Test cases

Doctor Appointment Booking

1. **Requirements :** Hospital name, doctor name, speciality, slot time and availability Consultation fee Slot confirmation and rescheduling

Hospital Name

Scenarios

1. **Valid Hospital Name**
   - **Test Scenario:** User selects a valid hospital name from the list.
   - **Expected Result:** The system displays available doctors and their specialties for the selected hospital.
   - **Example:** User selects "City Hospital" and sees "Dr. kumar - Cardiologist" and "Dr. mathan - Dermatologist".
2. **Invalid Hospital Name**
   - **Test Scenario:** User enters a hospital name not present in the list.
   - **Expected Result:** The system displays an error message or no results found.
   - **Example:** User enters "Unknown Hospital" and sees an error message "Hospital not found."
3. **No Hospital Name Selected**
   - **Test Scenario:** User tries to proceed without selecting a hospital name.
   - **Expected Result:** The system prompts the user to select a hospital.
   - **Example:** User clicks "Next" without selecting a hospital and sees "Please select a hospital."

2. Doctor Name

Scenarios

1. **Valid Doctor Name**
   - **Test Scenario:** User selects a valid doctor name from the list.
   - **Expected Result:** The system displays available slots for the selected doctor.
   - **Example:** User selects "Dr. kumar " and sees available slots from 10:00 AM to 3:00 PM.
2. **Invalid Doctor Name**
   - **Test Scenario:** User enters a doctor name not present in the list.
   - **Expected Result:** The system displays an error message or no results found.
   - **Example:** User enters "Dr. Unknown" and sees an error message "Doctor not found."
3. **No Doctor Name Selected**
   - **Test Scenario:** User tries to proceed without selecting a doctor name.
   - **Expected Result:** The system prompts the user to select a doctor.
   - **Example:** User clicks "Next" without selecting a doctor and sees "Please select a doctor."

3. Speciality

Scenarios

1. **Valid Speciality**
   - **Test Scenario:** User selects a valid specialty from the list.
   - **Expected Result:** The system displays doctors and their available slots for the selected specialty.
   - **Example:** User selects "Cardiology" and sees "Dr. Kumar" with available slots.
2. **Invalid Speciality**
   - **Test Scenario:** User enters a specialty not present in the list.
   - **Expected Result:** The system displays an error message or no results found.
   - **Example:** User enters "Unknown Specialty" and sees an error message "Specialty not found."
3. **No Speciality Selected**
   - **Test Scenario:** User tries to proceed without selecting a specialty.
   - **Expected Result:** The system prompts the user to select a specialty.
   - **Example:** User clicks "Next" without selecting a specialty and sees "Please select a specialty."

4. Slot Time and Availability

Scenarios

1. **Valid Slot Time**
   - **Test Scenario:** User selects a valid slot time.
   - **Expected Result:** The system confirms the slot and proceeds to payment.
   - **Example:** User selects "10:00 AM - 10:30 AM" and the system confirms availability.
2. **Unavailable Slot Time**
   - **Test Scenario:** User selects a slot time that is already booked.
   - **Expected Result:** The system displays a message indicating the slot is unavailable.
   - **Example:** User selects "11:00 AM - 11:30 AM" and sees "This slot is unavailable."
3. **No Slot Time Selected**
   - **Test Scenario:** User tries to proceed without selecting a slot time.
   - **Expected Result:** The system prompts the user to select a slot time.
   - **Example:** User clicks "Next" without selecting a slot time and sees "Please select a slot time."

5. Consultation Fee

Scenarios

1. **Valid Consultation Fee Display**
   - **Test Scenario:** System displays the correct consultation fee for the selected doctor and slot.
   - **Expected Result:** The consultation fee is shown and user can proceed to payment.
   - **Example:** User sees "Consultation Fee: Rs : 100" for Dr. Kumar.
2. **Incorrect Consultation Fee Display**
   - **Test Scenario:** System displays an incorrect consultation fee.
   - **Expected Result:** The system should not allow the user to proceed and show an error message.
   - **Example:** User sees "Consultation Fee: Rs : 200" instead of the correct Rs : 100.

