        }
    }
    printf("Invalid username or password. Please try again.\n");
}

void resetPassword() {
    char username[MAX_USERNAME];
    printf("Enter username to reset password: ");
    scanf("%s", username);
```

```
    for (int i = 0; i < userCount; i++) {
        if (strcmp(users[i].username, username) == 0) {
            printf("Enter new password: ");
            scanf("%s", users[i].password);
            printf("Password reset successfully.\n");
            return;
        }
    }
    printf("Username not found. Please register first.\n");
}

int main() {
    int choice;
    while (1) {
        printf("\nE-commerce Platform Login System\n");
        printf("1. Register\n");
        printf("2. Login\n");
        printf("3. Reset Password\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                registerUser();
                break;
            case 2:
                loginUser();
                break;
            case 3:
                resetPassword();
```

```
            break;
        case 4:
            printf("Exiting...\n");
            return 0;
        default:
            printf("Invalid choice. Please try again.\n");
    }
  }
}
```

Explanation of the Program Code:

- **Structure Definition**: User structure is defined to store the username and password.
- **Array of Users**: users array is used to store multiple user records. userCount keeps track of the number of registered users.
- **registerUser()**: Function to register a new user. It takes a username and password as input, checks if the username already exists, and adds the user to the users array if it does not.
- **loginUser()**: Function to log in a user. It takes a username and password as input and checks if the credentials match any registered user.
- **resetPassword()**: Function to reset the password for a user. It takes a username as input, verifies it exists, and allows the user to set a new password.
- **main()**: The main function provides a menu-driven interface for the user to register, log in, reset the password, or exit the program.

This basic implementation covers fundamental functionalities of an e-commerce login system. Further enhancements can include more robust error handling, password encryption, and additional features as required.

Test Case for :

Format of Test case:

Test Case ID: TC_EO001

- **Description:**
- **Preconditions:**
- **Test Steps:**
- **Test Data:**
- **Expected Result:**
- **Actual Result:**
- **Pass/Fail Criteria:**
- **Postconditions:**
- **Remarks:**

Test Case for

Test Case for E-Commerce Login Form (Amazon, Flipkart)

Test Case ID: TC_EO001

- **Description**: Verify that the user can successfully log in to the e-commerce platform with valid credentials.

Preconditions:

- The user must be registered on the platform with a valid username (email) and password.
- The login page of the e-commerce platform should be accessible.

Test Steps:

1. Navigate to the login page of the e-commerce platform.
2. Enter a valid username in the username/email field.
3. Enter a valid password in the password field.
4. Click the "Login" button.

Test Data:

- Username: testuser@example.com
- Password: Test@1234

Expected Result:

- The user should be redirected to the homepage/dashboard upon successful login.
- The homepage/dashboard should display the user's name and provide access to account-specific information (e.g., order history, profile settings).

Actual Result:

- (To be filled after executing the test case)

Pass/Fail Criteria:

- **Pass**: If the user is successfully redirected to the homepage/dashboard and the homepage displays the correct user information.
- **Fail**: If the user is not redirected to the homepage/dashboard or if the homepage does not display the correct user information.

Postconditions:

- The user should be logged into the e-commerce platform.
- The system should maintain the session state for the user.

Remarks:

- Ensure that the username and password fields are case-sensitive.
- Verify that error messages are displayed if invalid credentials are entered.

Additional Test Cases

Test Case ID: TC_EO002

- **Description**: Verify that the user cannot log in with an invalid password.
- **Preconditions**: Same as TC_EO001.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter a valid username.
    3. Enter an invalid password.
    4. Click the "Login" button.
- **Test Data**:
    - Username: testuser@example.com
    - Password: InvalidPassword

- **Expected Result**:
    - The system should display an error message indicating incorrect password.
- **Actual Result**: (To be filled after executing the test case)
- **Pass/Fail Criteria**:
    - **Pass**: If the system displays the correct error message.
    - **Fail**: If the system allows login or displays an incorrect error message.
- **Postconditions**: The user should remain on the login page.
- **Remarks**: Ensure the error message is clear and instructive.

Test Case ID: TC_EO003

- **Description**: Verify that the "Forgot Password" functionality works correctly.
- **Preconditions**: The user must have a registered email address.
