

---

# Test Document

for

# KnowYourCibo

Version <1.0>

Prepared by

## Group #: 6

K S U Rithwin  
R V S Havish  
Mahanthi Vijay Kumar  
G N L Mahathi  
Rohan Aditya  
Rushikesh Chary  
Raj Vinayak Meena  
Laveti Bhanu Prakash  
Sai Dishanth  
Ravula Harshith Sai  
Gowtham Chand

## Group Name: Code Crafters

220537  
220879  
220602  
220395  
220741  
220336  
220861  
220583  
220282  
220878  
220313

[ksupendra22@iitk.ac.in](mailto:ksupendra22@iitk.ac.in)  
[rviswas22@iitk.ac.in](mailto:rviswas22@iitk.ac.in)  
[mvijay22@iitk.ac.in](mailto:mvijay22@iitk.ac.in)  
[gnagal22@iitk.ac.in](mailto:gnagal22@iitk.ac.in)  
[praditya22@iitk.ac.in](mailto:praditya22@iitk.ac.in)  
[drushikesh22@iitk.ac.in](mailto:drushikesh22@iitk.ac.in)  
[rajvm22@iitk.ac.in](mailto:rajvm22@iitk.ac.in)  
[lbhanu22@iitk.ac.in](mailto:lbhanu22@iitk.ac.in)  
[banoths22@iitk.ac.in](mailto:banoths22@iitk.ac.in)  
[ravula22@iitk.ac.in](mailto:ravula22@iitk.ac.in)  
[cgchand22@iitk.ac.in](mailto:cgchand22@iitk.ac.in)

Course: CS253

Mentor TA: *Sarthak*

Date: 1 April, 2024

## Contents

<b>CONTENTS.....</b>	<b>II</b>
<b>REVISIONS.....</b>	<b>II</b>
<b>1 INTRODUCTION.....</b>	<b>3</b>
<b>2 UNIT TESTING.....</b>	<b>4</b>
<b>3 INTEGRATION TESTING.....</b>	<b>12</b>
<b>4 SYSTEM TESTING.....</b>	<b>17</b>
<b>5 CONCLUSION.....</b>	<b>23</b>
<b>APPENDIX A - GROUP LOG.....</b>	<b>25</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
v1.0	Group-code crafters	Initiated the document. Added all the necessary details.	31/03/24

# 1 Introduction

**Test strategy:**

System testing and integration testing is conducted manually to evaluate the complete and integrated software product's functionality, usability, and compliance with requirements. Automated testing focuses on unit tests for individual components. Therefore both manual testing and automated testing are used.

**When was the testing conducted?**

During the development phase, we conducted preliminary tests focused on the software's core functionalities to verify its operational capability. Most of the comprehensive testing, however, took place post-implementation. This approach facilitated the efficient handling of edge cases through the incorporation of specific conditional statements, which allowed for minimal adjustments without significantly impacting the overall software architecture, enabling localized modifications.

**Who were the testers?**

The developers only took the role of testers. However, we adhered to a common industry practice by making sure that the creators of specific features did not test their own work. This method allowed multiple viewpoints to scrutinize each aspect of the software, revealing certain misconceptions held by some of the developers. When these mistakes were pointed out, the developers often anticipated additional issues that might arise when the software encountered unusual or extreme inputs.

**What coverage criteria were used?**

For manual system testing, coverage included all user stories, functional requirements, and use case scenarios. Automated testing coverage criteria focused on function and statement coverage for unit tests, and interface and interaction coverage for integration tests.

**Have you used any tool for testing?**

Jest was utilized for automated unit and integration testing, providing a framework for testing JavaScript code, including asynchronous operations and mocking dependencies. No specific tools were mentioned for manual system testing; it was conducted based on predefined test cases and scenarios.

## 2 Unit Testing

### 1. POST / Login

Backend function for login was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / Login function.

**Test Owner:** Havish

**Test Date:** [20/03/2024] - [20/03/2024]

**Test Results:**

Correct Email and Password – “Login Successful”

Correct Username and wrong Password  
(or)

Wrong Email and correct Password – “Incorrect Login details”

**Structural Coverage:** 85.71%

**Branch coverage :** 100%

**Additional Comments:** The tests thoroughly assess the login process, ensuring correct handling of successful logins, incorrect credentials, and error conditions.

**GitHub Test Link:** [login-test](#)

### 2. POST / Review

Backend function for Review was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / Review function.

