**FinSmart app**

**\*Positive test case for login and registration page with otp.**

* 1. **Successful Login:**

Verify that a user can log in successfully using valid, registered credentials (username/email and password).

* 2. **"Remember Me" Functionality:**

Verify that if "Remember Me" is selected, the system remembers the user's credentials for subsequent visits.

* 3. **Forgot Password/Account Recovery:**

Verify that the "Forgot Password" link successfully initiates the password reset process, and a new password can be set and used for login.

* 4. **Case Sensitivity:**

Verify that the login process correctly handles case sensitivity for the username/email and password as per the system's design.

* 5. **Login with Enter Key:**

Verify that pressing the "Enter" key after entering credentials initiates the login process.

* 6. **Browser Compatibility:**

Verify successful login across different popular browsers (Chrome, Firefox, Edge, Safari).

* 7. **Device Compatibility:**

Verify successful login across various devices (desktop, mobile, tablet).

* 8. **Successful Registration:**

Verify that a user can successfully register an account using valid and unique information in all required fields.

* 9. **Password Requirements:**

Verify that the system enforces and accepts passwords meeting specified criteria (e.g., minimum length, special characters, uppercase/lowercase).

* 10. **Email Confirmation:**

Verify that a confirmation email is sent to the registered email address and that clicking the confirmation link successfully activates the account.

* 11. **Terms and Conditions Acceptance:**

Verify that the user can accept the terms and conditions and that registration proceeds accordingly.

12. **utofill Functionality:**

Verify that browser autofill features work as expected on the registration form.

* 13. **Verify OTP Generation:**

Confirm the OTP is generated correctly and matches the specified complexity (e.g., a 6-digit numeric code).

* 14. **erify OTP Delivery:**

Ensure the OTP is sent to the correct registered phone number or email address.

* 15. **Multi-Channel Delivery:**

Test delivery across SMS, email, and push notifications to verify functionality on all required channels.

* 16. **Single OTP Per Request:**

Verify only one OTP is generated and sent for a single authentication request.

* 17.[**Resend OTP**](https://www.google.com/search?cs=0&sca_esv=c97b6394e1ab69a5&sxsrf=AE3TifMwMPaYURFaSp42zY6syh5hiKeEFQ%3A1759660883895&q=Resend+OTP&sa=X&ved=2ahUKEwjnmubO74yQAxU1xjgGHR1OGskQxccNegQITBAB&mstk=AUtExfDcQYHXRPuObivbUGCKOVpxU08vijmyLllM8ZUuSkqh1UQGiGBF_xqdOAvU-nJgzveaS91asfTYjbAc2WRhGJlSJC7QUHZ-u4qlqzsca29Z-J8N1PZKDZ8oiqGBNNnGwlblOeCF0GEcaYGP-yM-nliBRk5Cqk4w0LVgpu2YkIoJp-RVbrb0GW05ZknwKkXloBOgcyxHxjp5pJyLc5rHoUVhrQsl6ESVNY-ANY-vziZ1DEDCM2QyCuOFlFjCUk4-Lcewuc25RlE9OEYqgXGxM1B98fvi0qvenGZIgqFAa0mcVA&csui=3)**Functionality:**

Test that the "Resend OTP" button works correctly and a new OTP is generated and sent.

* 18. **Multiple Resends:**

 Check that only the latest OTP received after clicking "Resend" is valid, not the previous ones.

* 19. **Multiple Requests for OTP:**

Verify the system's behavior when a user requests OTPs multiple times in a short period.

\*Negative test case on registration and login page of user with otp

**1.invalid Credentials:**

Attempt to log in with an invalid username/email, an invalid password, or a combination of both, and verify that appropriate error messages are displayed.

* 2. **Empty Fields:**

Attempt to log in with empty username/email and/or password fields and verify error messages.

* 3. **Incorrect Password Multiple Attempts:**

Test the system's behavior after multiple consecutive failed login attempts (e.g., account lockout, CAPTCHA).

* 4. **Unregistered User:**

Attempt to log in with credentials belonging to an unregistered user and verify the error message.

* 5. **Session Expiration:**

Verify that the user is logged out automatically after a predefined period of inactivity.

