Testing Protocol

Key Feature 1: Creating An Account

Test Procedure

1. Navigate to the login page and click on the "Sign Up" button.
2. Input a full name, valid email address, and password that meets the platform's security requirements. Submit the form.
3. Try to enter Invalid Information:
    - Test with missing fields (e.g., leave the email blank).
    - Use an invalid email format (e.g., "user@domain").
    - Enter a password that doesn't meet requirements (e.g., fewer than the minimum number of characters).
4. Try signing up using an email address already registered on the platform.

Expected Results

1. Upon input of valid information:
    - The system creates an account.
    - The user sees a message confirming account creation.
2. Upon input of invalid information:
    - Users are shown clear error messages for each issue (e.g., "Please enter a valid email address").
    - The account is not created, and the user remains on the sign-up page.
3. Duplicate Account:
    - The user is prevented from creating a duplicate account.
    - A message appears, such as "An account with this email already exists. Please log in."

Key Feature 2: User Login Functionality

Test Procedure:

1. Attempt to login with valid username and password.
2. Attempt to login with incorrect credentials.
3. Test session persistence by logging in, refreshing the page, and verifying access remains.

Expected Results:

1. Successful login grants access to authenticated pages like "My Events" and "Profile."
2. Incorrect credentials show an error message and deny access.
3. Session persists upon page refresh.

Workarounds for Bugs:

- If session persistence fails, implement additional client-side storage.
- For login errors, ensure proper error handling and user feedback in the UI.

Key Feature 3: Profile Information

Test Procedure

1. Access the Profile Page:
    - Log in to the platform and navigate to the profile page via the menu.
2. Enter Profile Details:
    - Fill in all profile fields with valid inputs:
        - Full Name: "John Doe"
        - Pronouns: "They/Them"
        - Social Media Link: A valid URL (e.g., "https://www.twitter.com/johndoe").
        - Profile Image: Upload a valid image file (e.g., .jpg, .png).
3. Test Validation:
    - Save the profile.
    - Enter invalid inputs:
        - Social Media Link: Use an invalid URL format (e.g., "not_a_link").
        - Profile Image: Attempt to upload a non-image file (e.g., .pdf).
4. Click the "Save Profile" button and verify the response.
5. Refresh the profile page and ensure all entered details persist.

Expected Results

1. Valid Inputs:
    - All entered details are successfully saved.
    - The profile page updates to display the saved information.
    - After refreshing, the saved data remains intact and properly formatted.
2. Invalid Inputs:
    - Clear error messages are shown for invalid entries (e.g., "Please enter a valid URL" for social media links).
    - Invalid or blank fields do not overwrite existing profile information.
3. Profile Image:
    - A valid image file uploads successfully and displays as the profile picture.
    - Non-image files or excessively large images trigger an error message and prevent saving.

Key Feature 4: RSVP to Event

Test Procedure:

1. RSVP to an event (once logged in).
2. Get tickets for an event.

3. Verify events appear in "My Events" and a notification is sent once RSVP’d.
4. Try RSVP functionality without logging in.

Expected Results:

1. “Get tickets” button redirects to a ticketing site (e.g., Ticketmaster)
2. “RSVP” button adds the event to "My Events."
3. Notification is triggered in the Notification Center upon clicking the “RSVP” button.
4. Unauthenticated users attempting to RSVP are prompted to log in.

Workarounds for Bugs:

- Ensure API calls to Ticketmaster are monitored and retried on failure.