A decorative border resembling a scroll, with a light gray fill and a black outline. It features rounded corners and a vertical strip on the left side. The top right corner has a small, stylized scroll element.

CHAPTER-4

SYSTEM TESTING

4. TESTING

4.1 Introduction

Testing is the process of detecting errors. Testing performs a very special role for quality assurance and for ensuring the reliability of the software. Its basic function is to detect defects in software. Implementation is the process which tells the reliability, efficiency and flexibility. The result of testing is used later on during maintenance also. Testing is done to check whether the proposed system works as requested by the client. During testing is also check whether for the given input the expected output obtained or not. It also helps us in the rectification of errors in the system. There are different types of testing such as unit testing, integration testing, system testing. We used system testing and unit testing for testing the system.

4.1.1 Test Strategy

The testing strategy used will focus on reliability and performance of the product. This includes the reporting client hardware, software, management.

4.2 Types of Testing

“Program testing can be used to show the presence of bugs, but never to show their absence!” Clearly, the success of testing is revealing errors in program depend critically on the test cases.

The two basic approaches are

- 1.Black box or functional testing
- 2.White box or structural testing

4.2.1 Black box or functional testing

Black box testing is also known as functional testing. A software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only known's the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The testers ever examinethe programming code and do not need any further knowledge of the program other than its specifications.

4.2.2 box or structural testing

White box testing includes analysing data flow, control flow, information flow, coding practices and exception and error handling within the system, to test the intended and unintended software behaviour. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities. White box testing requires access to source code.

Though white box testing can be performed any time in the life cycle after the code is developed; it is a good practice to perform White box testing during the unit testing phase.

4.3 Different Levels of testing

During testing process different levels of testing are used. Each levels of testing aiming to test different aspects of the system. The three different levels of testing are,

4.3.1 Unit testing

Unit testing focuses verification efforts on the smallest unit of software i.e, the module. Using detailed design and the process specification testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins. Here different modules are tested against the specification produced during design for the module. It is necessary to verify the code written during the coding phase.

4.3.2 Integration testing

After the unit testing, we have to perform the integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence emphasis on testing module interaction. During this testing many unit tested modules are combined into sub systems which are then tested.

The following are the types of Integration Testing:

- Top-Down Testing
- Bottom-Up Testing

Top-Down Testing

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinated to the main program module is incorporated into the structure in either a depth first or breadth first manner.

Bottom-Up Testing

This method begins the construction and testing the modules at the lowest level in the program structure. Since the modules are integrated from the bottom-up processing required for modules subordinated to a given level is always available.

The bottom-up integration strategies may be implemented with the following steps:

- The low-level modules are combined into clusters that perform a specific software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case.
- Input and output.
- The cluster is tested.
- Drivers are modulated and clusters are combined moving upward in the program structure.

4.3.3 System testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

4.3.4 Acceptance testing

User acceptance testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

4.4 Software Test Report

Test Case

Sl. No	Test Case	Expected Result	Obtained Result
1	Empty field in required Textbox	Enter some values in textbox	Successful

Table. 4.4.1 Empty Textbox Validation for Buyer

The screenshot displays the BizCart4all website's registration page. At the top, there is a navigation bar with contact information (bizcart4all@gmail.com, +91 7019115360) and social media links. Below this is a header with the BizCart4all logo, navigation links (HOME, VENDORS, SHOP, ABOUT US, CONTACT US), and a search bar. The main content area features a registration form titled 'WELCOME TO BIZCART4ALL' and 'Register'. The form includes fields for Name, Email, and Password. The Email field is highlighted with a red border and a message box saying 'Please fill in this field.' Below the form is a green 'Register' button. To the right of the form is an illustration of a person signing up. At the bottom of the form, there is a link to 'Login' for users who already have an account.

Fig.4.4.1 Empty Text Box Validation for Buyer

Sl. No	Test Case	Expected Result	Obtained Result
1	Empty field in required Textbox	Enter some values in textbox	Successful

Table. 4.4.2 Empty Textbox Validation for Seller

The screenshot shows the 'Profile' page for a seller named 'Seller10'. The page has a sidebar with navigation links: Dashboard, Add Product, All Product, Orders, Payments, Chat-Customer, Chat-Support, Profile (selected), and Logout. The main content area contains a form with the following fields and validation messages:

- Shop Name:** Shop Name is required
- Phone Number:** Phone Number is required
- GST Number:** Invalid GST Number
- Shop Address:** Address is required
- State:** State Name is required
- District:** District is required

At the bottom of the form is a 'Submit' button. The top right corner shows the user is logged in as 'Seller10'.