**Test Owner:** [vijay]

**Test Date:** [20/03/2024] - [20/03/2024]

**Test Results:**

User login - Should add the review comment and increment user review count  
(or)

Should Update an existing review if the user has already reviewed the item

If User not logged in - Displays error message (“You must be logged in to add a review”)

**Structural Coverage:** 92%

**Branch coverage :** 100%

**GitHub Test Link:** [review test link](#)

### 3. POST / Submit Rating

Backend function for submit rating was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / Submit Rating function.

**Test Owner:** [vijay]

**Test Date:** [23/03/2024] - [23/03/2024]

**Test Results:**

User login - Should successfully submit a rating and increment user number of ratings

If User not logged in - Displays error message ("User must be logged in to rate items")

**Structural Coverage:** 96.15%

**Branch coverage :** 87.5%

**GitHub Test Link:** [submit rating test link](#)

### 4. POST / Remove Rating

Backend function for Remove Rating was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / Remove Rating function.

**Test Owner:** [vijay]

**Test Date:** [23/03/2024] - [23/03/2024]

**Test Results:**

User login - Should successfully remove a rating and decrement user number of ratings

If User not logged in - Displays error message ("User must be logged in to remove a review")

**Structural Coverage:** 90.47%

**Branch coverage :** 100%

**GitHub Test Link:** [remove rating test link](#)

### 5. POST /Send-Otp

Backend function for send-otp was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST /send-otp function.

**Test Owner:** Havish

**Test Date:** [20/03/2024] - [20/03/2024]

**Test Results:**

- Sending OTP with correct details – “OTP sent to email successfully”
- Handling failure in sending OTP – “Error sending OTP”

**Structural Coverage:** 85.71%

**Branch Coverage:** 100%

**Additional Comments:** The tests ensure that the OTP is generated and emailed to the user correctly, and appropriate error handling is in place for mail delivery failures.

**GitHub Test Link:** [send-otp test](#)

## 6. POST /set-password

Backend function for set-password was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST /set-password function.

**Test Owner:** Mahathi

**Test Date:** [20/03/2024] - [20/03/2024]

**Test Results:**

- Setting password with verified OTP – “Password set successfully”
- Attempting to set password with mismatching confirm password – “Passwords do not match”
- Attempting to set password without OTP verification – “OTP verification required”

**Structural Coverage:** 90%

**Branch Coverage:** 100%

**Additional Comments:** The tests covered scenarios for setting the password including OTP verification, password mismatch handling, and absence of OTP verification.

**GitHub Test Link:** [set-up password test](#)

## 7. POST /signup

Backend function for signup was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST /signup function.

**Test Owner:** Mahathi

**Test Date:** [20/03/2024] - [20/03/2024]

**Test Results:**

- Successful user signup with valid details – “Signup complete. You can now log in.”
- Attempt to signup with missing details – “Please fill in all the fields.”

**Structural Coverage:** 100%

**Branch Coverage:** 100%

**Additional Comments:** The tests cover successful user signups and the handling of incomplete signup attempts.

**GitHub Test Link:** [signup test](#)

## 8. POST /verify-otp

Backend function for verify-otp was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST /verify-otp function.

**Test Owner:** Havish

**Test Date:** [20/03/2024] - [20/03/2024]

**Test Results:**

- Correct OTP verification – “OTP verified successfully. Please set your password.”
- Incorrect OTP or expired OTP – “Invalid or expired OTP”

**Structural Coverage:** 100%

**Branch Coverage:** 100%

**Additional Comments:** The tests effectively validate the OTP verification process, handling both successful verifications and failed attempts due to incorrect or expired OTPs.

**GitHub Test Link:** [verify-otp test](#)

## 9. POST /submit-feedback

Backend function for submit-feedback was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST /submit-feedback function.

**Test Owner:** Havish

**Test Date:** [20/03/2024] - [20/03/2024]

**Test Results:**

- Feedback submitted with valid details – “Feedback submitted successfully and emailed.”
- Handling failures in submitting feedback – “Error submitting feedback.”

**Structural Coverage:** 80%

**Branch Coverage:** 100%

**Additional Comments:** The tests ensure that feedback is captured correctly and emailed as expected. Tests also cover error handling for mail sending issues.

**GitHub Test Link:** [feedback test](#)

## 10. POST / like-item

Backend function for liking or adding to favorites was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST /like-item function.

**Test Owner:** Rushikesh Chary

**Test Date:** [21/03/2024] - [21/03/2024]

**Test Results:**

- Added to favorites successfully – “Item added to your favorites list”
- Handling failures – “Error in adding to your favorites”

**Structural Coverage:** 91.89%

**Branch Coverage:** 83.33%

**Additional Comments:**

User login - Should successfully like an item and get added to his favorites list.

