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 testdifferent 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. Usingdetailed 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. Modulesare 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 programstructure. Since the modules are integrated from the bottom-up processing required for modulessubordinated 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 testsa configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

4.3.4 Acceptance testing

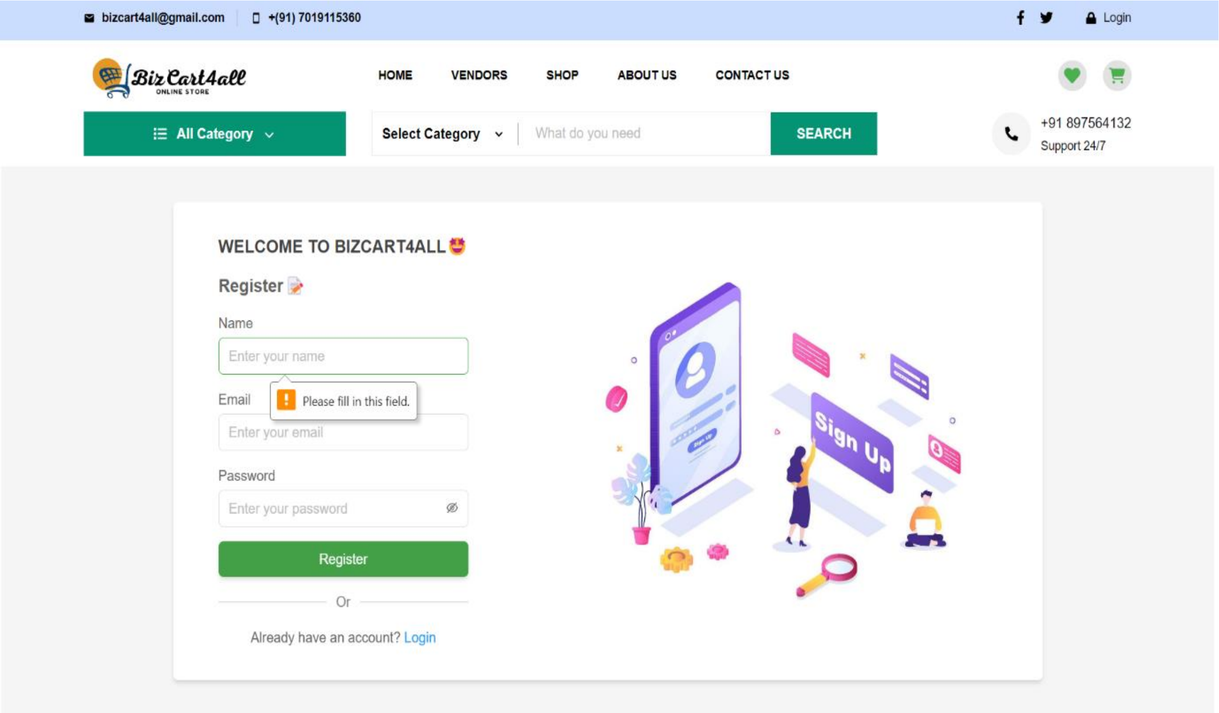
User acceptance testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

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



WELCOME TO BIZCART4ALL

Register

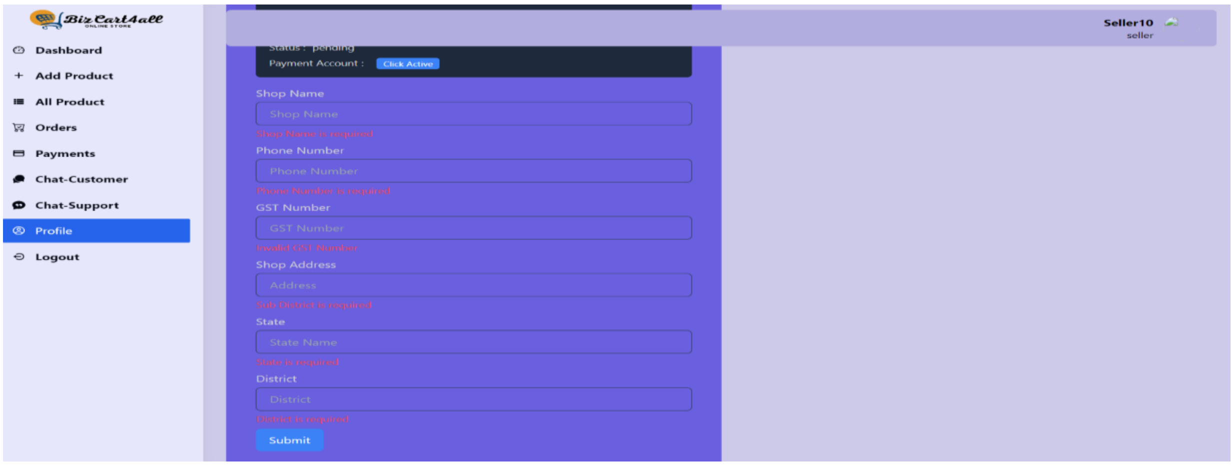
Please fill in this field.

