Generated Test Cases via gemma2-9b-it LLM Model and Few-shot Prompting Technique

---

Data - 1

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

User Profiles

Feature description -

Each user can create a personalized profile that includes information such as their name, bio, profile picture, location, interests, and other relevant details. Profiles serve as a digital representation of individuals on the platform, allowing others to learn more about them.

Userstory title -

As a user, I want to create a personalized profile so that I can share my information with others on the platform.

Acceptance criteria -

['Users can add and edit their name, bio, profile picture, location, and interests.', 'The profile picture should support standard image formats.', 'Users can set privacy settings for each field.', 'The profile must be saved successfully.', 'Users should receive a confirmation upon saving their profile.']

Generated Test Cases:

Here are 5 test cases for the Social Media Platform's User Profiles feature, adhering to the provided template and covering all acceptance criteria:

Test Case 1

Test Case ID: sM87j3

Test Case Title: Verify that users can add and edit their name

Test Case Description: This test case verifies the functionality of adding and editing a user's name in their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in to the social media platform.

Test Data: Valid user names (e.g., "John Doe", "Jane Smith").

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Edit Profile" button.
3. Modify the existing name field with a valid name.
4. Click the "Save Changes" button.

Postconditions:

- The profile page is updated with the new name.

Expected Result: The system successfully updates the user's name in the profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: pL52k1

Test Case Title: Verify that users can add and edit their bio

Test Case Description: This test case verifies the functionality of adding and editing a user's bio in their profile.

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

- User is logged in to the social media platform.

Test Data: Various bio content (e.g., short descriptions, longer narratives).

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Edit Profile" button.
3. Enter or modify the bio field with different content.
4. Click the "Save Changes" button.

Postconditions:

- The profile page is updated with the new bio.

Expected Result: The system successfully saves the user's input as their bio on the profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: vG9t6y

Test Case Title: Verify that users can add and edit their profile picture

Test Case Description: This test case verifies the functionality of adding and changing a user's profile picture.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in to the social media platform.

Test Data: Images in supported formats (e.g., JPG, PNG) of various sizes.

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Edit Profile" button.
3. Click on the "Change Profile Picture" button and select an image file.
4. Verify the profile picture display updates with the new image.

Postconditions:

- The profile page is updated with the new profile picture.

Expected Result: The system successfully uploads and displays the selected image as the user's profile picture.

Severity: Major

Type of Testing: Functionality

Test Case Approach: Positive

Test Case 4

Test Case ID: q2R4w9

Test Case Title: Verify that users can set privacy settings for profile fields

Test Case Description: This test case verifies the ability of users to control the visibility of different profile fields.

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

- User is logged in to the social media platform.

Test Data: Various privacy settings options (e.g., public, private, friends only).

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Edit Profile" button.
3. Adjust the privacy settings for individual fields (name, bio, location, interests).
4. Verify the selected privacy settings are reflected in the profile settings.

Postconditions:

- Profile settings are updated with chosen privacy levels.

Expected Result: The system allows users to selectively control the visibility of their profile data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: n7B3j5

Test Case Title: Verify that the user receives a confirmation upon saving their profile

Test Case Description: This test case verifies that the user receives a success notification after saving changes to their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in to the social media platform.

Test Data: Changes made to any profile field.

Test Steps:

1. Navigate to the user's profile page.
2. Edit any profile field.
3. Click the "Save Changes" button.

Postconditions:

- A confirmation message is displayed to user.

Expected Result: The system displays a clear confirmation message or notification to the user, indicating that their profile changes have been successfully saved.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 2

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

News Feed

Feature description -

The news feed is a central feature of the social media platform where users can view a curated stream of content from accounts they follow or are connected to. It typically displays posts, updates, photos, videos, and other media shared by friends, family, brands, and other users in chronological or algorithmically determined order.

Userstory title -

As a user, I want to see a curated news feed so that I can stay updated with content from my connections.

Acceptance criteria -

['The news feed displays posts from followed accounts.', 'Content is shown in chronological or algorithmic order.', 'Users can interact with posts (like, comment, share).', 'The feed updates in real-time.', 'Users can filter content based on type or topic.']

Generated Test Cases:

Here are 5 test cases for the provided Social Media Platform User Story, following the template you've provided:

Test Case 1

Test Case ID: nsfD34

Test Case Title: Verify that the news feed displays posts from followed accounts

Test Case Description: This test case ensures that the news feed only shows posts from accounts the user is following, excluding posts from accounts they haven't followed.

Test Suite: News Feed Functionality

Test Priority: High

Preconditions:

* User is logged into the platform.

* User has followed at least two different accounts.

Test Data:

* Data from the two followed accounts.

Test Steps:

1. Navigate to the user's news feed.

2. Verify that the posts displayed are from the two followed accounts.

3. Attempt to access the news feed of a user who is not followed.

4. Verify that no posts from the unfollowed user appear in the feed.

Postconditions: (None)

Expected Result: The news feed displays only posts from accounts that the user is currently following.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: ghj76k

Test Case Title: Verify that content is shown in chronological or algorithmic order

Test Case Description: This test case determines whether the news feed displays content in chronological order or utilizes an algorithmic ranking system.

Test Suite: News Feed Functionality

Test Priority: Medium

Preconditions:

Test Data:

* 5 posts with distinct timestamps from various followed accounts.

Test Steps:

1. Refresh the news feed page.

2. Observe the order in which the posts appear.

3. Note the timestamps of each post.

4. Verify that the posts are displayed either in strictly chronological order based on timestamp or in accordance with an algorithmic ranking system.

Postconditions: (None)

Expected Result: The posts are displayed either in strict chronological order or according to a clearly defined algorithmic ranking system.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: qwe43r

Test Case Title: Verify users can interact with posts (like, comment, share)

Test Case Description: This test case validates that users can successfully interact with posts by liking, commenting, and sharing.

Test Suite: News Feed Interactions

Test Priority: High

Preconditions:

* User is logged in.

* There are at least 2 posts visible in the news feed.

Test Data:

* Text for a comment

* Optional: A photo or small video for sharing

Test Steps:

1. Select a post in the news feed.

2. Click on the "like" button.

3. Click on the "comment" button and enter a comment.

4. Click on the "share" button and share the post.

5. Successfully completing the actions above confirms both the feature's functionality and potential unforeseen UI or data handling issues.

Postconditions:

* The like count for the selected post increases by one.

* The commented text appears under the selected post.

* The shared post is visible on the user's profile or timeline, depending on platform settings.

Expected Result: All user interactions with the post are recorded and reflected accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: erty89uio

Test Case Title: Verify that the feed updates in real-time

Test Case Description: This test case ensures the news feed dynamically updates to reflect new posts from followed accounts as they are published.

Test Suite: News Feed Updates

Test Priority: High

Preconditions:

* User is logged in.

* At least one account the user follows is actively posting.

Test Data:

* A new post generated by one of the followed accounts.

Test Steps:

1. Open the news feed page.

2. Observe the existing posts.

3. While monitoring the news feed, acquire a new post by the followed account (e.g., through a notification).

4. Verify that the new post appears in the news feed within a reasonable timeframe (seconds to minutes).

5. Repeat steps 2 to 4 multiple times.

Postconditions: (None)

Expected Result: The news feed seamlessly incorporates new posts as they become available, demonstrating real-time update functionality.

Severity: Major

Type of Testing: Performance, Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: asdfsdf

Test Case Title: Verify that users can filter content based on type or topic

Test Case Description: This test case confirms the news feed allows users to filter content based on post type (e.g., photos, videos) or topics of interest.

Test Suite: News Feed Filtering

Test Priority: Medium

Preconditions:

* The platform supports filtering posts by type or topic (either as core functionality or a future enhancement).

* A diverse set of posts is present: some photos, some videos, some text posts, related to various topics.

Test Data:

* At least three different post types (text, photo, video)

 * Users should also have a way to select specific topic filters to test

Test Steps:

1. Navigate to the news feed.

2. Click on the filter options (e.g., "All," "Photos," "Videos").

3. Verify that the feed displays only posts matching the selected filter.

4. Apply different topic filters (if available).

5. Confirm that the feed accurately reflects the selected topic filters.

Postconditions: (None)

Expected Result: The filtration mechanisms effectively refine the news feed based on user preferences, displaying only relevant content.

Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive

Data - 3

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Messaging

Feature description -

Messaging enables direct communication between users through private messages or group chats. Users can send text messages, multimedia files, emojis, stickers, and other forms of communication in real-time or asynchronously, fostering private conversations and connections.

Userstory title -

As a user, I want to send private messages so that I can communicate directly with my friends.