If User not logged in - Displays error message (“You are not logged in”)

**GitHub Test Link:** [like-test](#)

## 11. POST / dislikeItem



Backend function for dislike or removal from favorites was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / dislikeItem function.

**Test Owner:** Rushikesh Chary

**Test Date:** [21/03/2024] - [21/03/2024]

**Test Results:**

User login - Should successfully dislike or remove an item from his favorites from his profile page and from restaurant page where item is present.

If User not logged in - No option for disliking will be available.

**Structural Coverage:** 91.89%

**Branch coverage :** 83.33%

**GitHub Test Link:** [dislikeItem-test](#)

## 12. POST / editProfile

Backend function for editing profile was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / editProfile function.

**Test Owner:** Rushikesh Chary

**Test Date:** [21/03/2024] - [21/03/2024]

**Test Results:**

User login

- (entered correct password)Should successfully edit their profile (change their first name, last name and password)
- (entered incorrect password)Cannot edit his profile.

User not logged in - No option for editing profile.

**Structural Coverage:** 91.89%

**Branch coverage :** 83.33%

**GitHub Test Link:** [editProfile-test](#)

## 13. POST / searchResult

Backend function for searching for an item by its name/category/restaurant was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / searchResult function.

**Test Owner:** Rushikesh Chary

**Test Date:** [21/03/2024] - [21/03/2024]

**Test Results:**

- Item found with given input in search - "Displays all those items with restaurant and rating"
- item not found with given input in search - "Displays no items found"

**Structural Coverage:** 94.44%

**Branch coverage :**100%

**GitHub Test Link:** [searchPage-test](#)

#### 14. GET/ profileRender

Backend function for rendering the profile page of a particular user was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle GET/ profileRender function.

**Test Owner:** Rushikesh Chary

**Test Date:** [21/03/2024] - [21/03/2024]

**Test Results:**

User login - Should successfully render the user's profile page with all his details.

If User is not logged in - Cannot access this page .No option for profile access will be available.

**Structural Coverage:** 91.89%

**Branch coverage :** 83.33%

**GitHub Test Link:** [profileRender-test](#)

#### 15. GET / editProfileGet

Backend function for rendering the editProfile page was tested and all the testable branches have functioned as expected.

**Unit Details:** Backend function to handle POST / editProfileGet function.

**Test Owner:** Rushikesh Chary

**Test Date:** [21/03/2024] - [21/03/2024]

**Test Results:**

User login - Should successfully access this page with some of his details filled in already.

If User not logged in - Cannot access this page.

**Structural Coverage:** 91.89%

**Branch coverage :** 83.33%

**GitHub Test Link:** [editProfileGet-test](#)

## 3 Integration Testing

### 1. Checking the interface between login and corresponding previous page

**Module Details:** Login

**Test Owner:** Rushikesh Chary

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** When entering the correct user login credentials, the software redirects to the corresponding page from which login was done.

**Additional Comments:** N/A

### 2. Logout

**Module Details:** Logout

**Test Owner:** Rushikesh Chary

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** Logout functionality works as expected, and upon logout from any page, the user is successfully redirected to the homepage.

**Additional Comments:** N/A

### 3. Signup Redirection and Login Success

**Module Details:** Signup and Login Redirection

**Test Owner:** Mahathi

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:**

- Signup Redirection Test: After successfully signing up a new user with the credentials (username and password), the application redirected to the login page as expected.
- Login Success Test: Entered the correct details for the newly signed up user on the login page, and the login was successful. The user was authenticated and granted access to the system.

**Additional Comments:**

The signup process not only created a new user account but also correctly redirected to the login page, thereby linking the signup and login processes smoothly. The login functionality accepted the new credentials without issue, confirming that the new account was set up properly and the authentication flow is functioning as designed. This test ensures the cohesiveness of the signup to login transition for new users.

### 4.Session Management:

**Module Details:** Login

**Test Owner:** Havish

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** After logging in, attempting to access restricted pages without logging out redirects the user to the login page, indicating proper session management.

**Additional Comments:** Session handling is effective in maintaining user authentication throughout the session.

## 5.OTP Sending and Verification Process

**Module Details:** OTP Workflow (Send and Verify)

**Test Owner:** Mahathi

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:**

- Send OTP Test: Requested an OTP for the email "test@example.com". An OTP was successfully sent to the email address provided.
- Verify OTP Test: Using the OTP received, the verification process was initiated. The OTP was correctly accepted by the system and verified successfully.