6. Slot Confirmation

Scenarios

1. **Successful Slot Confirmation**
   - **Test Scenario:** User successfully confirms the slot and receives a confirmation message.
   - **Expected Result:** The system sends a confirmation message with appointment details.
   - **Example:** User receives "Your appointment with Dr. Kumar at 10:00 AM on June 25th is confirmed."
2. **Slot Confirmation Failure**
   - **Test Scenario:** Slot confirmation fails due to a system error.
   - **Expected Result:** The system displays an error message and does not confirm the slot.
   - **Example:** User sees "Slot confirmation failed. Please try again."

7. **Rescheduling**

Scenarios

1. **Successful Rescheduling**
   - **Test Scenario:** User successfully reschedules an appointment.
   - **Expected Result:** The system confirms the new slot and updates the appointment details.
   - **Example:** User reschedules from "June 25th, 10:00 AM" to "June 26th, 11:00 AM" and receives a new confirmation.
2. **Rescheduling to an Unavailable Slot**
   - **Test Scenario:** User tries to reschedule to a slot that is unavailable.
   - **Expected Result:** The system displays a message indicating the slot is unavailable.
   - **Example:** User tries to reschedule to "June 25th, 11:00 AM" and sees "This slot is unavailable."
3. **Rescheduling Without Selecting a New Slot**
   - **Test Scenario:** User tries to reschedule without selecting a new slot.
   - **Expected Result:** The system prompts the user to select a new slot.
   - **Example:** User clicks "Reschedule" without selecting a new slot and sees "Please select a new slot."

Summary of Testing Approach

1. **Test Preparation:**
   - Create test data for hospitals, doctors, specialties, slot times, and consultation fees.
   - Ensure test environment mimics the production environment.
2. **Test Execution:**
   - Execute the scenarios above using both valid and invalid inputs.
   - Document actual results and compare them with expected results.
3. **Defect Reporting:**
   - Log any discrepancies as defects.
   - Prioritize and fix defects before re-executing the test cases.
4. **User Acceptance Testing (UAT):**
   - Involve end-users to perform UAT and validate the system against real-world scenarios.

By following these detailed test scenarios, i can ensure a robust and reliable doctor appointment booking system.

Gym Workout Session Booking

1. **Requirements** **:** Check-in, session type, session time Trainer name, Location, Feedback, workout report

1. Check-in

Scenarios

1. **Valid Check-in**
   - **Test Scenario:** User successfully checks in using valid credentials.
   - **Expected Result:** The system confirms check-in and allows the user to proceed.
   - **Example:** User enters a valid membership ID and password and sees "Check-in successful."
2. **Invalid Check-in**
   - **Test Scenario:** User attempts to check in with invalid credentials.
   - **Expected Result:** The system displays an error message and denies access.
   - **Example:** User enters an incorrect membership ID or password and sees "Invalid credentials."
3. **Check-in without Credentials**
   - **Test Scenario:** User tries to check in without entering credentials.
   - **Expected Result:** The system prompts the user to enter credentials.
   - **Example:** User clicks "Check-in" without entering ID and password and sees "Please enter your membership ID and password."

2. Session Type

Scenarios

1. **Valid Session Type Selection**
   - **Test Scenario:** User selects a valid session type from the list.
   - **Expected Result:** The system displays available session times for the selected type.
   - **Example:** User selects "Yoga" and sees available session times for yoga classes.
2. **Invalid Session Type**
   - **Test Scenario:** User enters a session type not present in the list.
   - **Expected Result:** The system displays an error message or no results found.
   - **Example:** User enters "Piloxing" and sees "Session type not found."
3. **No Session Type Selected**
   - **Test Scenario:** User tries to proceed without selecting a session type.
   - **Expected Result:** The system prompts the user to select a session type.
   - **Example:** User clicks "Next" without selecting a session type and sees "Please select a session type."

3. Session Time

Scenarios

1. **Valid Session Time Selection**
   - **Test Scenario:** User selects a valid session time from the available slots.

- **Expected Result:** The system confirms the session time and allows the user to proceed.
- **Example:** User selects "10:00 AM - 11:00 AM" for a yoga session and sees a confirmation.

2. **Unavailable Session Time**
   - **Test Scenario:** User selects a session time that is already fully booked.
   - **Expected Result:** The system displays a message indicating the slot is unavailable.
   - **Example:** User selects "10:00 AM - 11:00 AM" and sees "This slot is fully booked."
3. **No Session Time Selected**
   - **Test Scenario:** User tries to proceed without selecting a session time.
   - **Expected Result:** The system prompts the user to select a session time.
   - **Example:** User clicks "Next" without selecting a session time and sees "Please select a session time."