Acceptance criteria -

['Users can send text messages, images, and videos.', 'Messages can be sent in real-time.', 'Users can start private conversations or group chats.', 'Messages should show read receipts.', 'Users should be notified of new messages.']

Generated Test Cases:

Here are 5 test cases for the provided user story and acceptance criteria:

Test Case 1

* **Test Case ID:** MSG-001
* **Test Case Title:** Verify that users can send text messages.
* **Test Case Description:** This test case will verify the functionality of sending text messages between two users on the platform. It will ensure that the message is delivered successfully and is displayed in the recipient's inbox.
* **Test Suite:** Messaging
* **Test Priority:** High
* **Preconditions:**
    * Two registered users are logged into their accounts.
* **Test Data:** Sample text messages.
* **Test Steps:**
    1. Open the messaging interface.

    2. Select the recipient from the contact list.
    3. Compose a text message.
    4. Send the message.
* **Postconditions:**
    * The message is sent successfully.
    * The recipient receives the message in their inbox.
* **Expected Result:** The message is delivered to the recipient without errors. The recipient can view the message in their inbox.

* **Severity:** Critical
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** MSG-002
* **Test Case Title:** Verify that users can send images and videos.
* **Test Case Description:** This test case will confirm that users can attach and send images and videos in private messages. It will check if these media files are successfully received and displayed by the recipient.
* **Test Suite:** Messaging
* **Test Priority:** High
* **Preconditions:**
    * Two registered users are logged into their accounts.
    * The sender has images or videos available on their device.
* **Test Data:** Sample images and videos.
* **Test Steps:**
    1. Open the messaging interface.

    2. Select the recipient.
    3. Choose an image or video to attach.
    4. Send the message containing the media file.
* **Postconditions:**
    * The message is sent with the attached media.
    * The recipient receives the message with the attached media.
* **Expected Result:** The image or video is successfully attached to the message and sent to the recipient. The recipient can view the attached media within the chat.

* **Severity:** Critical
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** MSG-003
* **Test Case Title:** Verify real-time message delivery.
* **Test Case Description:** This test case will confirm that messages are delivered in real-time, providing an immediate response to the sender.
* **Test Suite:** Messaging
* **Test Priority:** High
* **Preconditions:**
    * Two registered users are logged into their accounts.
* **Test Data:** Sample text messages.
* **Test Steps:**
    1. Both users open a private chat.
    2. One user sends a text message.

    3. Observe if the other user receives the message immediately.
* **Postconditions:**
    * The message is sent and received instantaneously.
* **Expected Result:** The receiving user should see the message appear in their inbox immediately upon sending.

* **Severity:** Critical
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** MSG-004
* **Test Case Title:** Verify new message notifications.
* **Test Case Description:** This test case will ensure that users receive notifications when they have new messages.
* **Test Suite:** Messaging
* **Test Priority:** High
* **Preconditions:**
    * Two registered users are logged into their accounts.
* **Test Data:** Sample text message
* **Test Steps:**
    1. One user logs out of the application.
    2. The other user sends a private message to the logged-out user.
    3. The logged-out user logs back into the application.
* **Postconditions:**
    * The user receives a notification about the new message.

* **Expected Result:** Upon logging back in, the user will be notified of the new message received, and the message will be displayed in their inbox.

* **Severity:** Critical
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** MSG-005
* **Test Case Title:** Verify group chat functionality.
* **Test Case Description:** This test case will check the ability to create and participate in group chats.

* **Test Suite:** Messaging
* **Test Priority:** High
* **Preconditions:**
    * Three or more registered users.
* **Test Data:** A group name
* **Test Steps:**
    1. One user initiates a new group chat.
    2. Other users are invited to join the group chat.
    3. Users send messages within the group chat.
* **Postconditions:**
    * The group chat is created successfully.
    * All invited users are members of the group.
* **Expected Result:** The group chat is created, members can join, and messages are displayed in the chat for all participants.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Data - 4

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Content Sharing

Feature description -

Users can share various types of content with their followers or the public, including text posts, photos, videos, links, articles, and documents. Content sharing fosters engagement, creativity, and self-expression among users, allowing them to share their thoughts, experiences, and interests with others.

Userstory title -

As a user, I want to share posts so that I can express my thoughts and interests with others.

Acceptance criteria -

['Users can create and share text posts, photos, and videos.', 'Content can be shared with followers or publicly.', 'Users can edit or delete their posts.', 'Posts should support hashtags and tagging.', 'Users should receive feedback through likes, comments, and shares.']

Generated Test Cases:

Test Cases for Social Media Platform - Content Sharing Feature

Test Case ID: SCS_001

Test Case Title: Verify that users can create and share text posts

Test Case Description: This test case verifies the functionality of creating and sharing text posts on the social media platform. It ensures users can input text content, add relevant information, and successfully share it with their followers or the public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

* User is logged in to the platform.

* User has an existing profile with followers.

* Required access permissions are granted.

Test Data:

* Different text content for posts (short, long, with emojis, etc.)

* Selection of audience (followers or public)

Test Steps:

1. Navigate to the "Create Post" section.

2. Input a text message into the post area.

3. Select the desired audience (followers or public).

4. Click the "Share" button.

5. Verify the post appears on the user's profile and the designated audience's feed.

Postconditions:

* The shared post is visible in the user's profile and the selected audience feed.

* Editing and deleting options are available for the post.

Expected Result: The user successfully creates and shares a text post with the chosen audience.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: SCS_002

Test Case Title: Verify that users can share photos

Test Case Description: This test case verifies the functionality of sharing photos on the social media platform. It ensures users can upload photos, add relevant information, and successfully share them with their followers or the public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

* User is logged in to the platform.

* The user has access to upload photos.

* A valid image file is available for testing.

Test Data:

* Different types of image files (JPEG, PNG, GIF)

* Different image sizes

* Selection of audience (followers or public)

Test Steps:

1. Navigate to the "Create Post" section.

2. Click the "Upload Photo" button.

3. Select a valid image file from the device.

4. Add a caption (optional) and select the desired audience.

5. Click the "Share" button.

6. Verify the photo appears on the user's profile and the designated audience's feed.

Postconditions:

* The shared photo is visible in the user's profile and the selected audience feed.

* Editing and deleting options are available for the post containing the photo.

Expected Result: The user successfully shares a photo with the chosen audience.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: SCS_003

Test Case Title: Verify that users can share videos

Test Case Description: This test case verifies the functionality of sharing videos on the social media platform. It ensures users can upload videos, add relevant information, and successfully share them with their followers or the public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

* User is logged in to the platform.

* The user has access to upload videos.

* A valid video file is available for testing.

Test Data:

* Different video file formats (MP4, AVI, etc.)

* Different video resolutions

* Selection of audience (followers or public)

Test Steps:

1. Navigate to the "Create Post" section.

2. Click the "Upload Video" button.

3. Select a valid video file from the device.

4. Add a caption (optional) and select the desired audience.

5. Click the "Share" button.

6. Verify the video appears on the user's profile and the designated audience's feed.

Postconditions:

* The shared video is visible in the user's profile and the selected audience feed.

* Editing and deleting options are available for the post containing the video.

Expected Result: The user successfully shares a video with the chosen audience.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: SCS_004

Test Case Title: Verify that users can share content publicly and with followers

Test Case Description: This test case verifies the functionality of choosing the audience for content sharing, ensuring users can select to share with either all users (public) or only their followers.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

* User is logged in to the platform.

* The user has a minimum number of followers for testing public vs. follower sharing.

Test Data:

* Text post content

* Photo content

* Video content

Test Steps:

1. Create a text post, photo, or video.

2. Select the "Followers" audience option.

3. Share the content.

4. Verify the content is only visible to the user's followers and not the public.

5. Repeat steps 1-4, this time selecting the "Public" audience option.

6. Share the content.

7. Verify the content is visible to all users on the platform.

Postconditions:

* The content is successfully shared based on the selected audience type.

Expected Result: Users can accurately choose between public and follower audiences for content sharing.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: SCS_005

Test Case Title: Verify that users can delete their posts

Test Case Description: This test case verifies the functionality of deleting posts created by a user, ensuring they can remove their content from the platform.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

* User is logged in to the platform.

* The user has created a post (text, photo, or video).

Test Data:

* Existing post content (text, photo, or video)

Test Steps:

1. Navigate to the user's profile and locate the post to be deleted.

2. Click the "Delete" option for the selected post.

3. Click "Confirm" to finalise the deletion.

4. Verify the post is no longer visible on the user's profile or feed.

Postconditions:

* The deleted post is removed from the platform.