Already have an account? Login

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



Status : pending
Payment Account : Click Active

Shop Name: Shop Name
Shop Name is required

Phone Number: Phone Number
Phone Number is required

GST Number: GST Number
Invalid GST Number

Shop Address: Address
Sub District is required

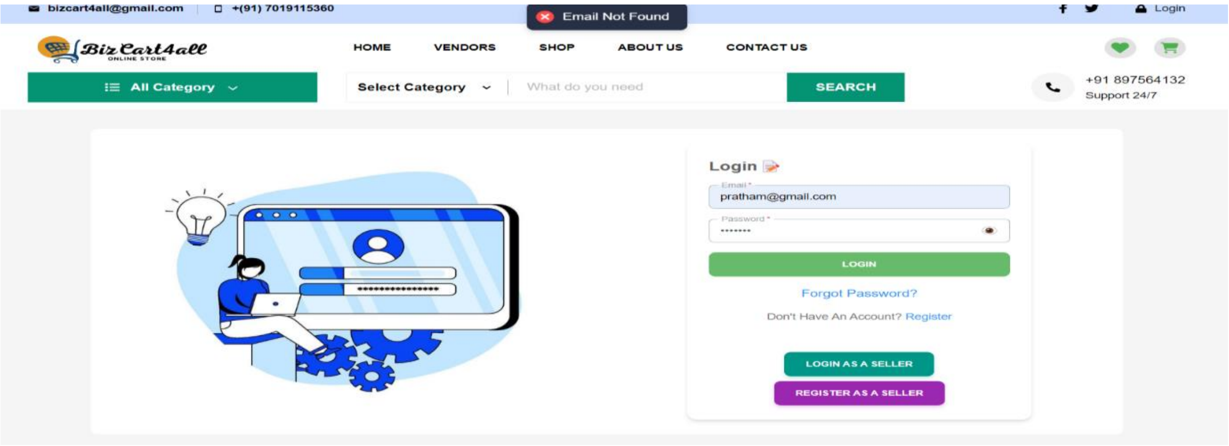
State: State Name
State is required

District: District
District is required

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



Email Not Found

Login

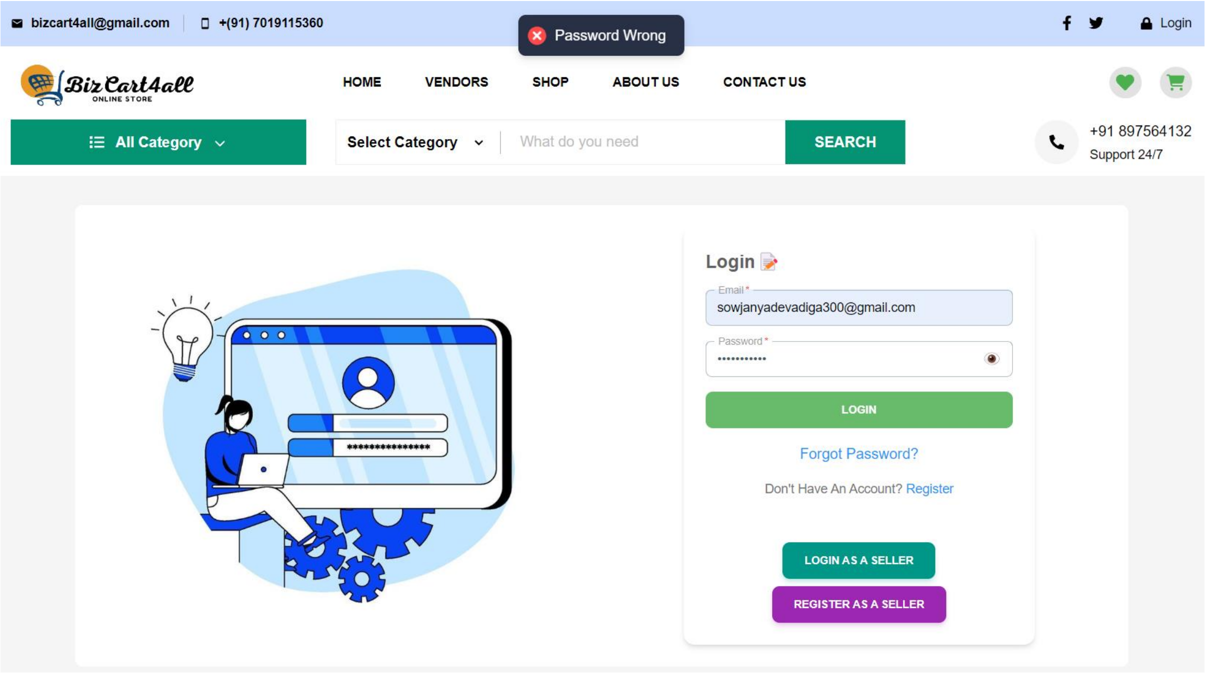
Email *: pratham@gmail.com