4. Trainer Name

Scenarios

1. **Valid Trainer Name Selection**
   - **Test Scenario:** User selects a valid trainer from the list.
   - **Expected Result:** The system displays the trainer's available slots and session types.
   - **Example:** User selects "Trainer John" and sees his available session slots for yoga and cardio.
2. **Invalid Trainer Name**
   - **Test Scenario:** User enters a trainer name not present in the list.
   - **Expected Result:** The system displays an error message or no results found.
   - **Example:** User enters "Trainer Unknown" and sees "Trainer not found."
3. **No Trainer Name Selected**
   - **Test Scenario:** User tries to proceed without selecting a trainer name.
   - **Expected Result:** The system prompts the user to select a trainer.
   - **Example:** User clicks "Next" without selecting a trainer and sees "Please select a trainer."

5. **Location**

Scenarios

1. **Valid Location Selection**
   - **Test Scenario:** User selects a valid location from the list.
   - **Expected Result:** The system displays available trainers and session types for the selected location.
   - **Example:** User selects "Downtown Gym" and sees available trainers and session types for that location.
2. **Invalid Location**
   - **Test Scenario:** User enters a location not present in the list.
   - **Expected Result:** The system displays an error message or no results found.
   - **Example:** User enters "Unknown Location" and sees "Location not found."
3. **No Location Selected**
   - **Test Scenario:** User tries to proceed without selecting a location.
   - **Expected Result:** The system prompts the user to select a location.
   - **Example:** User clicks "Next" without selecting a location and sees "Please select a location."

6. Feedback

Scenarios

1. **Submit Feedback**
   - **Test Scenario:** User submits feedback for a completed session.
   - **Expected Result:** The system confirms the submission and displays a thank you message.
   - **Example:** User rates the session 5 stars and writes "Great workout!" and sees "Thank you for your feedback."
2. **Submit Feedback without Rating**
   - **Test Scenario:** User tries to submit feedback without providing a rating.
   - **Expected Result:** The system prompts the user to provide a rating.
   - **Example:** User writes a comment but does not rate the session and sees "Please provide a rating."
3. **Submit Feedback with Inappropriate Content**
   - **Test Scenario:** User submits feedback with inappropriate content.
   - **Expected Result:** The system filters out inappropriate content and prompts the user to revise.
   - **Example:** User writes inappropriate comments and sees "Your feedback contains inappropriate content. Please revise."

7. Workout Report

Scenarios

1. **Generate Workout Report**
   - **Test Scenario:** User requests a workout report for a specific period.
   - **Expected Result:** The system generates and displays the workout report for the selected period.
   - **Example:** User requests a report for June 2024 and sees a detailed summary of all workouts completed in that month.
2. **Generate Workout Report with Invalid Date Range**
   - **Test Scenario:** User enters an invalid date range for the workout report.
   - **Expected Result:** The system displays an error message and prompts the user to enter a valid date range.
   - **Example:** User selects a date range from July 2024 to June 2024 and sees "Invalid date range. Please select a valid period."
3. **No Workout Report Available**
   - **Test Scenario:** User requests a workout report but no sessions were completed in the selected period.
   - **Expected Result:** The system displays a message indicating no data is available.
   - **Example:** User requests a report for a period with no workouts and sees "No workouts found for the selected period."

Summary of Testing Approach

1. **Test Preparation:**
   - Create test data for sessions, trainers, locations, feedback entries, and workout logs.
   - Ensure the test environment closely resembles the production environment.
2. **Test Execution:**
   - Execute the scenarios above using both valid and invalid inputs.

   - Document actual results and compare them with expected results.
3. **Defect Reporting:**
   - Log any discrepancies as defects.
   - Prioritize and fix defects before re-executing the test cases.
4. **User Acceptance Testing (UAT):**
   - Involve end-users to perform UAT and validate the system against real-world scenarios.

By following these detailed test scenarios, i can ensure a robust and user-friendly gym workout session booking system.