Expected Result: The user successfully deletes their post from the platform.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 5

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Privacy Settings

Feature description -

Privacy settings allow users to control who can view their profile, posts, and other activity on the platform. Users can choose between public, private, or restricted privacy settings, determining who can see their content, send them friend requests, or interact with them.

Userstory title -

As a user, I want to set privacy settings so that I can control who sees my information.

Acceptance criteria -

['Users can set their profile to public, private, or restricted.', 'Privacy settings can be changed at any time.', 'Users can control visibility of individual posts.', 'The system should respect the privacy settings immediately.', 'Users should be notified of any changes in privacy settings.']

Generated Test Cases:

Here are 5 detailed test cases for the provided input data, following the Test Case Creation Template:

Test Case 1:

Test Case ID: PM_PS_001

Test Case Title: Verify that users can set their profile to public

Test Case Description: This test case verifies that users can successfully change their profile privacy setting to public and confirms that their profile is now viewable to all users.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in

Test Data: User profile information

Test Steps:

1. Navigate to the user profile settings.

2. Select the 'Public' option from the privacy settings menu.

3. Save the changes to the privacy settings.

Postconditions:

- User profile privacy is updated to public.

Expected Result: The system successfully updates the user's profile to public visibility, allowing anyone on the platform to view their profile information.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: PM_PS_002

Test Case Title: Verify that users can set their profile to private

Test Case Description: This test case verifies that users can successfully change their profile privacy setting to private and confirms that their profile is only viewable to themselves and their approved connections.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: User profile information

Test Steps:

1. Navigate to the user profile settings.

2. Select the 'Private' option from the privacy settings menu.

3. Save the changes to the privacy settings.

Postconditions:

- User profile privacy is updated to private.

Expected Result: The system successfully updates the user's profile to private visibility, restricting access to only the user themselves and their approved connections.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: PM_PS_003

Test Case Title: Verify that users can set their profile privacy settings to restricted

Test Case Description: This test case verifies that users can successfully change their profile privacy setting to restricted and confirms that they can control which users can view their content.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in

Test Data: User profile information, list of users

Test Steps:

1. Navigate to the user profile settings.
2. Select the 'Restricted' option from the privacy settings menu.
3. Choose specific users or groups who can view the user's profile and content.
4. Save the changes to the privacy settings.

Postconditions:

- User profile privacy is updated to restricted.

Expected Result: The system successfully updates the user's profile to restricted visibility, allowing the user to selectively control which users can view their content and profile information.

Severity: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: PM_PS_004

Test Case Title: Verify user can change privacy settings at any time

Test Case Description: This test case verifies that a user can modify their profile privacy setting (public, private, or restricted) at any time after initial setup.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in
- User initially sets their profile to a specific privacy setting (public, private, or restricted).

Test Data: User profile information

Test Steps:

1. Access the user profile settings.
2. Change the privacy setting to a different option.
3. Save the changes.

Postconditions:

- User profile privacy is updated to the new chosen setting.

Expected Result: The system allows the user to change their privacy setting without encountering errors or restrictions. The user profile visibility updates accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: PM_PS_005

Test Case Title: Verify that users are notified of their privacy setting changes

Test Case Description: This test case verifies that the system successfully notifies users when they modify their profile privacy settings.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: User profile information

Test Steps:

1. Access the user profile settings.
2. Change the privacy setting from one option to another (e.g., public to private).
3. Save the changes.
4. Check the user interface for any notifications related to the privacy setting change.

Postconditions:

- User profile privacy is updated.

Expected Result: The system displays a notification confirming the user of their successfully applied privacy setting change.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 6

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Notifications

Feature description -

Notifications alert users about new activity on the platform, such as likes, comments, mentions, friend requests, and event invitations. Notifications help users stay updated and engaged, prompting them to interact with the platform and respond to interactions from others.

Userstory title -

As a user, I want to receive notifications so that I am aware of new activity related to my profile.

Acceptance criteria -

['Users receive notifications for likes, comments, and shares.', 'Users are notified of new friend requests and mentions.', 'Notifications are displayed in real-time.', 'Users can customize notification preferences.', 'Notifications include a link to the relevant activity.']

Generated Test Cases:

Here are 5 test cases for the provided user story and acceptance criteria:

Test Case 1

Test Case ID: SMPL-NTF-001

Test Case Title: Verify that users receive notifications for likes

Test Case Description: This test case aims to verify that users receive notifications when their posts, comments, or profile pictures receive likes from other users.

Test Suite: Notifications

Test Priority: High

Preconditions:

* User is logged into the platform.

* User has at least one post, comment, or profile picture.

* Another user is logged in and has permission to like the user's content.

Test Data:

Test Post: Sample post with text, image, or video

Test Steps:

1. The first user publishes a test post (text, image, or video).

2. The second user likes the test post.

3. Check if the first user receives a notification for the like on their device (mobile app or web browser).

Postconditions:

* The notification should display the name of the user who liked the post.
* The notification should include a link to the liked post.

Expected Result:

The first user receives a real-time notification indicating they received a like on the test post.

Severity: Major

Type of Testing: Functionality, Usability

Test Case Approach: Positive

Test Case 2

Test Case ID: SMPL-NTF-002

Test Case Title: Verify that users receive notifications for comments

Test Case Description: This test case verifies that users are notified when others comment on their posts or profile pictures.

Test Suite: Notifications

Test Priority: High

Preconditions:

* User is logged into the platform.
* User has published a post or uploaded a profile picture.
* Another user is logged in and has permission to comment.

Test Data:

Test Post: Sample post (text, image, video)

Test Steps:

1. The first user publishes a test post.

2. The second user leaves a comment on the test post.

3. Check if the first user receives a notification on their device for the comment.

Postconditions:

- The notification should display the name of the user who commented.
- The notification should include a link to the commented post.

Expected Result:

The first user receives a real-time notification indicating they received a comment on the test post.

Severity: Major

Type of Testing: Functionality, Usability

Test Case Approach: Positive

Test Case 3

Test Case ID: SMPL-NTF-003

Test Case Title: Verify that users receive notifications for new friend requests

Test Case Description: This test case aims to confirm that users receive notifications when new friend requests are sent to them.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User1 is logged into the platform.
- User2 is logged into the platform.
- User1 and User2 are not currently friends.

Test Data:

No test data needed

Test Steps:

1. User2 sends a friend request to User1.
2. Check if User1 receives a notification for the friend request on their device.

Postconditions:

* The notification should display the name of the user who sent the request.
* The notification should include an option to accept or decline the friend request.

Expected Result:

User1 receives a notification indicating they have a new friend request from User2.

Severity: Major

Type of Testing: Functionality, Usability

Test Case Approach: Positive

Test Case 4

Test Case ID: SMPL-NTF-004

Test Case Title: Verify that users can customize notification preferences

Test Case Description: This test case verifies that users can modify their notification settings to control the types of notifications they receive.

Test Suite: Notifications

Test Priority: Medium

Preconditions:

* User is logged into the platform.

Test Data:

* Different notification categories (likes, comments, mentions, friend requests, etc.)

Test Steps:

1. Navigate to the user's profile settings and locate the notification preferences section.
2. Disable notifications for one or more notification categories.

3. While notifications are disabled for the selected categories, attempt to trigger activities within those categories (like, comment, mention, send friend request).

Postconditions:

* Confirm that notifications are not received for the disabled categories.

Expected Result:

The user's notification settings are saved, and they do not receive notifications for the categories they have disabled.

Severity: Major

Type of Testing: Functionality, Usability

Test Case Approach: Positive

Test Case 5

Test Case ID: SMPL-NTF-005

Test Case Title: Verify that notifications display a link to relevant activity

Test Case Description: This test case confirms that notifications provide a clickable link that directs users to the specific activity being mentioned.

Test Suite: Notifications

Test Priority: High

Preconditions:

* User is logged into the platform.
* User has published a post (text, image, or video)

Test Data:

* Test Post: Sample post with text, image, or video

Test Steps:

1. A second user comments on the test post.

2. Check if the first user receives a notification for the comment.
3. Tap or click on the notification's link.

Postconditions:

* The link should take the user directly to the commented post on the platform.

Expected Result:

The notification link correctly directs the user to the specified comment on the test post.

Data - 7

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Search Functionality

Feature description -

Search functionality enables users to discover content, profiles, and topics of interest on the platform. Users can search for specific keywords, hashtags, accounts, or topics, helping them find relevant content and connect with others who share similar interests.

Userstory title -

As a user, I want to use search functionality to find specific content or profiles on the platform.

Acceptance criteria -

['Users can search for content using keywords or hashtags.', 'Search results include profiles, posts, and topics.', 'Results are displayed in order of relevance.', 'Users can filter search results by content type.', 'The search functionality provides suggestions as the user types.']