* 6. **Login After Logout:**

Verify that a user cannot access restricted content by using the browser's back button after logging out.

* 7. **Existing User Registration:**

Attempt to register with an email address or username that is already registered, and verify that an appropriate error message is displayed.

8. **nvalid Email Format:**

Attempt to register with an invalid email address format (e.g., missing "@" or domain), and verify error messages.

9. **assword Mismatch:**

Attempt to register where the "password" and "confirm password" fields do not match, and verify error messages.

* 10, **Weak Password:**

Attempt to register with a password that does not meet the specified strength requirements, and verify error messages.

* 11. **Empty Required Fields:**

Attempt to register with empty required fields and verify that validation errors are displayed.

* 12. **Terms and Conditions Not Accepted:**

Attempt to register without accepting the terms and conditions, and verify that registration is prevented or an error is shown.

13. **invalid Character Input:**

Attempt to input invalid or restricted characters into various fields (e.g., numbers in name fields), and verify the system's response.

14. **invalid OTP Entry:**

User enters an incorrect or invalid OTP, and an appropriate error message is displayed, preventing completion of the transaction.

* 15. **Expired OTP:**

User attempts to use an OTP after its validity period has expired, and the authentication fails.

16.[**OTP Expiration**](https://www.google.com/search?cs=0&sca_esv=c97b6394e1ab69a5&sxsrf=AE3TifMwMPaYURFaSp42zY6syh5hiKeEFQ%3A1759660883895&q=OTP+Expiration&sa=X&ved=2ahUKEwjnmubO74yQAxU1xjgGHR1OGskQxccNegQILxAB&mstk=AUtExfDcQYHXRPuObivbUGCKOVpxU08vijmyLllM8ZUuSkqh1UQGiGBF_xqdOAvU-nJgzveaS91asfTYjbAc2WRhGJlSJC7QUHZ-u4qlqzsca29Z-J8N1PZKDZ8oiqGBNNnGwlblOeCF0GEcaYGP-yM-nliBRk5Cqk4w0LVgpu2YkIoJp-RVbrb0GW05ZknwKkXloBOgcyxHxjp5pJyLc5rHoUVhrQsl6ESVNY-ANY-vziZ1DEDCM2QyCuOFlFjCUk4-Lcewuc25RlE9OEYqgXGxM1B98fvi0qvenGZIgqFAa0mcVA&csui=3)**Time Limit:**

Verify the OTP expires after its set time limit and cannot be used further

* 17. **Single Use:**

Confirm that once an OTP is used, it cannot be reused for another transaction.

* 18. **Multiple Resends:**

Check that only the latest OTP received after clicking "Resend" is valid, not the previous ones.

19.[**Network Failures**](https://www.google.com/search?cs=0&sca_esv=c97b6394e1ab69a5&sxsrf=AE3TifMwMPaYURFaSp42zY6syh5hiKeEFQ%3A1759660883895&q=Network+Failures&sa=X&ved=2ahUKEwjnmubO74yQAxU1xjgGHR1OGskQxccNegQIZBAB&mstk=AUtExfDcQYHXRPuObivbUGCKOVpxU08vijmyLllM8ZUuSkqh1UQGiGBF_xqdOAvU-nJgzveaS91asfTYjbAc2WRhGJlSJC7QUHZ-u4qlqzsca29Z-J8N1PZKDZ8oiqGBNNnGwlblOeCF0GEcaYGP-yM-nliBRk5Cqk4w0LVgpu2YkIoJp-RVbrb0GW05ZknwKkXloBOgcyxHxjp5pJyLc5rHoUVhrQsl6ESVNY-ANY-vziZ1DEDCM2QyCuOFlFjCUk4-Lcewuc25RlE9OEYqgXGxM1B98fvi0qvenGZIgqFAa0mcVA&csui=3)**:**

Test the system's behavior when OTP delivery fails due to network connectivity issues.