Subscription

Requirements： 30-day free trial period

1. Start of the 30-Day Free Trial

Scenarios

- **Start Free Trial with Valid Information :**
  - **Test Scenario:** User starts the 30-day free trial with valid personal and payment information.
  - **Expected Result:** The system starts the free trial, displays the trial end date, and sends a confirmation email.
  - **Example:** User enters valid credit card information and sees "Your 30-day free trial has started. It will end on July 25th."
- **Start Free Trial with Invalid Payment Information :**
  - **Test Scenario:** User tries to start the free trial with invalid payment information.
  - **Expected Result:** The system displays an error message and does not start the trial.
  - **Example:** User enters an expired credit card and sees "Invalid payment information. Please try again."
- **Start Free Trial Without Payment Information :**
  - **Test Scenario:** User tries to start the free trial without providing payment information.
  - **Expected Result:** The system prompts the user to enter payment information.
  - **Example:** User clicks "Start Free Trial" without entering payment details and sees "Please enter your payment information."

2. During the 30-Day Free Trial

Scenarios

1. **Access Subscription Benefits During Free Trial**
   - **Test Scenario:** User accesses the subscription benefits during the free trial period.
   - **Expected Result:** The system grants access to all benefits available to subscribers.
   - **Example:** User accesses premium content and sees it without restrictions.
2. **View Trial Period Remaining**
   - **Test Scenario:** User checks the remaining days of the free trial period.
   - **Expected Result:** The system displays the number of days remaining in the free trial.
   - **Example:** User checks the trial status and sees "15 days remaining in your free trial."
3. **Cancel Free Trial Before Expiration**
   - **Test Scenario:** User cancels the free trial before the 30 days are up.
   - **Expected Result:** The system cancels the trial, displays a cancellation confirmation, and sends a confirmation email.
   - **Example:** User cancels on day 20 and sees "Your free trial has been canceled. You will lose access on July 15th."

3. End of the 30-Day Free Trial

Scenarios

1. **Automatic Transition to Paid Subscription**
   - **Test Scenario:** The free trial period ends and the subscription transitions to a paid plan automatically.

- **Expected Result:** The system charges the user’s payment method and sends a confirmation email for the paid subscription.
- **Example:** On day 31, the user’s credit card is charged, and they receive an email stating "Your subscription has been activated."

2. **Failed Payment at End of Free Trial**
   - **Test Scenario:** The system tries to charge the payment method at the end of the trial, but the payment fails.
   - **Expected Result:** The system notifies the user of the payment failure and suspends the subscription benefits.
   - **Example:** User's credit card is declined, and they see "Payment failed. Please update your payment information."
3. **Manual Extension Request Before Trial End**
   - **Test Scenario:** User requests an extension of the trial period before it ends.
   - **Expected Result:** The system processes the request and extends the trial period if allowed.
   - **Example:** User requests a 7-day extension on day 29 and sees "Your trial has been extended until August 1st."

4. After the 30-Day Free Trial

Scenarios

1. **Access Benefits After Transition to Paid Subscription**
   - **Test Scenario:** User accesses the subscription benefits after the trial has transitioned to a paid subscription.
   - **Expected Result:** The system grants continued access to all subscription benefits.
   - **Example:** User accesses premium content on day 31 and sees it without interruptions.
2. **Cancel Paid Subscription After Free Trial**
   - **Test Scenario:** User cancels the subscription after the free trial has ended and transitioned to a paid plan.
   - **Expected Result:** The system processes the cancellation and stops future billing.
   - **Example:** User cancels on day 35 and sees "Your subscription will end on the next billing date."
3. **Reactivate Subscription After Cancellation**
   - **Test Scenario:** User reactivates their subscription after previously canceling it.
   - **Expected Result:** The system reactivates the subscription and charges the user if applicable.
   - **Example:** User reactivates their subscription on day 40 and sees "Your subscription has been reactivated."

Summary of Testing Approach

1. **Test Preparation:**
   - Create test accounts with varying payment methods and personal information.
   - Set up a test environment mimicking the production environment.
2. **Test Execution:**
   - Execute the scenarios above using both valid and invalid inputs.
   - Document actual results and compare them with expected results.
3. **Defect Reporting:**
   - Log any discrepancies as defects.
   - Prioritize and fix defects before re-executing the test cases.
4. **User Acceptance Testing (UAT):**
   - Involve end-users to perform UAT and validate the system against real-world scenarios.

By following these detailed test scenarios, I can ensure a smooth and reliable subscription experience for users, especially during and after the 30-day free trial period.