Generated Test Cases:

Here are 5 test cases for the Social Media Platform's Search Functionality, following the provided template and ensuring each acceptance criterion is covered:

Test Case 1

Test Case ID: smps001

Test Case Title: Verify that users can search for content using keywords

Test Case Description: This test case verifies that users can input keywords into the search bar and retrieve relevant content, including posts, profiles, and topics.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in to the platform
- Content has been posted on the platform, including posts with relevant keywords, user profiles with relevant keywords, and topics with relevant keywords.

Test Data:

- List of relevant keywords commonly used on the platform

Test Steps:

1. Navigate to the search bar on the platform.
2. Input a keyword from the prepared list.
3. Click the search button.
4. Review the search results displayed.
5. Verify that the results include relevant posts, profiles, and topics containing the entered keyword.

Postconditions:

- Search results are displayed on the screen.

Expected Result: The system displays a list of relevant search results that match the entered keyword, including posts, profiles, and topics.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: smps002

Test Case Title: Verify that users can search for content using hashtags

Test Case Description: This test case examines the functionality of searching for content using hashtags within the platform.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in to the platform.
- Content containing relevant hashtags has been posted on the platform.

Test Data:

- List of relevant hashtags commonly used on the platform.

Test Steps:

1. Navigate to the search bar on the platform.
2. Input a hashtag from the prepared list (e.g., #Travel, #Music).
3. Click the search button.
4. Review the search results displayed.
5. Verify that the results include posts containing the entered hashtag.

Postconditions:

- Search results are displayed on the screen.

Expected Result: The system displays a list of relevant search results that contain the entered hashtag, including posts associated with that hashtag.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: smps003

Test Case Title: Verify that users can filter search results by content type

Test Case Description: This test case verifies the functionality of filtering search results based on different content types (e.g., posts, profiles, topics).

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in to the platform.
- Multiple content types (posts, profiles, topics) with relevant keywords exist.

Test Data:

- Keywords related to different content types.

Test Steps:

1. Navigate to the search bar on the platform.
2. Input a keyword related to a specific content type.
3. Click the search button.
4. Locate the filter options for content type.
5. Select a specific content type from the filter options.
6. Review the filtered search results.

Postconditions:

- Search results are displayed on the screen, filtered by the selected content type.

Expected Result: The system filters the search results to display only the content matching the selected content type (e.g., only posts, only profiles, only topics).

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: smps004

Test Case Title: Verify that search results are displayed in order of relevance

Test Case Description: This test case assesses the relevance ranking of search results to ensure the most pertinent content appears first.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in to the platform.
- Multiple content pieces (posts, profiles, topics) are available related to a chosen keyword.

Test Data:

- List of diverse keywords related to multiple content types.

Test Steps:

1. Navigate to the search bar on the platform.
2. Input a keyword from the prepared list.
3. Click the search button.

4. Examine the first 5-10 search results.
5. Assess the relevance of each result to the entered keyword.

Postconditions:

- Search results are displayed on the screen.

Expected Result: The system displays search results in a descending order of relevance to the entered keyword.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: smps005

Test Case Title: Verify that the search functionality provides suggestions as the user types.

Test Case Description: This test case examines whether the search bar offers real-time suggestions to assist users in finding what they seek.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in to the platform.

Test Data:

- Common search terms or partial keywords used on the platform.

Test Steps:

1. Navigate to the search bar on the platform.
2. Begin typing a common search term or a partial keyword.
3. Observe if the system provides auto-suggestions as the user types.

Postconditions:

- Search suggestions are displayed below the search bar.

Expected Result: The system automatically suggests relevant search terms or partial matches as the user types in the search bar.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 8

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Hashtags and Tagging

Feature description -

Hashtags and tagging allow users to categorize their content and make it more discoverable to others. Users can add relevant hashtags to their posts or tag other users, brands, or locations, increasing the visibility and reach of their content.

Userstory title -

As a user, I want to use hashtags and tagging so that I can categorize my content and reach a wider audience.

Acceptance criteria -

['Users can add hashtags to their posts.', 'Users can tag other profiles, brands, or locations.', 'Hashtags and tags are clickable and lead to related content.', 'Users can search for content using hashtags.', 'The system should suggest popular hashtags as users type.']

Generated Test Cases:

Here are five detailed test cases for the "Hashtags and Tagging" feature, following the provided template:

Test Case 1

Test Case ID: hgt2r8

Test Case Title: Verify that users can add hashtags to their posts

Test Case Description: This test case verifies the functionality of adding hashtags to a post and ensures they are displayed correctly.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in.

- User has access to create posts.

Test Data: Sample hashtags (#socialmedia, #testing, #platform)

Test Steps:

1. Navigate to the post creation interface.

2. Compose a sample post content.

3. Type a hashtag (e.g., #socialmedia) and press Enter.

4. Observe if the hashtag is displayed correctly in the post.

5. Repeat steps 3-4 with different hashtags.

Postconditions:

- The post is saved with the added hashtags.

Expected Result: Hashtags are added to the post as clickable links, with the '#' symbol preceding them.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: tgg7j5

Test Case Title: Verify that users can tag other profiles in their posts

Test Case Description: This test case ensures users can tag other profiles (@username) within their posts and that these tags are correctly linked to the mentioned profiles' pages.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in.

- There are at least two registered users on the platform.

Test Data: Username of a different user on the platform.

Test Steps:

1. Navigate to the post creation interface.
2. Compose a sample post content.
3. Type the "@" symbol followed by the target user's username (e.g., @johndoe)
4. Observe if the username is displayed as a clickable link.
5. Click on the tagged username link.

Postconditions:

- The post is saved with the tag.

Expected Result: The tagged username is displayed as a clickable link leading to the user's profile page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: fvbnr8

Test Case Title: Verify that hashtags and tags are clickable and lead to related content

Test Case Description: This test case confirms that clicking on a hashtag or tag leads to a page displaying content associated with that term.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in.
- Posts containing hashtags and tags exist on the platform.

Test Data: Hashtags and tags from existing posts

Test Steps:

1. Locate a post with hashtags or tags.
2. Click on a hashtag within the post.
3. Observe the content displayed on the resulting page.
4. Click on a tagged username.
5. Observe the content displayed on the resulting profile page.

Postconditions:

- The system does not exhibit errors or crashes.

Expected Result: Clicking on a hashtag displays a page with other posts containing the same hashtag. Clicking on a tagged username leads to that user's profile page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: pqlweru

Test Case Title: Verify that users can search for content using hashtags.

Test Case Description: This test case verifies if users can input hashtags into the search bar and retrieve relevant posts and user profiles.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in.
- Posts containing specific hashtags are published.

Test Data: Sample hashtags related to trends or popular topics.

Test Steps:

1. Access the platform's search bar.
2. Enter a sample hashtag (e.g., #travel).
3. Observe the search results displayed.
4. Ensure the results contain posts and profiles relevant to the entered hashtag.

Postconditions:

- The search results are displayed without errors.

Expected Result: The search returns relevant posts and user profiles that have used the specified hashtag.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: djn8we

Test Case Title: Verify that the system suggests popular hashtags as users type.

Test Case Description: This test case evaluates the functionality of hashtag suggestions while a user is typing in the post creation interface.

Test Suite: Hashtags and Tagging

Test Priority: Medium

Preconditions:

- User is logged in.
- The platform has an active database of popular hashtags.

Test Data: Starting characters of common hashtags.

Test Steps:

1. Navigate to the post creation interface.

2. Begin typing characters relevant to a popular hashtag (e.g., "#f").

3. Observe if the system suggests popular hashtags containing those characters.

4. Select a suggested hashtag.

5. Verify that the suggested hashtag is correctly displayed in the post.

Postconditions:

- User interface remains stable during suggestion display

Expected Result: The system provides a list of relevant hashtags as suggestions, which the user can select to add to their post.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 9

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Emojis and Reactions

Feature description -

Emojis and reactions provide users with expressive tools to react to posts and messages. Users can use emojis to convey emotions such as joy, sadness, love, or excitement, enhancing communication and engagement on the platform.

Userstory title -

As a user, I want to use emojis and reactions so that I can express my feelings about posts and messages.

Acceptance criteria -

['Users can react to posts with a variety of emojis.', 'Emojis can be used in comments and messages.', 'Users can see a count of reactions on posts.', 'The platform should support a wide range of emojis.', 'Users can remove or change their reactions.']

Generated Test Cases:

Here are 5 test cases for the "Emojis and Reactions" feature, following the provided template:

Test Case ID: SM-Emoji1

Test Case Title: Verify that users can react to posts with a variety of emojis

Test Case Description: This test case verifies the ability of users to react to posts using multiple emojis provided by the platform.