\*Positive test case for bank linking account via third party API

* 1. Verify that users can link their bank account using valid credentials and permissions.
  2. Confirm that the API successfully fetches and displays the correct account information (e.g., account type, balance, transaction history).
  3. Verify that the linked account is reflected accurately in the application.
  4. Verify that linked account information (e.g., balance) updates accurately and in real-time.
  5. Test the ability to link or unlink accounts (e.g., joint accounts, multiple accounts from the same bank).
  6. Verify that account statements are generated correctly for linked accounts.
  7. Verify that the API adheres to relevant data protection standards (e.g., GDPR) and local banking laws.
  8. Ensure that consent mechanisms are in place and working correctly.
  9. Ensure APIs are well-documented for smoother integration and testing.
  10. Check for the proper implementation of biometric verification and two-factor authentication during the linking process.
  11. Verify that account statements correctly display transactions and consolidated balances from linked accounts.
  12. Confirm that account balances are accurately reflected after transactions are processed through linked accounts.
  13. Ensure that users receive accurate notifications and alerts regarding successful linking, linking failures, or suspicious activity related to linked accounts.
  14. Check that the linking process adheres to relevant regulatory requirements, such as KYC (Know Your Customer) and anti-money laundering regulations.
  15. Verify that the application supports simultaneous access to the same linked account from multiple devices without data corruption or synchronization issues.

\*Negetive test case on bank linking account via third party API

* 1. Attempt to link an account with incorrect or expired login credentials.
* 2. Verify how the API handles invalid or missing API keys or tokens.
* 3. Verify that the application gracefully handles network errors (e.g., connection timeouts, DNS resolution failures).
* 4. Attempt to link an already linked account.
* 5. Test the API's behavior when the bank's system is undergoing maintenance or is unavailable.
* 6. Verify how the system handles scenarios like the user's account being locked or suspended.
* 7.Ensure APIs are well-documented for smoother integration and testing.
* 8. Attempt to link an account number that does not exist in the bank's database.
* 9. **Incorrect Account Holder Name:** Provide an account number and an account holder name that does not match.
* 10. **Invalid IFSC Code:** Enter an incorrect or invalid [IFSC (Indian Financial System Code)](https://www.google.com/search?sca_esv=cfd77b4f2a719fd7&cs=0&sxsrf=AE3TifMnNevAJL560cleXJ0d-wIctKP98Q%3A1759662861454&q=IFSC+%28Indian+Financial+System+Code%29&sa=X&ved=2ahUKEwjw-M399oyQAxUM7zgGHczbOskQxccNegQIIRAB&mstk=AUtExfBnkgHZ7fxBQ7_PLl2Ej62S-6OT_czZpTzHHnsCehTZWrOrdL12BUkRS43Zr8HvLWPPYl8JFpUwHsaRfa79euVMd2C6iaBJVlRn6ZWsZ-V4xl37j23T4jHvrWel1Sy63qJ5FPBSI6GjT7F9axHU-P5zxtjmwsW3xqOd1UuJIJ3zu1g8-et3xU7JqLlfRdu35Tb8mpWlepW7yGPL3z98ygILwyqx8bwYj5JeH81Z0eR9sYucWZH9Q3WpzdJmPxm7AqJtvVSlUdbg_M4pLx53TW5i&csui=3) for the bank.
* **11.Invalid or Expired Credentials:** Attempt to link an account using expired debit cards, mobile numbers, or other credentials.
* 12. **Invalid Character Types:** Enter alphanumeric characters in a field expecting only numbers (e.g., account number) or vice versa.
* 13. **Excessive Length:** Input an account number or other field that exceeds the allowed character limit.
* 14. **Special Characters:** Include disallowed special characters in input fields where they are not permitted.
* 15. **Missing Mandatory Fields:** Attempt to link an account without providing all the required fields (e.g., account number, bank name, IFSC).
* 16.**Incorrect Mobile Number:** Provide an invalid mobile number for an SMS authentication or notification, and verify a message is received.
* 17. **No Response to Invalid Input:** Verify the system displays a meaningful error message when an invalid input is provided, rather than failing to respond or crashing.

\*Senario on expense tracking

Functional Test Scenarios

These scenarios ensure the core functionalities of the expense tracker work as expected.