**Additional Comments:**

The integration between the OTP sending and verification functions is seamless. The test confirms that the OTP generated and sent to the user's email can be used effectively to verify the user in the subsequent step. This is crucial for processes such as password reset or two-factor authentication where OTP verification is a mandatory step.

## 6.Integration with Ratings and Reviews:

**Module Details:** Login

**Test Owner:** Havish

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** After logging in with valid credentials, the user is able to see the ratings and reviews given by him.

**Additional Comments:** Integration with the ratings and reviews module is functional, allowing seamless access post-login.

## 7.OTP Verification and Setting New Password

**Module Details:** OTP Verification and New Password Setting

**Test Owner:** Mahathi

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:**

- OTP Verification Test: The received OTP was submitted for verification using the "verify-otp" endpoint. The system acknowledged that the OTP was correct.
- Set New Password Test: Upon OTP verification, a new password was set using the "set-password" endpoint. The system updated the user's password and provided a confirmation that the new password was successfully established.

**Additional Comments:**

This test affirms the effective linkage between OTP verification and the user's ability to set a new password. The process is crucial for new user account creation where setting up a password is a mandatory step following OTP verification. We can ensure that new users will be able to securely set their password, contributing to the robustness of account creation security measures.

## 8. Navigation bar Integration:

**Module Details:** Navigation Bar

**Test Owner:** Mahathi

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** All navigation links are functional and correctly direct users to the intended pages.

**Additional Comments:** Integration with the navigation bar is seamless, providing easy access to key sections of the website.

## 9. Search Functionality Integration:

**Module Details:** Search

**Test Owner:** Havish

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** Upon entering search queries for restaurant names and food items, relevant results are displayed accurately

**Additional Comments:** Integration with the search functionality enhances user experience by enabling quick access to desired content.

## 10. Ratings and Reviews Integration:

**Module Details:** Ratings and Reviews

**Test Owner:** Havish

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** Attempting to submit ratings and reviews without logging in displays a message box indicating the need to log in. After logging in, users can successfully submit ratings and reviews.

**Additional Comments:** Integration with ratings and reviews functionality ensures that user feedback is accurately captured and contributes to enhancing the user experience.

## 11.Profile Page Integration with Favorites and Reviews:

**Module Details:** Profile Page, Favorites, Ratings and Reviews

**Test Owner:** Rushikesh Chary

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** The profile page correctly displays the user's favorites list and accurately counts the total number of items they have rated and reviewed.

**Additional Comments:** Integration with the profile page provides users with easy access to their saved favorites and a summary of their engagement with the website's content.

## 12.Password Recovery Integration:

**Module Details:** forgot password

**Test Owner:** Vijay

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** Users can successfully recover their password by following the password recovery process and setting a new password.

**Additional Comments:** Integration with the password recovery feature enhances user convenience and security by providing a mechanism for recovering lost passwords.

## 13.Restaurant Page Integration with Favorites:

**Module Details:** Restaurant Page, Favorites

**Test Owner:** Vijay

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** Users can successfully add food items to their favorites list by clicking the "Add to Favorites" button on the restaurant page.

**Additional Comments:** Integration with the favorites functionality enhances user engagement by allowing them to save their preferred food items for future reference.

## 14.Search Integration with Restaurant Pages:

**Module Details:** Search, Restaurant Pages

**Test Owner:** Vijay

**Test Date:** [22/03/2024] - [22/03/2024]

**Test Results:** Clicking on search results for restaurant names accurately redirects users to the respective restaurant pages.

**Additional Comments:** Integration with search functionality enhances user navigation by providing direct access to relevant restaurant pages.

## 4 System Testing

### 1. Performance Testing:

**Requirement:** The system should be able to handle a certain number of concurrent users without experiencing performance degradation.

**Test Owner:** Rushikesh Chary

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:** Conducted performance testing to assess system response times and resource utilization under normal and peak load conditions using an external software **apache benchmark http server**. Monitored server response times, database query performance, and system stability under load. Analyzed performance metrics to ensure that the system meets performance requirements.

**Additional Comments:** The system performed well under normal and peak load conditions, meeting performance requirements and providing a responsive user experience.

### 2. Data Integrity Testing:

**Requirement:** The system should ensure the integrity of user data by accurately storing and retrieving information from the database.