Test Suite: User Interaction

Test Priority: High

Preconditions:

  * User is logged into the platform.
  * A post exists on the user's news feed.

Test Data: A list of diverse emojis provided by the platform (happy, sad, angry, laughing, heart, thumbs-up, etc.)

Test Steps:

1. Navigate to the post on the user's news feed.

2. Click on the emoji reaction button below the post.

3. Select various emojis from the available options.

4. Verify that the selected emoji is displayed as a reaction to the post.

5. Observe the reaction count next to the emoji increments with each added reaction.

Postconditions:

* User can easily switch between different emojis to react to the post.

Expected Result: The system successfully allows users to react to posts with a variety of emojis, and the reaction count accurately reflects the number of reactions.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SM-Emoji2

Test Case Title: Verify that emojis can be used in comments and messages

Test Case Description: This test case validates the functionality of using emojis within comments on posts and direct messages.

Test Suite: User Interaction

Test Priority: Medium

Preconditions:

  * User is logged into the platform.

  * A post exists on the user's news feed or a user to message is found.

Test Data:

* A message or comment to be tested

* Various emojis provided by the platform

Test Steps:

1. Navigate to a post or open a chat window to send a direct message.

2. Compose a message or comment containing one or more emojis.

3. Send the message.

Postconditions:

* Ensure the sent message is displayed correctly.

Expected Result: The emojis are displayed correctly within the comment or message, ensuring proper communication through expressive emojis.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SM-Emoji3

Test Case Title: Verify that users can see a count of reactions on posts

Test Case Description: This test case verifies that the platform accurately displays the number of reactions a post has received.

Test Suite: Post Functionality

Test Priority: High

Preconditions:

  * User is logged into the platform.

  * At least one post exists on the user's news feed.

Test Data:

* A post with reactions.

Test Steps:

1. Navigate to a post on the user's news feed with reactions.

2. Observe the reaction count displayed next to each emoji.

3. Share the post with others. (Optional to increase reaction count for testing)

Postconditions:

* React to the post with several emojis.

Expected Result: The reaction counts on the post accurately reflect the number of reactions it received, updating dynamically as users interact with the post.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SM-Emoji4

Test Case Title: Verify that the platform supports a wide range of emojis

Test Case Description: This test case validates that the social media platform supports a diverse and comprehensive set of emojis.

Test Suite: User Interaction

Test Priority: Medium

Preconditions: User is logged into the platform.

Test Data: A comprehensive list of commonly used emojis from various categories (emotions, objects, symbols, flags, etc.).

Test Steps:

1. Navigate to a post or a chat window.
2. Attempt to use each emoji from the provided list in a comment or reaction.

Postconditions:

* User is able to view and select all emojis from the list.

Expected Result: The platform successfully displays and allows users to select a wide range of emojis without encountering any errors or limitations.

Severity: Important

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SM-Emoji5

Test Case Title: Verify that users can remove or change their reactions

Test Case Description: This test case examines the functionality for users to modify or remove their previous reactions to posts.

Test Suite: User Interaction

Test Priority: Medium

Preconditions:

    * User is logged into the platform.

    * User has reacted to a post with an emoji (e.g., a like).

Test Data: The same post that the user reacted to

Test Steps:

1. Navigate to the post where the user reacted.

2. Locate the user's previously added reaction (e.g., the like emoji).

3. Click on the already added reaction.

4. Select the option to remove the reaction.

Postconditions:

* Verify the reaction count reflects the removal.

Expected Result: The system successfully allows users to remove or change their reactions to posts, updating the reaction count accordingly.

Data - 10

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Live Streaming

Feature description -

Live streaming enables users to broadcast real-time video content to their followers or the public. Users can share live events, performances, Q&A sessions, tutorials, and behind-the-scenes footage, allowing for authentic and interactive experiences with their audience.

Userstory title -

As a user, I want to broadcast live video so that I can share real-time events with my followers.

Acceptance criteria -

['Users can start and stop live streams.', 'Live streams can be public or limited to followers.', 'Viewers can interact with the streamer through comments and reactions.', 'The stream should display viewer count and live comments.', 'Users can save the live stream for later viewing.']

Generated Test Cases:

Here are 5 test cases for the provided Live Streaming feature:

Test Case 1: Start & Stop a Live Stream

Test Case ID: LIV-001

Test Case Title: Verify that users can start and stop live streams

Test Case Description: This test case verifies the ability of a user to initiate and terminate a live streaming session. It focuses on ensuring the workflow is seamless and the expected actions occur when starting and stopping the stream.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User has a valid internet connection

Test Data:

- No Test Data Needed

Test Steps:

1. Navigate to the live streaming feature within the social media platform.
2. Click the "Start Live Stream" button.
3. Monitor the platform for confirmation of the live stream initiation. Verify that the stream starts broadcasting and the user interface reflects the live status.
4. Broadcast test content for a short period (e.g., 30 seconds).
5. Click the "Stop Live Stream" button.
6. Verify that the stream terminates successfully and the user interface updates accordingly.

Postconditions:

- Live stream is stopped
- User interface reflects the stream is no longer live

Expected Result: The user successfully initiates and stops a live stream without encountering errors or unexpected behavior.

Severity: Critical

Type of Testing: Functional testing

Test Case Approach: Positive

Test Case 2: Public vs. Follower-Only Live Streams

Test Case ID: LIV-002

Test Case Title: Verify that live streams can be set to public or follower-only

Test Case Description: This test case ensures the functionality of choosing the viewership scope for a live stream ? either publicly accessible or restricted to the streamer's followers.

Test Suite: Live Streaming

Test Priority: Medium

Preconditions:

- User is logged in
- Creator has followers

Test Data:

- No Test Data Needed

Test Steps:

1. Navigate to the live streaming feature.
2. Start a new live stream.

3. Access the live stream settings.
4. Select "Public" for the viewership scope.
5. Initiate the live stream.
6. Verify that the stream is visible to all users on the platform.
7. Re-enter the live stream settings.
8. Select "Followers Only" for the viewership scope.
9. Initiate the live stream again.
10. Attempt to view the stream from a non-follower account.

Postconditions:

- Live stream settings are updated accordingly

Expected Result:

- Public streams are visible to all users.
- Follower-only streams are only accessible to users who follow the creator. Non-followers should not be able to view these streams.

Severity: Major

Type of Testing: Functional testing

Test Case Approach: Positive

Test Case 3: Viewer Interaction

Test Case ID: LIV-003

Test Case Title: Verify that viewers can interact with the streamer through comments and reactions.

Test Case Description: This test case determines if viewers can leave comments and utilize reactions (like emojis) to engage with the live streamer and their content.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- A live stream is running

Test Data:

- Sample comments to be left (e.g., "Great stream!", "Keep up the good work!")
- Assorted reactions (e.g., ?, ?, ?)

Test Steps:

1. Join an active live stream.
2. Enter a sample comment in the designated comment section and submit it.

3. Verify that the comment appears on the live stream for all viewers.

4. Utilize different reactions to the live stream content.

5. Verify that the reactions are displayed visibly to the streamer and other viewers.

Postconditions:

- Comments and reactions are applied to live stream.

Expected Result:

Viewers are able to leave comments and utilize reactions without encountering errors or limitations. The streamer and other viewers should see these interactions in real-time.

Severity: Major

Type of Testing: Functional testing

Test Case Approach: Positive

Test Case 4: Stream View Count & Comment Visibility

Test Case ID: LIV-004

Test Case Title: Verify that the stream displays viewer count and live comments

Test Case Description: This test case ensures the platform effectively displays the number of viewers watching a live stream and that the live comments are presented as expected.

Test Suite: Live Streaming

Test Priority: Medium

Preconditions:

- At least two user accounts (one to start the stream, one to view)

Test Data: No Test Data Needed

Test Steps:

1. Start a live stream using one user account.
2. Navigate to the live stream as a second user.
3. Observe the display of the viewer count. Verify that it accurately reflects the number of current viewers (including the streamer).
4. Leave a comment in the live stream as the second user.
5. Verify that the comment appears in the live stream chat within your view.

Postconditions: The viewer count reflects the live stream activity.

Expected Result:

- The viewer count updates dynamically as viewers join or leave.

- Comments are displayed promptly in the chat, maintaining chronological order.

Severity: Major

Type of Testing: Functional testing

Test Case Approach: Positive

Test Case 5: Saving Live Streams

Test Case ID: LIV-005

Test Case Title: Verify that users can save live streams for later viewing.