* [**Add Expense**](https://www.google.com/search?sca_esv=c97b6394e1ab69a5&cs=0&sxsrf=AE3TifN7424HQMU6CjkEc8iYbQOzljYgDg%3A1759663680227&q=Add+Expense&sa=X&ved=2ahUKEwie6_yD-oyQAxXvzqACHcBJAFsQxccNegQIRRAC&mstk=AUtExfCvsIOK3EMeOJzlXirOBN4bsFcYWtA1jlBmvx9glI30dv6wFlJ6_b73GxRGHh-JO5tq3rdgZqRT6-lzGsWKN6tXWWfUE7Irb3P16Col8Kyq8myjz4M2MKpaotpu7OcJ-4kvPTmcwj5VoYmCCMUSAu2XYxH2uFW14cX3_Gmg1Hj7a8h2oMU8fabGEE9Iuzzs4lhWq4ue1TQYDzYqB320C0aXkGpD6I8tfmbIfq-7dCq1tshoSZdWgJMfwfb7pYKu5n92cUYwlDXrXke9uVchyBUH&csui=3)**:**
  + Add an expense with all required fields filled (amount, category, date, name).
  + Add an expense with minimal required fields.
  + Add an expense with a description.
  + Add an expense with an attached receipt image.
  + Add an expense with a future date.
* [**Edit Expense**](https://www.google.com/search?sca_esv=c97b6394e1ab69a5&cs=0&sxsrf=AE3TifN7424HQMU6CjkEc8iYbQOzljYgDg%3A1759663680227&q=Edit+Expense&sa=X&ved=2ahUKEwie6_yD-oyQAxXvzqACHcBJAFsQxccNegQISRAC&mstk=AUtExfCvsIOK3EMeOJzlXirOBN4bsFcYWtA1jlBmvx9glI30dv6wFlJ6_b73GxRGHh-JO5tq3rdgZqRT6-lzGsWKN6tXWWfUE7Irb3P16Col8Kyq8myjz4M2MKpaotpu7OcJ-4kvPTmcwj5VoYmCCMUSAu2XYxH2uFW14cX3_Gmg1Hj7a8h2oMU8fabGEE9Iuzzs4lhWq4ue1TQYDzYqB320C0aXkGpD6I8tfmbIfq-7dCq1tshoSZdWgJMfwfb7pYKu5n92cUYwlDXrXke9uVchyBUH&csui=3)**:**
  + Edit an existing expense to change its amount or category.
  + Edit an expense to update its date or description.
  + Remove an attached receipt from an existing expense.
* [**Delete Expense**](https://www.google.com/search?sca_esv=c97b6394e1ab69a5&cs=0&sxsrf=AE3TifN7424HQMU6CjkEc8iYbQOzljYgDg%3A1759663680227&q=Delete+Expense&sa=X&ved=2ahUKEwie6_yD-oyQAxXvzqACHcBJAFsQxccNegQIPxAC&mstk=AUtExfCvsIOK3EMeOJzlXirOBN4bsFcYWtA1jlBmvx9glI30dv6wFlJ6_b73GxRGHh-JO5tq3rdgZqRT6-lzGsWKN6tXWWfUE7Irb3P16Col8Kyq8myjz4M2MKpaotpu7OcJ-4kvPTmcwj5VoYmCCMUSAu2XYxH2uFW14cX3_Gmg1Hj7a8h2oMU8fabGEE9Iuzzs4lhWq4ue1TQYDzYqB320C0aXkGpD6I8tfmbIfq-7dCq1tshoSZdWgJMfwfb7pYKu5n92cUYwlDXrXke9uVchyBUH&csui=3)**:**
  + Delete a single expense.
  + Attempt to delete an expense that is currently in use elsewhere (e.g., in a report).
* [**Receipt Management**](https://www.google.com/search?sca_esv=c97b6394e1ab69a5&cs=0&sxsrf=AE3TifN7424HQMU6CjkEc8iYbQOzljYgDg%3A1759663680227&q=Receipt+Management&sa=X&ved=2ahUKEwie6_yD-oyQAxXvzqACHcBJAFsQxccNegQIShAC&mstk=AUtExfCvsIOK3EMeOJzlXirOBN4bsFcYWtA1jlBmvx9glI30dv6wFlJ6_b73GxRGHh-JO5tq3rdgZqRT6-lzGsWKN6tXWWfUE7Irb3P16Col8Kyq8myjz4M2MKpaotpu7OcJ-4kvPTmcwj5VoYmCCMUSAu2XYxH2uFW14cX3_Gmg1Hj7a8h2oMU8fabGEE9Iuzzs4lhWq4ue1TQYDzYqB320C0aXkGpD6I8tfmbIfq-7dCq1tshoSZdWgJMfwfb7pYKu5n92cUYwlDXrXke9uVchyBUH&csui=3)**:**