**Test Owner:** Havish

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:** Conducted data integrity testing manually to verify that data entered by users is correctly stored in the database and retrieved accurately when requested. Tested various data types, including text, numbers, dates, and images. Checked for data consistency and accuracy across different modules and transactions.

**Additional Comments:** Data integrity testing confirms that the system maintains the integrity and consistency of user data, providing reliable information for users and administrators.

### 3. Error Handling Testing:

**Requirement:** The system should effectively handle errors and exceptions to provide users with informative error messages and recover gracefully from unexpected situations.

**Test Owner:** Vijay

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:** Conducted error handling testing to identify and validate the system's response to various error conditions, such as invalid inputs, network errors, and system failures. Tested error scenarios across different functionalities and user interactions.



Verified that appropriate error messages are displayed, and the system gracefully recovers from errors without data loss or system instability.

**Additional Comments:** Error handling testing ensures that the system provides a robust and reliable user experience by guiding users through error situations and maintaining system integrity.

#### 4. Concurrency Testing:

**Requirement:** The system should support multiple users accessing and interacting with the system simultaneously without data corruption or performance degradation.

**Test Owner:** Rushikesh Chary

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:** Conducted concurrency testing using **Apache Benchmark http server** to evaluate the system's behavior under concurrent user loads. Simulated multiple users accessing the system concurrently and performing typical user interactions, such as logging in, browsing, and submitting data. Monitored system performance, resource utilization, and data consistency under concurrent load conditions.

**Additional Comments:** Concurrency testing confirms that the system can handle multiple users simultaneously without experiencing data corruption or significant performance degradation, ensuring scalability and reliability.

```
Server Software:      cloudflare
Server Hostname:      knowyourcibo.onrender.com
Server Port:          80

Document Path:        /
Document Length:      69 bytes

Concurrency Level:    100
Time taken for tests:  4.094 seconds
Complete requests:    100
Failed requests:      0
Non-2xx responses:    100
Total transferred:    37100 bytes
HTML transferred:     6900 bytes
Requests per second:  24.43 [#/sec] (mean)
Time per request:     4093.704 [ms] (mean)
Time per request:     40.937 [ms] (mean, across all concurrent requests)
Transfer rate:        8.85 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    21    37  13.4    32    89
Processing: 132  1857 1066.9  1881  3869
Waiting:    132  1818 1060.3  1836  3630
Total:      180  1894 1066.3  1930  3905
```

## 5.Session Management Testing:

**Requirement:** The system should manage user sessions securely and efficiently to maintain user authentication and session state.

**Test Owner:** Mahathi

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:** Conducted session management testing to verify that user sessions are created, maintained, and terminated correctly. Tested session expiration, session persistence across different pages, and session hijacking prevention measures. Verified that users are automatically logged out after a period of inactivity and that session tokens are securely transmitted and validated.

**Additional Comments:** Session management testing ensures that the system maintains user authentication and session state securely, protecting against unauthorized access and session hijacking.

## 6.Email Notification Testing:

**Requirement:** The system should send email notifications to users for important events, such as account registration, password reset, and order confirmation.

**Test Owner:** Mahathi

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:** Conducted email notification testing to validate that email notifications are sent correctly for various system events. Verified the content, formatting, and delivery of email notifications. Tested email delivery to different email providers and verified that emails are not marked as spam. Checked for error handling in case of email delivery failures.

**Additional Comments:** Email notification testing ensures that users receive timely and accurate email notifications for important system events, enhancing communication and user engagement.

## 7.Requirement: User Registration and Login

**Test Owner:** Havish

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:**

- Verified that users can register with valid information and receive a confirmation email.
- Tested user login with valid credentials and verified successful authentication.

- Checked for error handling in case of invalid inputs or login attempts.
- Ensured that users are redirected to the correct pages after registration and login.

**Additional Comments:** The user registration and login functionality meet the requirements outlined in the SRS document, providing a secure and seamless authentication process.

## 10.Requirement: Rating and Review Submission

**Test Owner:** Vijay

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:**

- Tested rating and review submission for different food items.
- Verified that users can provide ratings and optional reviews with appropriate validation.
- Checked for error handling in case of incomplete or invalid submissions.
- Ensured that submitted ratings and reviews are displayed correctly on the restaurant page.

**Additional Comments:** The rating and review submission functionality meet the requirements specified in the SRS document, allowing users to share feedback effectively.

## 11.Requirement: User Profile Management

**Test Owner:** Rushikesh Chary

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:**

- Tested user profile editing for various fields such as name, email, password, and preferences.
- Verified that users can update their profile information with valid inputs and save changes successfully.
- Checked for error handling in case of invalid or incomplete profile updates.
- Ensured that updated profile information reflects accurately across the system.