Test Case Description: This test case checks if users can successfully save a live stream broadcast for playback at a later time.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- A live stream has completed broadcasting.

Test Data:

- No Test Data Needed

Test Steps:

1. Access the live stream archive.
2. Locate a recently completed live stream.
3. Select the "Save" or "Download" option for the chosen live stream.
4. Verify that the live stream is saved in a user-accessible location (e.g., downloads folder or within the platform's content library).

Postconditions:

- Saved stream is accessible

Expected Result: The specified live stream is successfully saved and can be retrieved by the user for later viewing.

Severity: Major

Type of Testing: Functional testing

Test Case Approach: Positive

Data - 11

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Stories

Feature description -

Stories are temporary posts that disappear after a certain period, typically 24 hours. Users can share ephemeral content such as photos, videos, and text overlays in their stories, providing a more casual and spontaneous way to share moments and updates with their followers.

Userstory title -

As a user, I want to share stories so that I can share temporary moments with my followers.

Acceptance criteria -

['Users can create stories with photos, videos, and text overlays.', 'Stories disappear after 24 hours.', 'Users can see who viewed their stories.', 'Stories can be shared publicly or with selected followers.', 'Users can add stickers and filters to their stories.']

Generated Test Cases:

Here are 5 test cases for the provided "Stories" feature:

Test Case ID: STS_001

Test Case Title: Verify that users can create stories with photos, videos, and text overlays.

Test Case Description: This test case verifies the user's ability to create stories using various media types: photos, videos, and text overlays.

Test Suite: Stories Feature

Test Priority: High

Preconditions:

- User is logged in.

Test Data:

- A photo from the user's device

- A short video clip from the user's device

- A predefined text string for the overlay

Test Steps:

1. Navigate to the "Stories" creation section.

2. Select the "Photo" option and choose a photo from the device.

3. Add a text overlay using the provided text tool.

4. Select the "Video" option and choose a video clip from the device.

5. Evaluate if the photo, video, and text overlay are displayed correctly within the story.

Postconditions:

- User has successfully added a story with photo, video, and text.

Expected Result: The system successfully creates and displays the story with the selected photo, video, and text overlay.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: STS_002

Test Case Title: Verify that stories disappear after 24 hours

Test Case Description: This test case verifies the automatic deletion of stories after the defined timeframe (24 hours).

Test Suite: Stories Feature

Test Priority: High

Preconditions:

- A user publishes a story.
- Time is allowed to pass (at least 24 hours).

Test Data: No test data needed.

Test Steps:

1. Access the user's story.
2.
3. Check if the story is no longer visible or accessible.

Postconditions:

- The story is no longer present on the timeline.

Expected Result: After 24 hours, the story is automatically deleted and is not visible to viewers.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: STS_003

Test Case Title: Verify that users can see who viewed their stories

Test Case Description: This test case verifies user visibility regarding those who have viewed their stories.

Test Suite: Stories Feature

Test Priority: Medium

Preconditions:

- User publishes a story.
- Another user logs in and views the story.

Test Data: No test data needed.

Test Steps:

1. User who published the story accesses their story view history.
2. Verify the list of viewers includes the user information from step 2.

Postconditions:

- Story view history is accessible.

Expected Result: The user sees a list of usernames or profiles who have viewed their story.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: STS_004

Test Case Title: Verify private story sharing options

Test Case Description: This test case verifies the functionality of allowing users to share stories with specific followers rather than publicly.

Test Suite: Stories Feature

Test Priority: High

Preconditions:

- User creates an account and follows other users.

Test Data: Predefined list of follower accounts.

Test Steps:

1. User publishes a story with private sharing enabled.
2. User attempts to view the story (from another account which isn't a follower)
3. Verify the story is not visible to the non-follower users.
4. Verify the story is visible to the listed follower accounts.

Postconditions:

- Story privacy settings are applied.

Expected Result: Private stories are only viewable by users who the original poster has explicitly allowed to view them. Non-followers cannot access the stories.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive/Negative

Test Case ID: STS_005

Test Case Title: Verify story sticker and filter functionality

Test Case Description: This test case confirms the availability and usability of stickers and filters for enriching story content.

Test Suite: Stories Feature

Test Priority: Medium

Preconditions:

- User is logged in and has a story ready to be shared.

Test Data: Predefined stickers and filters provided within the platform

Test Steps:

1. Go to the story creation section.
2. Select a suitable photo or video for the story.
3. Access the stickers and filter options.
4. Choose a variety of stickers and apply them to the story.
5. Experiment with different filter options and apply them to the story visually.

Postconditions:

- Story includes applied stickers and filters.

Expected Result: The platform allows users to add stickers and apply filters to their stories, enhancing their visual appeal and providing creative expression.

--

Data - 12

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Analytics and Insights

Feature description -

Analytics and insights provide users with data and metrics about their activity, audience engagement, and content performance on the platform. Users can access information such as follower demographics, post reach, engagement metrics, and audience growth trends, helping them optimize their presence and strategy on the platform.

Userstory title -

As a user, I want to access analytics and insights so that I can understand my audience and optimize my content.

Acceptance criteria -

['Users can view data on follower demographics.', 'Post reach and engagement metrics are available.', 'Users can track audience growth trends.', 'Analytics are displayed in a user-friendly format.', 'Data can be filtered by time range and content type.']

Generated Test Cases:

Here are 5 test cases for the provided input data, following the template:

Test Case 1

Test Case ID: sM7g6b

Test Case Title: Verify that users can view data on follower demographics.

Test Case Description: This test case verifies that the system displays demographic information about a user's followers, such as age, gender, and location.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

* User has a minimum of 100 followers.

* User is logged into the platform.

Test Data: Sample user with followers from various demographics.

Test Steps:

1. Navigate to the user's analytics dashboard.

2. Locate the "Follower Demographics" section.

3. Verify that data is displayed for age, gender, and location.

4. Ensure the data is presented in a clear and readable format (e.g., charts, graphs, tables).

Postconditions: No specific postconditions.

Expected Result: The system successfully displays demographic data about the user's followers.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: v3R4c2

Test Case Title: Verify that post reach and engagement metrics are available.

Test Case Description: This test case verifies the accuracy and availability of metrics related to the reach and engagement of a user's posts.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

* User has published at least three posts.

* User is logged into the platform.

Test Data: Sample posts with differing content types and publishing times.

Test Steps:

1. Go to the user's analytics dashboard.

2. Locate the "Post Performance" section.

3. Verify that metrics such as reach (impressions, unique users), likes, comments, and shares are displayed for each post.

4. Check if the metrics are sorted by date or post type.

Postconditions: User is returned to their analytics dashboard.

Expected Result: The system accurately displays reach and engagement metrics for each post.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: pL9d7e

Test Case Title: Verify that users can track audience growth trends.

Test Case Description: This test case confirms the functionality for tracking the growth of a user's audience over time.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

* User has been active on the platform for at least 30 days.

* User is logged into the platform.

Test Data: No specific test data required.

Test Steps:

1. Navigate to the user's analytics dashboard.

2. Locate the "Audience Growth" section.

3. Verify that a visual representation (e.g., line graph) shows the follower count trend over time (e.g., daily, weekly, monthly).

Postconditions: User is returned to their analytics dashboard.

Expected Result: The system accurately displays a visual representation of audience growth trends.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: x2j5b0

Test Case Title: Verify that analytics are displayed in a user-friendly format.

Test Case Description: This test case checks if the analytics data is presented in a clear, concise, and easy-to-understand manner.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

* User is logged into the platform.

Test Data: Various types of analytics data (e.g., demographics, post performance, audience growth).

Test Steps:

1. Access the user's analytics dashboard.
2. Review the layout and organization of the data.
3. Evaluate the use of visual aids (e.g., charts, graphs) and their clarity.
4. Assess the readability of the text and the overall user experience.

Postconditions: User remains on the analytics dashboard.

Expected Result: The analytics data is presented in a user-friendly format, using clear visuals and organized by category.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: g74h3y

Test Case Title: Verify data filtering capabilities for time range and content type.

Test Case Description: This test case ensures the system allows filtering analytics data by specific time periods and types of content.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

* User has published multiple posts of different types (e.g., text, image, video).

* User is logged into the platform.

Test Data: Sample posts published within varying timeframes.

Test Steps:

1. Go to the user's analytics dashboard.
2. Identify the data filtering options for time range (e.g., day, week, month, custom range) and content type.
3. Select a specific time range (e.g., last week).
4. Filter the data by a specific content type (e.g., video posts).
5. Verify that the displayed analytics accurately reflect the filtered data.

Postconditions: User remains on the analytics dashboard.

Expected Result: The system successfully filters analytics data based on selected time ranges and content types.