  + Upload different file types for receipts (e.g., JPG, PDF).
  + Test receipt upload with large files.
  + Verify that invalid receipt file formats are rejected.
* [**Budget & Categories**](https://www.google.com/search?sca_esv=c97b6394e1ab69a5&cs=0&sxsrf=AE3TifN7424HQMU6CjkEc8iYbQOzljYgDg%3A1759663680227&q=Budget+%26+Categories&sa=X&ved=2ahUKEwie6_yD-oyQAxXvzqACHcBJAFsQxccNegQIUxAC&mstk=AUtExfCvsIOK3EMeOJzlXirOBN4bsFcYWtA1jlBmvx9glI30dv6wFlJ6_b73GxRGHh-JO5tq3rdgZqRT6-lzGsWKN6tXWWfUE7Irb3P16Col8Kyq8myjz4M2MKpaotpu7OcJ-4kvPTmcwj5VoYmCCMUSAu2XYxH2uFW14cX3_Gmg1Hj7a8h2oMU8fabGEE9Iuzzs4lhWq4ue1TQYDzYqB320C0aXkGpD6I8tfmbIfq-7dCq1tshoSZdWgJMfwfb7pYKu5n92cUYwlDXrXke9uVchyBUH&csui=3)**:**
  + Create custom expense categories.
  + Assign an expense to a custom category.
  + Check if the budget calculation accurately reflects the added expenses.
  + Verify that budget warnings trigger when approaching or exceeding the limit.
* [**Reporting**](https://www.google.com/search?sca_esv=c97b6394e1ab69a5&cs=0&sxsrf=AE3TifN7424HQMU6CjkEc8iYbQOzljYgDg%3A1759663680227&q=Reporting&sa=X&ved=2ahUKEwie6_yD-oyQAxXvzqACHcBJAFsQxccNegQIVRAC&mstk=AUtExfCvsIOK3EMeOJzlXirOBN4bsFcYWtA1jlBmvx9glI30dv6wFlJ6_b73GxRGHh-JO5tq3rdgZqRT6-lzGsWKN6tXWWfUE7Irb3P16Col8Kyq8myjz4M2MKpaotpu7OcJ-4kvPTmcwj5VoYmCCMUSAu2XYxH2uFW14cX3_Gmg1Hj7a8h2oMU8fabGEE9Iuzzs4lhWq4ue1TQYDzYqB320C0aXkGpD6I8tfmbIfq-7dCq1tshoSZdWgJMfwfb7pYKu5n92cUYwlDXrXke9uVchyBUH&csui=3)**:**
  + Generate a report for a specific date range.
  + Generate a report for a specific category.
  + Export reports in different formats (e.g., CSV, PDF).
* [**User Experience**](https://www.google.com/search?sca_esv=c97b6394e1ab69a5&cs=0&sxsrf=AE3TifN7424HQMU6CjkEc8iYbQOzljYgDg%3A1759663680227&q=User+Experience&sa=X&ved=2ahUKEwie6_yD-oyQAxXvzqACHcBJAFsQxccNegQITRAC&mstk=AUtExfCvsIOK3EMeOJzlXirOBN4bsFcYWtA1jlBmvx9glI30dv6wFlJ6_b73GxRGHh-JO5tq3rdgZqRT6-lzGsWKN6tXWWfUE7Irb3P16Col8Kyq8myjz4M2MKpaotpu7OcJ-4kvPTmcwj5VoYmCCMUSAu2XYxH2uFW14cX3_Gmg1Hj7a8h2oMU8fabGEE9Iuzzs4lhWq4ue1TQYDzYqB320C0aXkGpD6I8tfmbIfq-7dCq1tshoSZdWgJMfwfb7pYKu5n92cUYwlDXrXke9uVchyBUH&csui=3)**:**
  + Verify the accuracy of auto-categorization for expenses.
  + Test searching and filtering of expenses based on various criteria.