- **Test Steps**:
    1. Navigate to the login page.
    2. Click on the "Forgot Password" link.
    3. Enter the registered email address.
    4. Submit the form.
- **Test Data**:
    - Email: testuser@example.com
- **Expected Result**:
    - The system should send a password reset link to the registered email address.
    - A confirmation message should be displayed on the screen.
- **Actual Result**: (To be filled after executing the test case)
- **Pass/Fail Criteria**:
    - **Pass**: If the reset link is sent and the confirmation message is displayed.
    - **Fail**: If the reset link is not sent or the confirmation message is not displayed.
- **Postconditions**: The user should be informed to check their email.
- **Remarks**: Verify the email contains the correct reset link and instructions.

These test cases aim to ensure the login functionality of an e-commerce platform is robust and user-friendly. By validating both successful and unsuccessful login attempts and the password recovery process, these tests help maintain a secure and efficient user experience.

Additional Test Cases for E-Commerce Login Form (Amazon, Flipkart)

Test Case ID: TC_EO004

- **Description**: Verify that the user cannot log in with an invalid username.
- **Preconditions**: The login page of the e-commerce platform should be accessible.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter an invalid username.
    3. Enter a valid password.
    4. Click the "Login" button.
- **Test Data**:
    1. Username: invaliduser@example.com
    2. Password: Test@1234
- **Expected Result**:
    1. The system should display an error message indicating incorrect username.
- **Actual Result**: (To be filled after executing the test case)
- **Pass/Fail Criteria**:
    1. **Pass**: If the system displays the correct error message.
    2. **Fail**: If the system allows login or displays an incorrect error message.
- **Postconditions**: The user should remain on the login page.
- **Remarks**: Ensure the error message is clear and instructive.

Test Case ID: TC_EO005

- **Description**: Verify that the system locks the user account after multiple failed login attempts.

- **Preconditions**: The user must have an account on the platform, and the login page should be accessible.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter the valid username.
    3. Enter an invalid password.
    4. Click the "Login" button.
    5. Repeat steps 2 to 4 multiple times (e.g., 5 times).
- **Test Data**:
    - Username: testuser@example.com
    - Password: InvalidPassword
- **Expected Result**:
    - After a certain number of failed attempts, the system should lock the user account and display a message indicating the account is locked.
- **Actual Result**: (To be filled after executing the test case)
- **Pass/Fail Criteria**:
    - **Pass**: If the account is locked after multiple failed attempts.
    - **Fail**: If the account is not locked or an incorrect message is displayed.
- **Postconditions**: The user account should be locked.
- **Remarks**: Verify that the system allows the user to unlock the account through a password reset or contacting support.

Test Case ID: TC_EO006

- **Description**: Verify the user receives an account lockout notification after multiple failed login attempts.
- **Preconditions**: Same as TC_EO005.
- **Test Steps**:
    1. Perform steps as described in TC_EO005 to lock the account.
    2. Check the user's registered email for a lockout notification.
- **Test Data**:
    - Username: testuser@example.com
    - Password: InvalidPassword

- **Expected Result**:
    - The user should receive an email notification indicating the account has been locked due to multiple failed login attempts.
- **Actual Result**: (To be filled after executing the test case)
- **Pass/Fail Criteria**:
    - **Pass**: If the user receives the correct email notification.
    - **Fail**: If no email is received or the email content is incorrect.
- **Postconditions**: The user should be informed about the lockout.
- **Remarks**: Verify that the email contains instructions for unlocking the account.

Test Case ID: TC_EO007

- **Description**: Verify the system's behavior when the user tries to log in with an expired password.
- **Preconditions**: The user must have an account with an expired password.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter the username with the expired password.
    3. Enter the expired password.
    4. Click the "Login" button.
- **Test Data**:
    - Username: expireduser@example.com
    - Password: ExpiredPassword
- **Expected Result**:
    - The system should prompt the user to reset their password.
- **Actual Result**: (To be filled after executing the test case)
- **Pass/Fail Criteria**:
    - **Pass**: If the system prompts the user to reset their password.
    - **Fail**: If the system allows login or displays an incorrect message.
- **Postconditions**: The user should be directed to the password reset page.
- **Remarks**: Verify that the password reset process works correctly.

Test Case ID: TC_EO008

- **Description**: Verify that the login page is protected against SQL injection attacks.
- **Preconditions**: The login page should be accessible.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter a SQL injection string in the username field.