Data - 13

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Events and Groups

Feature description -

Events and groups enable users to organize and participate in events, communities, or interest-based groups on the platform. Users can create events, RSVP to invitations, join groups, and participate in discussions, fostering community engagement and connection around shared interests.

Userstory title -

As a user, I want to join events and groups so that I can connect with people who share my interests.

Acceptance criteria -

['Users can create and join events.', 'Users can join interest-based groups.', 'Group discussions can be organized by topic.', 'Event details include date, time, location, and description.', 'Users can RSVP to events and see the list of attendees.']

Generated Test Cases:

Test Cases for Events and Groups Feature

1. Test Case ID: SG_TC01

Test Case Title: Verify that users can create and join events.

Test Case Description: This test case ensures that users can successfully create new events with required details and join existing events.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* A registered user is logged into the platform.

Test Data:

* Event details (title, date, time, location, description)

Test Steps:

1. Navigate to the "Events" section on the platform.
2. Click on the "Create Event" button.
3. Fill in the event details (title, date, time, location, description).
4. Save the new event.
5. Locate the created event on the events list.
6. Click on the "Join Event" button to join the event.
7. Verify that the user is listed as a participant in the event.

Postconditions:

* A new event is successfully created.
* The user is successfully joined to the event.

Expected Result: The user should be able to create an event and join it successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

2. Test Case ID: SG_TC02
Test Case Title: Verify that users can create groups based on interests.

Test Case Description: This test case confirms that users can establish groups centered around shared interests.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* A registered user is logged into the platform.

Test Data:

* Group name and description

Test Steps:

1. Navigate to the "Groups" section of the platform.
2. Click on the "Create Group" button.
3. Enter a unique group name and a descriptive summary.
4. Select relevant categories or tags that align with the group's interest.
5. Set visibility options (public or private).
6. Submit the new group.

Postconditions:

* A new group is created.
* The group is visible in the designated sections (public or private).

Expected Result: The user should be able to create a new group with the specified name, description, and visibility.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

3. Test Case ID: SG_TC03

Test Case Title: Verify that group discussions are organized by topic.

Test Case Description: This test case checks if the platform effectively facilitates topic-based discussions within groups.

Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

* A group has been successfully created (as per Test Case 2).

Test Data:

* Sample discussion topics

Test Steps:

1. Join the created group.
2. Access the group's discussion forum.
3. Start a new discussion thread with a clear and concise topic.
4. Add additional related discussions threads with distinct topics.
5. Observe the organization of discussion threads within the forum.

Postconditions:

* Discussions are organized into separate threads.
* Each thread has a distinct topic.

Expected Result: The platform should allow for clear and organized topic-based discussions within each group.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

4. Test Case ID: SG_TC04
Test Case Title: Verify that event details include date, time, location, and description.

Test Case Description: This test case ensures that all essential details are included when creating an event.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* A registered user is logged into the platform.

Test Data:

* Event details (Title, date, time, location, description)

Test Steps:

1. Navigate to the "Events" section on the platform.

2. Click on the "Create Event" button.

3. Fill in the event details: title, date, time, location, and description.

4. Save the new event.

5. Click on the created event to view its details.

6. Verify if all provided details (date, time, location, description) are displayed accurately.

Postconditions:

* The event details are saved.

* The event is displayed with all required details.

Expected Result: The event details should be complete and accurately displayed on the event page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

5. Test Case ID: SG_TC05

Test Case Title: Verify user RSVP functionality and attendee list for events.

Test Case Description: This test case validates the RSVP functionality and attendee list display for an event.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* An event has been successfully created (as per Test Case 4).

Test Data:

* User accounts

Test Steps:

1. Log in as a user with a registered account.
2. Navigate to the event page.
3. Click the "RSVP" button to attend the event.
4. Click the " attendee list" button to view the list of attendees.
5. Verify that the user's name appears on the attendee list.

Postconditions:

* The user is successfully RSVPed to the event.
* The RSVPD attendees are correctly displayed.

Expected Result: Users should be able to RSVP to events, and their attendance should be reflected in the event's attendee list.

Data - 14

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Monetization Options

Feature description -

Monetization options allow users to earn money through various means on the platform. This may include advertising revenue, sponsored content partnerships, affiliate marketing programs, subscriptions, virtual gifts, or donations from followers, providing creators and influencers with opportunities to monetize their content and audience.

Userstory title -

As a content creator, I want to access monetization options so that I can earn revenue from my content.

Acceptance criteria -

['Users can apply for monetization options.', 'Monetization includes advertising revenue and sponsored content.', 'Users can receive payments through the platform.', 'Detailed earnings reports are available.', 'Monetization eligibility requirements are clearly stated.']

Generated Test Cases:

Here are 5 test cases for the provided input data, following the template you've specified:

Test Case 1

* **Test Case ID:** SOCMNT-001
* **Test Case Title:** Verify that users can apply for monetization options.
* **Test Case Description:** This test case verifies the functionality for users to initiate the monetization application process.
* **Test Suite:** Monetization Options
* **Test Priority:** High
* **Preconditions:**
    * A registered user account exists.
    * User has reached a minimum account age and followers threshold (as defined in platform rules).
* **Test Data:** User profile information, sample content
* **Test Steps:**
    1. Log in as a registered user who meets the eligibility criteria.
    2. Navigate to the user profile settings or dedicated monetization section.
    3. Locate the application button or link for monetization options.
    4. Click on the button/link to initiate the application process.
    5. Fill in the required information (e.g., payment details, content categories).
    6. Submit the monetization application.
* **Postconditions:**
    * Application confirmation message or notification displayed.

    * Application status is visible in user profile settings.
* **Expected Result:** The user successfully submits a monetization application, and the system provides confirmation and updates their profile accordingly.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** SOCMNT-002
* **Test Case Title:** Verify that monetization includes advertising revenue.
* **Test Case Description:** This test confirms that the platform allows users to earn revenue through displayed ads on their content.
* **Test Suite:** Monetization Options
* **Test Priority:** High
* **Preconditions:**
    * A user has been approved for monetization.
    * A piece of user content (e.g., video, article) is published with ad display enabled.
* **Test Data:**
     - User content with ad spaces
* **Test Steps:**
    1. Log in as a monetized user.
    2. Access the published user content.
    3. Observe the presence of advertisement displays within the content.
* **Postconditions:**
    * Ad displays are correctly integrated.
* **Expected Result:** Advertisements are shown within the user's content, signifying the inclusion of

advertising revenue as a monetization option.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** SOCMNT-003

* **Test Case Title:** Verify that users can receive payments through the platform.

* **Test Case Description:** This test case confirms the functionality for transferring earned revenue to the user's designated account.

* **Test Suite:** Monetization Options

* **Test Priority:** High

* **Preconditions:**

    * A user account is monetized.

    * The user has accumulated a minimum amount of earnings.

* **Test Data:** Sample payment information

* **Test Steps:**

    1. Log in as a monetized user.

    2. Access the earnings dashboard or payment section.

    3. Review the available earnings balance.

    4. Select the withdrawal option.

    5. Enter the required payment details.

    6. Initiate the payment transfer.

* **Postconditions:**

    * Payment request is submitted.

    * A notification confirms the payment request.

* **Expected Result:** The system successfully processes the payment request, allowing the user to withdraw their earned revenue to their designated account.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** SOCMNT-004
* **Test Case Title:** Verify that detailed earnings reports are available.
* **Test Case Description:** This test confirms the platform provides comprehensive reports on user earnings.
* **Test Suite:** Monetization Options
* **Test Priority:** Medium
* **Preconditions:**
    * A user account is monetized.
    * Some time has passed during which the user has generated earnings (e.g., a week or month).
* **Test Data:** None
* **Test Steps:**
    1. Log in as a monetized user.
    2. Navigate to the earnings dashboard or reporting section.
    3. Access the detailed earnings report for a selected period.
* **Postconditions:** Report data is available.
* **Expected Result:** The platform displays a detailed earnings report for the chosen period, including information about ad revenue, sponsored content earnings, or other relevant revenue streams, along with breakdowns by date, content type, etc.
* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** SOCMNT-005

* **Test Case Title:** Verify that monetization eligibility requirements are clearly stated.

* **Test Case Description:** This test checks the clarity and accessibility of the rules and criteria for becoming eligible for monetization.

* **Test Suite:** Monetization Options

* **Test Priority:** Medium

* **Preconditions:**

  * A user is not yet monetized.

* **Test Data:** None

* **Test Steps:**

   1. Access the platform's help center, FAQ section, or account settings.

   2. Locate information about monetization requirements.

   3. Review the stated eligibility criteria and policies.

* **Postconditions:**

   * Information is accessible.