Product Feedback and Ratings

Requirements : Webpage displaying customer feedback and ratings

1. Submitting Feedback and Ratings

Scenarios

1. **Submit Valid Feedback and Rating**
   - **Test Scenario:** User submits valid feedback and a rating.
   - **Expected Result:** The system saves the feedback and rating, and it is displayed on the product page.
   - **Example:** User gives a 5-star rating and writes "Great product!" The feedback appears on the page.
2. **Submit Feedback without Rating**
   - **Test Scenario:** User submits feedback without providing a rating.
   - **Expected Result:** The system prompts the user to provide a rating.
   - **Example:** User writes "Great product!" but doesn't select a rating. They see "Please provide a rating."
3. **Submit Rating without Feedback**
   - **Test Scenario:** User submits a rating without writing feedback.
   - **Expected Result:** The system saves the rating, and it is displayed on the product page.
   - **Example:** User gives a 4-star rating without writing feedback. The rating appears on the page.
4. **Submit Feedback with Invalid Characters**
   - **Test Scenario:** User submits feedback containing invalid characters or code.
   - **Expected Result:** The system filters out invalid characters and prompts the user to revise the feedback.
   - **Example:** User writes "<script>alert('test'); </script>" and sees "Invalid characters detected. Please revise your feedback."

2. Viewing Feedback and Ratings

Scenarios

1. **View Feedback and Ratings for a Product**
   - **Test Scenario:** User views the product page with feedback and ratings.
   - **Expected Result:** The system displays all feedback and ratings for the product.
   - **Example:** User sees a list of feedback and ratings, such as "5 stars - Great product!" and "4 stars - Good value."
2. **View Feedback and Ratings with No Feedback Available**
   - **Test Scenario:** User views the product page for a product with no feedback.
   - **Expected Result:** The system displays a message indicating no feedback is available.
   - **Example:** User sees "No feedback available for this product."

3. Filtering Feedback and Ratings

Scenarios

1. **Filter Feedback by Rating**
   - **Test Scenario:** User filters feedback to see only certain ratings (e.g., 5-star ratings).
   - **Expected Result:** The system displays only the feedback with the selected rating.
   - **Example:** User selects "5 stars" filter and sees only 5-star feedback.
2. **Filter Feedback by Date**
   - **Test Scenario:** User filters feedback to see the most recent feedback first.
   - **Expected Result:** The system sorts and displays feedback by date.
   - **Example:** User selects "Most recent" filter and sees the latest feedback at the top.

4. Sorting Feedback and Ratings

Scenarios

1. **Sort Feedback by Highest Rating**
   - **Test Scenario:** User sorts feedback to see the highest-rated feedback first.
   - **Expected Result:** The system sorts and displays feedback by rating, highest first.
   - **Example:** User selects "Highest rating" sort option and sees 5-star feedback at the top.
2. **Sort Feedback by Lowest Rating**
   - **Test Scenario:** User sorts feedback to see the lowest-rated feedback first.
   - **Expected Result:** The system sorts and displays feedback by rating, lowest first.
   - **Example:** User selects "Lowest rating" sort option and sees 1-star feedback at the top.

5. Handling Edge Cases

Scenarios

1. **Submit Duplicate Feedback**
   - **Test Scenario:** User tries to submit duplicate feedback for the same product.
   - **Expected Result:** The system detects the duplicate and prevents submission.
   - **Example:** User submits "Great product!" twice and sees "You have already submitted this feedback."
2. **Submit Feedback Exceeding Character Limit**
   - **Test Scenario:** User submits feedback exceeding the maximum allowed character limit.
   - **Expected Result:** The system displays an error message and prevents submission.
   - **Example:** User writes a feedback that is too long and sees "Feedback exceeds the character limit."
3. **System Handling of No Internet Connection During Submission**
   - **Test Scenario:** User tries to submit feedback with no internet connection.
   - **Expected Result:** The system displays a connection error message and does not submit the feedback.
   - **Example:** User clicks "Submit" without internet and sees "No internet connection. Please try again."
4. **Handling Special Characters in Feedback**
   - **Test Scenario:** User submits feedback containing special characters.
   - **Expected Result:** The system properly displays special characters without issues.

- **Example:** User writes "Great product! #awesome" and sees the feedback displayed correctly.

Summary of Testing Approach

1. **Test Preparation:**
   - Create test accounts and test products with varying feedback and ratings.
   - Ensure the test environment mimics the production environment.
2. **Test Execution:**
   - Execute the scenarios above using both valid and invalid inputs.
   - Document actual results and compare them with expected results.
3. **Defect Reporting:**
   - Log any discrepancies as defects.
   - Prioritize and fix defects before re-executing the test cases.
4. **User Acceptance Testing (UAT):**
   - Involve end-users to perform UAT and validate the system against real-world scenarios.