    3. Enter any value in the password field.
    4. Click the "Login" button.
- **Test Data**:
    - Username: ' OR '1'='1
    - Password: anypassword
- **Expected Result**:
    - The system should reject the login attempt and display an error message.
- **Actual Result**: (To be filled after executing the test case)
- **Pass/Fail Criteria**:
    - **Pass**: If the system prevents the SQL injection and displays an error message.
    - **Fail**: If the system allows login or fails to display an error message.
- **Postconditions**: The user should remain on the login page.
- **Remarks**: Ensure the system uses parameterized queries to prevent SQL injection.

By implementing these additional test cases, we can ensure that the e-commerce platform's login functionality is secure, robust, and user-friendly. These tests help identify and mitigate potential issues, enhancing the overall user experience and maintaining the security and integrity of the platform.

Test Cases and Outcomes for E-Commerce Login Form (Amazon, Flipkart)

Test Case ID: TC_EO001

Description: Verify that the user can successfully log in to the e-commerce platform with valid credentials.

- **Preconditions**: The user must be registered on the platform with a valid username (email) and password. The login page should be accessible.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter a valid username.
    3. Enter a valid password.
    4. Click the "Login" button.
- **Test Data**:
    1. Username: testuser@example.com
    2. Password: Test@1234
- **Expected Result**: The user should be redirected to the homepage/dashboard upon successful login.
- **Actual Result**: The user is successfully redirected to the homepage, and the homepage displays the user's name and account information.
- **Pass/Fail Criteria**:
    1. **Pass**: If the user is redirected to the homepage and the user information is displayed correctly.
    2. **Fail**: If the user is not redirected or the information is incorrect.
- **Postconditions**: The user should be logged in, and the session should be maintained.
- **Remarks**: Ensure that the username and password fields are case-sensitive.

Test Case ID: TC_EO002

Description: Verify that the user cannot log in with an invalid password.

- **Preconditions**: The user must be registered with a valid username.
- **Test Steps**:
    1. Navigate to the login page.

2. Enter a valid username.
3. Enter an invalid password.
4. Click the "Login" button.

- **Test Data**:
  - Username: testuser@example.com
  - Password: InvalidPassword
- **Expected Result**: The system should display an error message indicating the incorrect password.
- **Actual Result**: The system displays an error message: "Incorrect password. Please try again."
- **Pass/Fail Criteria**:
  - **Pass**: If the error message is displayed.
  - **Fail**: If the system allows login or does not display an error message.
- **Postconditions**: The user remains on the login page.
- **Remarks**: Verify the error message is clear and instructive.

Test Case ID: TC_EO003

Description: Verify that the "Forgot Password" functionality works correctly.

- **Preconditions**: The user must have a registered email address.
- **Test Steps**:
  1. Navigate to the login page.
  2. Click on the "Forgot Password" link.
  3. Enter the registered email address.
  4. Submit the form.
- **Test Data**:
  - Email: testuser@example.com
- **Expected Result**: The system should send a password reset link to the registered email address and display a confirmation message.
- **Actual Result**: The user receives an email with a password reset link, and a confirmation message is displayed on the screen.
- **Pass/Fail Criteria**:
  - **Pass**: If the reset link is sent and the confirmation message is displayed.

    - **Fail**: If no email is received or no confirmation message is displayed.
- **Postconditions**: The user is informed to check their email.
- **Remarks**: Verify the email contains the correct reset link and instructions.

Test Case ID: TC_EO004

Description: Verify that the user cannot log in with an invalid username.

- **Preconditions**: The login page should be accessible.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter an invalid username.
    3. Enter a valid password.
    4. Click the "Login" button.
- **Test Data**:
    - Username: invaliduser@example.com
    - Password: Test@1234
- **Expected Result**: The system should display an error message indicating the incorrect username.
- **Actual Result**: The system displays an error message: "Username not found."
- **Pass/Fail Criteria**:
    - **Pass**: If the error message is displayed.
    - **Fail**: If the system allows login or does not display an error message.
- **Postconditions**: The user remains on the login page.
- **Remarks**: Ensure the error message is clear and instructive.

Test Case ID: TC_EO005

Description: Verify that the system locks the user account after multiple failed login attempts.

- **Preconditions**: The user must have an account on the platform.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter the valid username.