Fig.4.4.2 Empty Text Box Validation for Seller

Sl. No	Test Case	Expected Result	Obtained Result
2	Email validation	Valid Email	Successful

Table. 4.4.3 Email Validation

The screenshot shows the BizCart4all website header with navigation links: HOME, VENDORS, SHOP, ABOUT US, and CONTACT US. Below the header is a search bar with the text 'What do you need' and a 'SEARCH' button. The main content area features a large illustration of a person sitting at a desk with a laptop, surrounded by gears and a lightbulb. To the right of the illustration is a 'Login' form with the following fields and buttons:

- Email:** pratham@gmail.com
- Password:** (masked with asterisks)
- LOGIN** button
- [Forgot Password?](#) link
- [Don't Have An Account? Register](#) link
- LOGIN AS A SELLER** button
- REGISTER AS A SELLER** button

The top of the page shows the email 'bizcart4all@gmail.com' and a phone number '+ (91) 7019115360'. A notification banner at the top says 'Email Not Found'.

Fig 4.4.3 Email Validation

Sl. No	Test Case	Expected Result	Obtained Result
3	Password	Password didn't match	Successful

Table. 4.4.4 Password Validation

The screenshot displays the BizCart4all website interface. At the top, a blue header bar contains contact information (email: bizcart4all@gmail.com, phone: +91 7019115360) and a 'Login' button. Below the header, a navigation menu includes links for HOME, VENDORS, SHOP, ABOUT US, and CONTACT US. A green sidebar on the left features an 'All Category' dropdown. The main content area is divided into two sections. On the left, there is an illustration of a person sitting at a desk with a laptop, interacting with a large screen that displays a login form. On the right, a 'Login' form is visible, featuring input fields for 'Email' (containing 'sowjanyaadevadiga300@gmail.com') and 'Password' (masked with asterisks). A green 'LOGIN' button is positioned below the password field. Below the login button, there are links for 'Forgot Password?' and 'Don't Have An Account? Register'. At the bottom of the login section, there are two buttons: 'LOGIN AS A SELLER' (green) and 'REGISTER AS A SELLER' (purple). A dark blue error message box with a red 'x' icon and the text 'Password Wrong' is displayed above the login form.

Fig 4.4.4 Password Validation

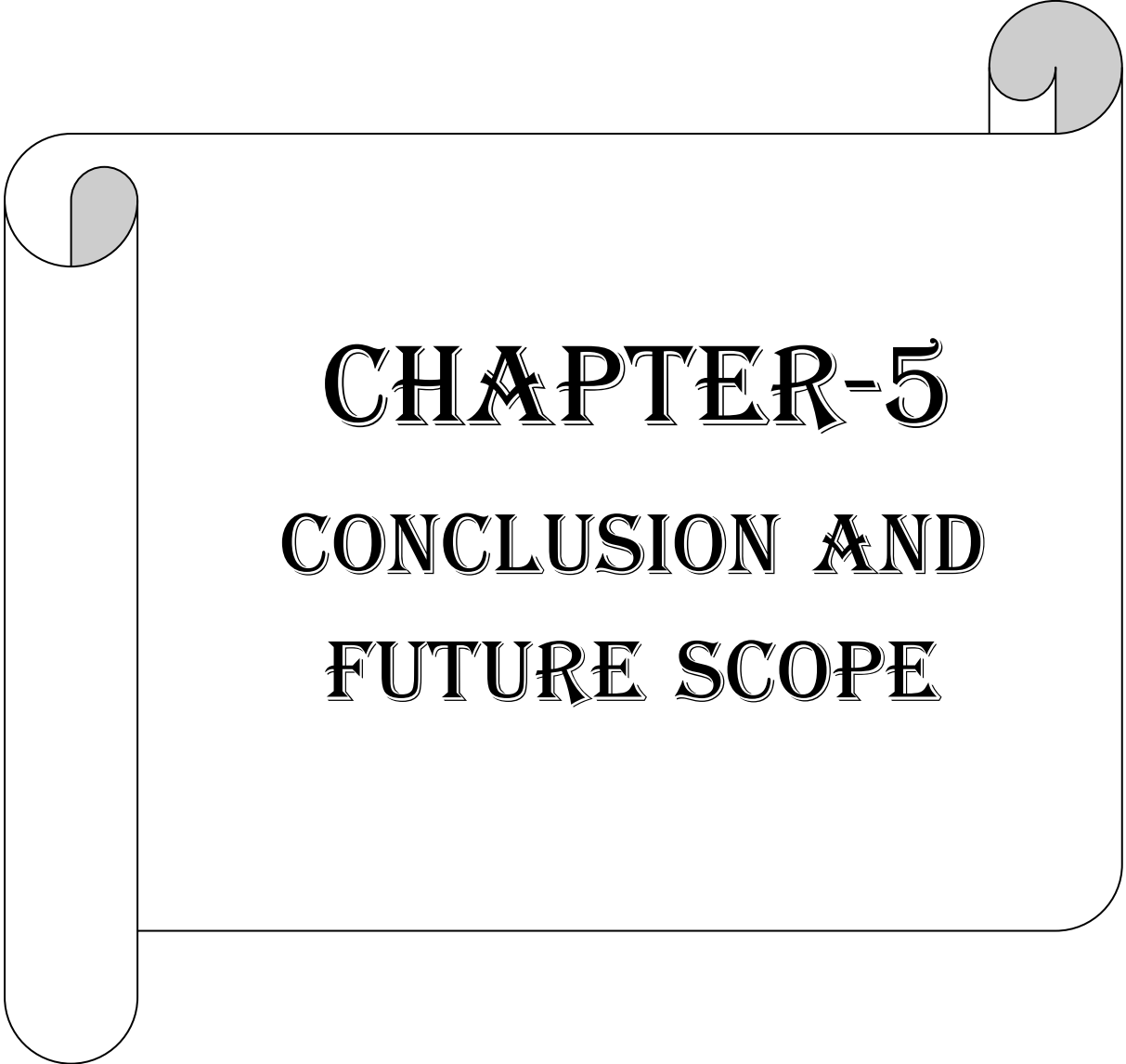
Sl. No	Test Case	Expected Result	Obtained Result
4	Password length less than 6 character	Must be minimum 6 character	Successful