Password *: ••••••••

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



Password Wrong

Login

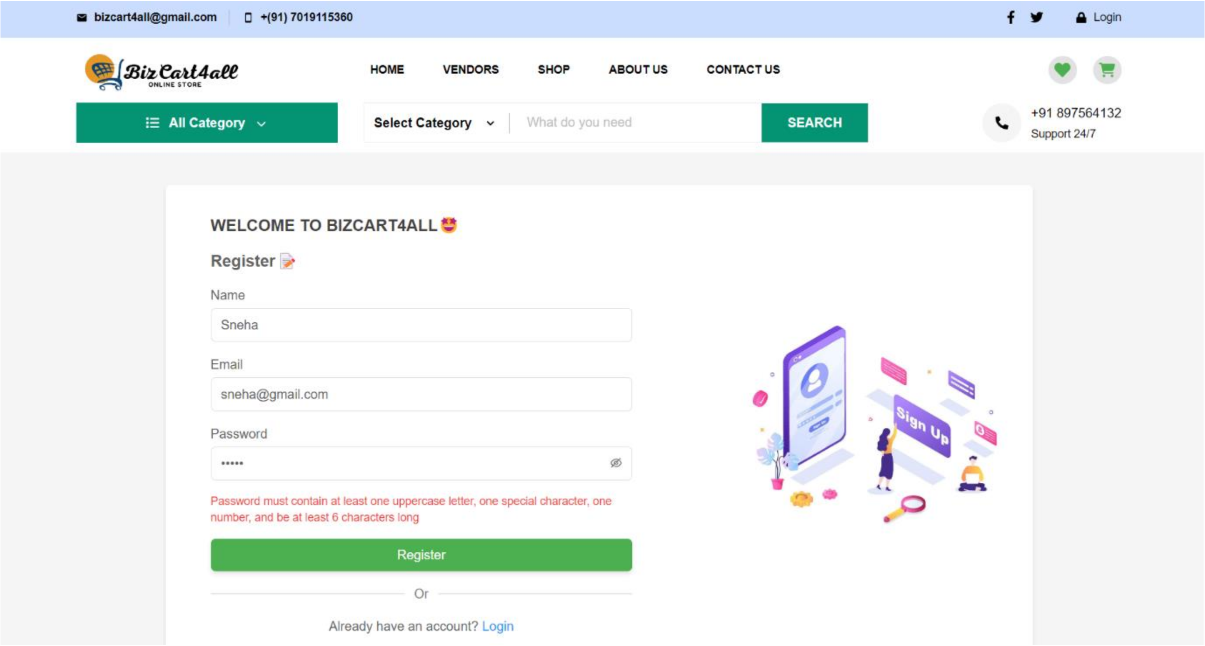
Email*: sowjanyadevadiga300@gmail.com

Password*: ••••••••••••

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



WELCOME TO BIZCART4ALL

Register

Name: Sneha

Email: sneha@gmail.com

Password: •••••

Password must contain at least one uppercase letter, one special character, one number, and be at least 6 characters long

Fig 4.4.5 Password Length and Pattern Validation

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.