By following these detailed test scenarios, I can ensure a robust and user-friendly system for displaying and managing customer feedback and ratings on a product webpage.

User Creation with KYC

Requirements : Validate email, phone, and documents "I am not a robot" verification

1. Email Validation

Scenarios

1. **Valid Email Format**
   - **Test Scenario:** User enters a valid email address format.
   - **Expected Result:** The system accepts the email address and proceeds to the next step.
   - **Example:** User enters "kamaleshwaran3514@gmail.com" and moves forward.
2. **Invalid Email Format**
   - **Test Scenario:** User enters an invalid email address format (e.g., missing @ or .com).
   - **Expected Result:** The system displays an error message and does not proceed.
   - **Example:** User enters "invalid.email" and sees "Invalid email format. Please enter a valid email address."
3. **Duplicate Email**
   - **Test Scenario:** User tries to register with an email address already in use.
   - **Expected Result:** The system notifies the user that the email is already registered.
   - **Example:** User enters an email that is already registered and sees "Email already in use. Please use a different email."

2. Phone Number Validation

Scenarios

1. **Valid Phone Number Format**
   - **Test Scenario:** User enters a valid phone number format.
   - **Expected Result:** The system accepts the phone number and allows the user to proceed.
   - **Example:** User enters "+1234567890" or "123-456-7890" and moves forward.
2. **Invalid Phone Number Format**
   - **Test Scenario:** User enters an invalid phone number format (e.g., alphabetic characters, incorrect length).
   - **Expected Result:** The system displays an error message and does not proceed.
   - **Example:** User enters "abc123" or "12345" and sees "Invalid phone number format. Please enter a valid phone number."
3. **Duplicate Phone Number**
   - **Test Scenario:** User tries to register with a phone number already associated with another account.
   - **Expected Result:** The system notifies the user that the phone number is already in use.
   - **Example:** User enters a phone number that is already registered and sees "Phone number already in use. Please use a different phone number."

3. Document Verification (KYC)

Scenarios

1. **Upload Valid Document**
   - **Test Scenario:** User uploads a valid document for KYC verification (e.g., passport, ID card).
   - **Expected Result:** The system accepts the document and proceeds with verification.
   - **Example:** User uploads a clear and valid passport scan.
2. **Upload Invalid Document**
   - **Test Scenario:** User uploads an invalid document (e.g., expired document, blurry image).
   - **Expected Result:** The system displays an error message and prompts the user to upload a valid document.
   - **Example:** User uploads an expired ID card scan and sees "Invalid document. Please upload a valid document."
3. **Document Size and Format**
   - **Test Scenario:** User uploads a document that exceeds the allowed size or is in an unsupported format.
   - **Expected Result:** The system notifies the user of size/format requirements and does not proceed.
   - **Example:** User uploads a 10MB file or a document in a format other than PDF or JPEG and sees "Document size exceeds limit. Please upload a file under 5MB in PDF or JPEG format."

4. "I am not a robot" Verification

Scenarios

1. **Successful CAPTCHA Verification**
   - **Test Scenario:** User successfully completes the "I am not a robot" verification.
   - **Expected Result:** The system confirms the verification and allows the user to proceed.
   - **Example:** User completes the CAPTCHA challenge and proceeds to the next step.
2. **Failed CAPTCHA Verification**
   - **Test Scenario:** User fails to complete the CAPTCHA challenge correctly.
   - **Expected Result:** The system displays an error message and prompts the user to retry the CAPTCHA.
   - **Example:** User incorrectly identifies traffic lights in the CAPTCHA and sees "CAPTCHA verification failed. Please try again."
3. **No CAPTCHA Available**
   - **Test Scenario:** CAPTCHA service is unavailable or not loaded properly.
   - **Expected Result:** The system displays a message indicating the inability to verify and prompts the user to try again later.
   - **Example:** CAPTCHA fails to load due to network issues, and the user sees "CAPTCHA service unavailable. Please try again later."

Summary of Testing Approach

1. **Test Preparation:**
   - Set up test environments with varying email formats, phone numbers, and mock document verification services.