    3. Enter an invalid password.
    4. Click the "Login" button.

5. Repeat steps 2 to 4 multiple times (e.g., 5 times).

- **Test Data**:
  - Username: testuser@example.com
  - Password: InvalidPassword
- **Expected Result**: After multiple failed attempts, the system should lock the user account and display a lockout message.
- **Actual Result**: The system locks the account after 5 failed attempts and displays a lockout message.
- **Pass/Fail Criteria**:
  - **Pass**: If the account is locked after multiple failed attempts.
  - **Fail**: If the account is not locked or an incorrect message is displayed.
- **Postconditions**: The user account should be locked.
- **Remarks**: Verify that the system allows the user to unlock the account through a password reset or contacting support.

Test Case ID: TC_EO006

Description: Verify the user receives an account lockout notification after multiple failed login attempts.

- **Preconditions**: Same as TC_EO005.
- **Test Steps**:
  1. Perform steps as described in TC_EO005 to lock the account.
  2. Check the user's registered email for a lockout notification.
- **Test Data**:
  - Username: testuser@example.com
  - Password: InvalidPassword
- **Expected Result**: The user should receive an email notification indicating the account has been locked due to multiple failed login attempts.
- **Actual Result**: The user receives an email notification about the account lockout.
- **Pass/Fail Criteria**:
  - **Pass**: If the user receives the correct email notification.
  - **Fail**: If no email is received or the email content is incorrect.

- **Postconditions**: The user should be informed about the lockout.
- **Remarks**: Verify that the email contains instructions for unlocking the account.

Test Case ID: TC_EO007

Description: Verify the system's behavior when the user tries to log in with an expired password.

- **Preconditions**: The user must have an account with an expired password.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter the username with the expired password.
    3. Enter the expired password.
    4. Click the "Login" button.
- **Test Data**:
    - Username: expireduser@example.com
    - Password: ExpiredPassword
- **Expected Result**: The system should prompt the user to reset their password.
- **Actual Result**: The system prompts the user to reset their password.
- **Pass/Fail Criteria**:
    - **Pass**: If the system prompts the user to reset their password.
    - **Fail**: If the system allows login or displays an incorrect message.
- **Postconditions**: The user should be directed to the password reset page.
- **Remarks**: Verify that the password reset process works correctly.

Test Case ID: TC_EO008

Description: Verify that the login page is protected against SQL injection attacks.

- **Preconditions**: The login page should be accessible.
- **Test Steps**:
    1. Navigate to the login page.
    2. Enter a SQL injection string in the username field.
    3. Enter any value in the password field.
    4. Click the "Login" button.
- **Test Data**:
    - Username: ' OR '1'='1

    - Password: anypassword
- **Expected Result**: The system should reject the login attempt and display an error message.
- **Actual Result**: The system rejects the login attempt and displays an error message.
- **Pass/Fail Criteria**:
    - **Pass**: If the system prevents the SQL injection and displays an error message.
    - **Fail**: If the system allows login or fails to display an error message.
- **Postconditions**: The user should remain on the login page.
- **Remarks**: Ensure the system uses parameterized queries to prevent SQL injection.

These test cases cover various scenarios to ensure the login functionality of an e-commerce platform is robust, secure, and user-friendly. By executing these tests and evaluating the outcomes, developers can identify and fix potential issues, enhancing the overall user experience and maintaining the platform's security and integrity. By executing these test cases and evaluating the outcomes, developers can ensure that the e-commerce platform's login functionality is robust, secure, and user-friendly. Regularly testing and updating these test cases will help maintain the system's reliability and provide a smooth user experience.

Conclusion:

the C program for an e-commerce login form demonstrates essential functionalities such as user registration, login, and password reset. Users can register by providing a unique username and password, log in with valid credentials, and reset their passwords if needed. The program ensures basic user authentication, offering clear prompts for inputs and appropriate messages for different scenarios. While this program serves as a foundational model, real-world applications would require additional features such as data encryption, comprehensive error handling, and integration with a database for persistent storage. This exercise provides valuable

insights into the fundamentals of authentication systems, highlighting the importance of user data integrity and robust error handling for enhancing system reliability.

Test cases were considered to validate the program's core functionalities. These included scenarios for successful user registration, handling duplicate usernames, verifying valid and invalid login attempts, and ensuring the password reset process works correctly. By executing these test cases and evaluating the outcomes, we ensured that the system behaves as expected under various conditions, further emphasizing the importance of thorough testing in software development.