Table. 4.4.5 Password Length Validation

The screenshot displays the registration interface of BizCart4all. At the top, a blue header bar contains contact information: an email icon followed by 'bizcart4all@gmail.com', a phone icon followed by '+ (91) 7019115360', and social media icons for Facebook, Twitter, and a 'Login' button. Below this is a navigation bar with the BizCart4all logo and menu items: HOME, VENDORS, SHOP, ABOUT US, and CONTACT US. A green 'All Category' dropdown and a search bar with a 'SEARCH' button are also present. On the right, there are icons for a heart and a shopping cart, along with a phone icon and the text '+91 897564132 Support 24/7'.

The main content area features a 'WELCOME TO BIZCART4ALL' message with a smiley face emoji. Below it is a 'Register' section with a red envelope icon. The registration form includes three input fields: 'Name' (containing 'Sneha'), 'Email' (containing 'sneha@gmail.com'), and 'Password' (containing six asterisks). A red error message is displayed below the password field: 'Password must contain at least one uppercase letter, one special character, one number, and be at least 6 characters long'. A green 'Register' button is positioned below the form. To the right of the form is an illustration of a person at a computer with a 'Sign Up' sign. At the bottom of the registration section, there is a horizontal line with 'Or' in the center, and a link that says 'Already have an account? Login'.

Fig 4.4.5 Password Length and Pattern Validation

A decorative border resembling a scroll, with a vertical strip on the left and a horizontal strip at the top, both featuring rounded ends and a grey shaded area indicating a rolled-up edge.

CHAPTER-5

CONCLUSION AND FUTURE SCOPE

6. CONCLUSION & FUTURE SCOPE

6.1 CONCLUSION

“BizCart4all”- Comprehensive Seller Services revolutionizes the multi-vendor e-commerce experience by addressing key challenges like inefficient database management and limited seller autonomy. It offers a user-friendly and secure platform for administrators, sellers, and buyers. With features such as advanced search and filtering options, robust security measures, and a responsive interface built with React.js, MongoDB, Redux, and Tailwind CSS, BizCart4all ensures a seamless and efficient shopping experience.

6.2 FUTURE ENHANCEMENT

To enhance the user experience on our e-commerce platform, we plan to integrate advanced image recognition technology to improve product listings, making it easier for sellers to upload accurate and visually appealing product images. Additionally, we aim to implement voice command functionality to provide users with a more seamless and hands-free interaction experience. To further personalize the shopping experience, we will introduce AI-driven recommendations that will suggest products based on user preferences and browsing history, ensuring a more tailored and engaging shopping journey.



REFERENCES

REFERENCES

- [1]. [Fashion Trade Show & Expo in Hong Kong | Global Sources \(2024\)](#)
- [2]. [Multi-vendor eCommerce Marketplace Development Guide - CSSChopper](#)
- [3]. [Top 15 Multi-Vendor Marketplace Platforms for Ecommerce 2024 \(cloudways.com\)](#)
- [4]. <https://tailwindcss.com>
- [5]. <https://legacy.reactjs.org/docs/getting-started.html>
- [6]. <https://www.w3schools.com/REACT/DEFAULT.ASP>
- [7]. <https://www.mongodb.com/docs/atlas/>
- [8]. Anti-Hacker Tool Kit 4th Edition by Mike Schema
- [9]. Software Engineering – A Practitioners approach 7th edition by Roger S. Pressman.