    - Prepare test scenarios covering edge cases and failure conditions.
2. **Test Execution:**
    - Execute the scenarios above using both valid and invalid inputs.
    - Verify system responses and error handling.
3. **Defect Reporting:**
    - Document any defects encountered during testing.
    - Prioritize and fix critical defects before deployment.
4. **User Acceptance Testing (UAT):**
    - Involve end-users or stakeholders to perform UAT and validate the system's usability and reliability.

By following these detailed test scenarios,I can ensure that the user creation process with KYC requirements, email and phone validation, document verification, and CAPTCHA verification is robust and user-friendly.

Product Booking with Exchange Offer

Requirements : 50% off on the old same product Base price, max discount Order return and cancellation options

1. Exchange Offer Validation

Scenarios

1. **Valid Exchange Offer**
   - **Test Scenario:** User selects the same product for exchange and applies the exchange offer.
   - **Expected Result:** The system calculates and applies a 50% discount on the base price of the exchanged product.
   - **Example:**
     - Base price of new product: Rs : 100
     - Old product's base price eligible for exchange: Rs : 80
     - Calculation: Discount applied = 50% of Rs : 80 = Rs : 40
     - Total price after exchange offer: Rs : 100 - Rs : 40 = Rs : 60
2. **Invalid Exchange Offer (Different Product)**
   - **Test Scenario:** User tries to apply the exchange offer with a different product.
   - **Expected Result:** The system does not apply the exchange offer discount.
   - **Example:** User selects a different product for exchange and sees "Exchange offer is only valid for the same product."
3. **Missing Exchange Product Information**
   - **Test Scenario:** User does not provide details of the old product for exchange.
   - **Expected Result:** The system prompts the user to provide information about the old product.
   - **Example:** User selects exchange offer but does not specify the old product details. They see "Please provide details of the old product for exchange."

2. Base Price and Maximum Discount Calculation

Scenarios

1. **Base Price Calculation**
   - **Test Scenario:** User checks the calculation of the base price without any discounts or offers.
   - **Expected Result:** The system correctly displays the base price of the product.
   - **Example:** Base price of the product: Rs : 100
2. **Maximum Discount Calculation**
   - **Test Scenario:** User checks the maximum discount applicable on the base price.
   - **Expected Result:** The system calculates and applies the maximum discount allowed (e.g., 50% of the base price).
   - **Example:** Maximum discount allowed: 50% of Rs : 100 = Rs : 50
3. **Exceeding Maximum Discount**
   - **Test Scenario:** User tries to apply a discount that exceeds the maximum allowed.
   - **Expected Result:** The system applies the maximum discount allowed and notifies the user.

- **Example:** User tries to apply a Rs : 60 discount on a Rs : 100 product, but sees "Maximum discount allowed is Rs : 50."

3. Order Return and Cancellation Options

Scenarios

1. **Order Return Policy**
    - **Test Scenario:** User wants to return the ordered product within the return period.
    - **Expected Result:** The system allows the user to initiate a return and provides return instructions.
    - **Example:** Return period: 30 days from delivery. User initiates return on day 25 and sees "Return initiated successfully."
2. **Order Cancellation Before Shipment**
    - **Test Scenario:** User requests cancellation of the order before it is shipped.
    - **Expected Result:** The system cancels the order and processes a refund if applicable.
    - **Example:** User cancels the order immediately after placing it and sees "Order canceled successfully. Refund in progress."
3. **Order Cancellation After Shipment**
    - **Test Scenario:** User requests cancellation of the order after it has been shipped.
    - **Expected Result:** The system provides instructions for returning the shipped product and issues a refund upon return.
    - **Example:** User receives the order, decides to cancel it, and sees "Return initiated. Refund will be processed upon receipt of returned item."

Summary of Testing Approach

1. **Test Preparation:**
    - Configure test products with defined base prices and exchange offer rules.
    - Set up test scenarios covering valid and invalid inputs for each functionality.
2. **Test Execution:**
    - Execute the scenarios using both typical and edge-case scenarios.
    - Verify calculations, discounts, and system responses.
3. **Defect Reporting:**
    - Document any discrepancies or defects encountered during testing.
    - Prioritize and address critical issues promptly.
4. **User Acceptance Testing (UAT):**
    - Involve end-users to perform UAT and ensure the functionality meets their expectations and requirements.

By following these detailed test scenarios, I can ensure thorough testing of the product booking process with exchange offers, base price calculations, maximum discounts, and order return/cancellation options, ensuring a seamless and user-friendly experience for customers.