**Additional Comments:** The user profile management functionality meets the requirements outlined in the SRS document, allowing users to customize their profiles effectively.

## 12.Requirement: Favorites Management

**Test Owner:** Rushikesh Chary

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:**

- Tested adding and removing items from the favorites list for different categories such as restaurants or food items.
- Verified that users can add items to favorites with a single click and remove them easily.
- Checked for error handling in case of adding duplicate items or removing non-existent items.
- Ensured that the favorites list is updated dynamically and reflects the latest changes made by the user.

**Additional Comments:** The favorites management functionality aligns with the requirements specified in the SRS document, allowing users to save and organize their preferred items conveniently.

### **13.Requirement: Feedback Submission**

**Test Owner:** Vijay

**Test Date:** [25/03/2024] - [25/03/2024]

**Test Results:**

- Tested feedback submission through the designated form or interface provided in the system.
- Verified that users can provide feedback on different aspects such as user experience, features, or suggestions for improvement.
- Checked for validation of feedback inputs and appropriate error handling.
- Ensured that submitted feedback is recorded and accessible to administrators for review and action.

**Additional Comments:** The feedback submission functionality meets the requirements outlined in the **SRS document**, enabling users to contribute valuable insights for system enhancement.

## 5 Conclusion

### How Effective and exhaustive was the testing?

#### All files Backend/controllers

91.76% Statements 223/243 94.54% Branches 52/55 95.23% Functions 20/21 91.98% Lines 218/237

Press n or j to go to the next uncovered block, b, p or k for the previous block.

Filter:

File	Statements	Branches
feedbackController.js	80%	100%
loginController.js	85.71%	100%
profilecontrollers.js	91.89%	83.33%
ratingController.js	96.15%	87.5%
removeratingController.js	90.47%	100%
reviewcontroller.js	92%	100%
searchcontroller.js	94.44%	100%
sendOtpController.js	85.71%	100%
setPasswordController.js	90%	100%
signupController.js	100%	100%
verifyOtpController.js	100%	100%

The testing process was highly effective and exhaustive, covering a wide range of functional and non-functional requirements. Through meticulous test planning, execution, and analysis, we ensured thorough coverage of all use cases and scenarios. As a result, the system's robustness and reliability were significantly enhanced, and the risk of critical defects in production was minimized.

```
Test Suites: 13 passed, 13 total
Tests:      43 passed, 43 total
Snapshots:  0 total
Time:       8.641 s
Ran all test suites.
```

### Which components have not been tested adequately?

While our testing efforts were comprehensive, certain components, such as those involving complex integrations with external systems or third-party APIs, may have received less extensive coverage. Additionally, non-functional aspects like performance, scalability, and security could benefit from further testing to ensure optimal system behavior under varying conditions.

### What difficulties have you faced during testing?

Despite our best efforts, we encountered challenges related to resource constraints and time limitations. Coordinating testing efforts across multiple teams and environments

proved to be demanding, especially in cases where dependencies were not readily available. Adapting to evolving requirements and managing changes during the testing phase also presented logistical hurdles.

**How could the testing process be improved?**

Moving forward, we can enhance the testing process by investing in automated testing tools and frameworks to expedite repetitive tasks and increase test coverage. Allocating dedicated resources for testing and establishing clear communication channels between development and testing teams will facilitate smoother collaboration and streamline issue resolution. Conducting thorough risk assessments early in the project lifecycle can help identify critical areas requiring intensive testing, ensuring that resources are allocated effectively. Additionally, conducting regular retrospectives to reflect on our testing strategies and identify areas for improvement will contribute to ongoing refinement of the testing process.

**Appendix A - Group Log**

<b>SL. no.</b>	<b>Date</b>	<b>Timings</b>	<b>Venue</b>	<b>Description</b>
1.	20/03/2024	3 pm - 6 pm	RM Building	Make the template for the testing document.
2.	21/03/2024	4 pm - 6 pm	Meet (Google Meet)	Distributed the work amongst the team members
3.	25/03/2024	4:30 pm - 6 pm	Meet (Google Meet)	Completed the first draft of the testing document and then discussed improving it further.
4.	28/03/2024	6:30 pm - 8 pm	Meet (Google Meet)	Reviewed the draft and also made the suggested changes during the meeting.
5.	31/03/2024	6:30 pm - 8 pm	RM Building	Completed the final draft and reviewed it and also made the suggested changes during the meeting.