\*Test scenario on budget

* **Successful budget creation:** Create a budget with all required fields filled in, including a name, time period, and at least one category with a valid amount. The budget should save successfully.
* **Create a budget with multiple categories:** Create a budget that includes various expense categories, such as groceries, entertainment, and housing, with different amounts allocated to each.
* **Set a future budget:** Create a budget for a future period to confirm the system can handle scheduling budgets in advance.
* **Create a budget with varied income sources:** Test creating a budget that accounts for multiple sources of income, such as a primary paycheck and freelance earnings.
* **Duplicate a budget:** Confirm that an existing budget can be duplicated, allowing users to easily create a new budget based on a previous one.
* **Edit an existing budget:** Modify the budget name, time frame, or category amounts and ensure the changes are saved correctly.
* **Invalid budget amounts:** Attempt to create a budget with non-numeric, zero, or negative values for budget categories.
* **Missing required fields:** Attempt to save a budget without a name, or with no time period selected.
* **Overlapping budgets:** Try to create a new budget that overlaps with an existing one for the same user. The system should prevent this and provide an error message.
* **Exceeding maximum values:** Input an amount that exceeds a defined system limit (e.g., maximum integer value) to test how the system responds to oversized data.
* **Special characters in fields:** Create a budget using special characters in the name or other text fields to ensure the system handles them without errors.
* **User interface testing:** Confirm that the budget creation interface is clear, displays validation errors correctly, and is easy to navigate.

\*Test scenario on alerts

* Verify that a user receives an alert instantly for every successful transaction (deposit, withdrawal, transfer).
* Confirm that a user is notified immediately if a transaction fails, including the reason for the failure.
* Test that alerts are triggered correctly for different types of transfers, such as internal, external, and instant payments.
* Check that notifications include accurate details like the transaction amount, date, time, and recipient or sender.
* Receive an alert for any unusual or suspicious login attempts, especially from an unrecognized device or location.
* Test the notification for a successful or failed password reset attempt.
* Verify that an alert is sent to confirm changes to personal information, such as phone number or address.
* Test the alert for a low account balance or when an account reaches a predefined threshold.
* Confirm that the alert for an expired card is sent well before the expiration date.

\*negative scenario

* **nvalid threshold values:** A user tries to set an alert for a portfolio value of $0 or a negative number.
  + *Expected behavior:* The app should reject the input with an appropriate error message like "Value must be greater than zero.".
* **Unsupported characters:** A user enters special characters, emojis, or symbols into text fields for alert descriptions or names.
  + *Expected behavior:* The app should either sanitize the input or display an error, preventing data corruption or a crash.
* **Exceeding character limits:** The user inputs an alert name or description that is longer than the maximum allowed limit.
  + *Expected behavior:* The app should prevent further typing once the limit is reached or show an error message upon submission.
* **Setting alerts without required data:** A user attempts to create an alert without selecting a required field, such as the stock or crypto asset to monitor.
  + *Expected behavior:* The app should prevent the alert creation and display a "This field is required" error message.
* **Poor or no network connectivity:** The user loses internet connection just as an alert trigger condition is met.
  + *Expected behavior:* The app should queue the notification and deliver it once connectivity is restored. It should not crash or lose the event.
* **Alerts during app downtime:** The FinSmart server is undergoing scheduled maintenance or is experiencing a service outage.
  + *Expected behavior:* Alert conditions should be processed server-side and delivered after the service is restored, or a system status message should inform the user of the potential delay.

\*Bugs in finsmart app

1. when we sign up in finsmart app it shows the code of only four country it doed not shows the code of india +91 in mobile number it does not support and mobile number is compulsory add for sign up so new user can not complete the sign up process.

2.when we click on login with an email address then it shows the email address and password.when unkown user add its email and password it cannot login in app but at the same time when we use same unknown user name it automatically generate random password for that email.

3.when we click on login with gsm so it does not shows the other country like india +91 it shows only four country so unknown user of that country does not login with gsm.