* **Expected Result:** The platform clearly outlines the monetization eligibility requirements in an understandable and easily accessible manner, covering factors like age, account status, content quality, and community guidelines.

* **Severity:** Major

* **Type of Testing:** Functional, Usability

* **Test Case Approach:** Usability

Data - 15

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Explore or Discover Tab

Feature description -

The explore or discover tab curates trending or popular content from across the platform, helping users discover new accounts, topics, and trends. The tab may feature personalized recommendations based on users' interests, engagement history, and social connections, facilitating content discovery and exploration.

Userstory title -

As a user, I want to explore trending content so that I can discover new and popular posts, accounts, and topics.

Acceptance criteria -

['The explore tab shows trending and popular content.', 'Content recommendations are personalized based on user interests.', 'Users can filter content by category or topic.', 'The tab updates regularly with new content.', 'Users can save or follow content from the explore tab.']

Generated Test Cases:

Social Media Platform Test Cases: Explore Tab

Here are 5 test cases for the "Explore Tab" feature, following your provided template and covering all acceptance criteria:

Test Case ID: explore_01

Test Case Title: Verify that the explore tab shows trending and popular content.

Test Case Description: This test case verifies that the explore tab displays trending and popular content from across the platform, including posts, accounts, and topics.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- User is logged into the platform.
- Platform is functioning and connected to the internet.

Test Data: No specific test data needed.

Test Steps:

1. Navigate to the explore tab.
2. Observe the displayed content.
3. Identify the source of the content (posts, accounts, topics).
4. Compare the presented content to known trending topics and popular accounts.

Postconditions:

- User remains on the explore tab.

Expected Result: The explore tab showcases trending and popular content from various categories, including recent viral posts, accounts with high engagement, and currently buzzing topics.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: explore_02
Test Case Title: Verify that content recommendations are personalized based on user interests.
Test Case Description: This test case aims to assess if the explore tab provides personalized recommendations aligned with a user's interests, based on their activity and interactions on the platform.

Test Suite: Explore Tab
Test Priority: High
Preconditions:

- User is logged in and has interacted with various content categories (posts, accounts, topics) to establish a profile.
- Platform has collected and analyzed user interaction data.

Test Data: User interaction history, including liked posts, followed accounts, and engaged topics.

Test Steps:

1. Navigate to the explore tab.
2. Observe the recommended content displayed.
3. Compare the recommended content to the user's past interactions and established interests.

Postconditions:

- User remains on the explore tab.

Expected Result: The explore tab recommends content relevant to the user's interests, showcasing posts, accounts, and topics they are likely to engage with based on their past activity.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: explore_03

Test Case Title: Verify that users can filter content by category or topic.

Test Case Description: This test case examines the functionality of content filtering on the explore tab, allowing users to refine their content discovery based on specific categories or topics.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged in.
- Platform has designated content categories and topics.

Test Data: Existing categories and topics on the platform.

Test Steps:

1. Navigate to the explore tab.
2. Identify the available filtering options (categories, topics).
3. Select different filter categories or topics.
4. Observe the filtered content displayed.

Postconditions:

- User remains on the explore tab.

Expected Result: Applying filters refine the content displayed on the explore tab, showcasing content specifically within the chosen category or topic.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: explore_04
Test Case Title: Verify that the explore tab updates regularly with new content.
Test Case Description: This test case checks the frequency of content updates on the explore tab, ensuring users always see fresh and relevant trending content.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged in.
- The platform's content stream is active and generating new content.

Test Data: Timestamps

Test Steps:

1. Visit the explore tab at a specific timestamp.
2. Observe the content displayed.
3. Wait for a predetermined amount of time (e.g., 30 minutes).
4. Return to the explore tab and observe if new content has been added.

Postconditions:

- User remains on the explore tab.

Expected Result: Upon returning to the explore tab after a designated interval, new content should be visible, demonstrating regular updates to the trending content feed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: explore_05

Test Case Title: Verify that users can save or follow content from the explore tab.

Test Case Description: This test case examines the user's ability to save or follow content found on the explore tab, allowing them to easily revisit or stay updated on discovered content.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged in.
- Explore tab displays different content types (posts, accounts, topics).

Test Data: Selected content from explore tab (posts, accounts, topics).

Test Steps:

1. Open the explore tab and navigate to the desired content (posts, accounts, topics).
2. Identify and interact with the "save" or "follow" button for the selected content.

Postconditions:

- User remains on the explore tab

Expected Result: Clicking the "save" or "follow" button successfully saves or subscribes to the selected content, allowing the user to access it later or receive notifications for future updates.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 16

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Accessibility Features

Feature description -

Accessibility features ensure that the platform is inclusive and accessible to users with disabilities. This may include features such as alt text for images, screen reader compatibility, keyboard shortcuts, color contrast options, and adjustable font sizes, enhancing usability and accessibility for all users.

Userstory title -

As a user with disabilities, I want accessible features so that I can use the platform comfortably.

Acceptance criteria -

['The platform provides alt text for images.', 'Screen reader compatibility is available.', 'Keyboard shortcuts can be used for navigation.', 'Color contrast and font size options are provided.', 'Accessibility features are clearly documented.']

Generated Test Cases:

Here are five test cases for the provided User Story, adhering to the Test Case Creation Template and addressing all acceptance criteria:

Test Case 1

* **Test Case ID:** AC_AFP1
* **Test Case Title:** Verify that the platform provides alt text for images.
* **Test Case Description:** This test case will verify that all images on the social media platform have descriptive alt text attributes.
* **Test Suite:** Accessibility Features
* **Test Priority:** High
* **Preconditions:**
  * User is logged in
* **Test Data:** Images with varying content (e.g., photos of people, landscapes, logos)
* **Test Steps:**
  1. Navigate to the platform's homepage.
  2. Examine multiple images displayed on the page.
  3. Right-click on each image and select "Inspect" or "View Page Source."
  4. Locate the `<img>` tag for each image and verify the presence of the "alt" attribute.
  5. Ensure the alt text accurately and concisely describes the image content.
* **Postconditions:**
  * The presence and accuracy of alt text are assessed
* **Expected Result:** All images have alt text attributes with descriptive content that accurately represents the image.

* **Severity:** Major
* **Type of Testing:** Functional, Usability
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** AC_AS1
* **Test Case Title:** Verify screen reader compatibility.
* **Test Case Description:** This test simulates how a screen reader interacts with the platform to ensure navigability and comprehension of content.
* **Test Suite:** Accessibility Features
* **Test Priority:** High
* **Preconditions:**
   * User is navigating using a screen reader software (e.g., NVDA, JAWS).
* **Test Data:** N/A
* **Test Steps:**
   1. Start the screen reader and navigate to the platform's homepage.
   2. Verify that screen reader announces page titles, headings, and navigation elements correctly.
   3. Explore various sections of the platform (e.g., news feed, profile, settings) and listen for accurate content descriptions.
   4. Attempt to perform basic actions (e.g., posting, commenting, liking) using keyboard commands and screen reader navigation.
* **Postconditions:**
   * Screen reader interaction and functionality are evaluated.
* **Expected Result:** The screen reader accurately interprets and announces all website content, providing a navigable and understandable experience for visually impaired users.
* **Severity:** Critical

* **Type of Testing:** Functional, Accessibility
* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** AC_KS1
* **Test Case Title:** Verify keyboard shortcut functionality.
* **Test Case Description:** This test case checks the usability of keyboard shortcuts for key platform actions.
* **Test Suite:** Accessibility Features
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged in
    * Keyboard is the primary input device.
* **Test Data:** N/A
* **Test Steps:**
    1. Open the platform and ensure focus is on the main content area.
    2. Press common keyboard shortcuts for actions like navigating menus (e.g., Tab), posting (e.g., Enter), liking (e.g., Spacebar), and commenting.
    3. Observe the response of the platform to each shortcut.
* **Postconditions:**
    * Keyboard shortcut functionality is evaluated.
* **Expected Result:** Keyboard shortcuts function as expected, providing a seamless and efficient way to navigate and interact with the platform without relying on a mouse.
* **Severity:** Major
* **Type of Testing:** Functional, Usability
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** AC_CC1

* **Test Case Title:** Verify color contrast and font size options.

* **Test Case Description:** This test checks the platform's accessibility by ensuring sufficient color contrast and providing options to adjust font sizes.

* **Test Suite:** Accessibility Features

* **Test Priority:** Medium

* **Preconditions:**

  * User is logged in

* **Test Data:** Black, white, and other contrasting colors; various font sizes.

* **Test Steps:**

  1. Navigate to the platform's settings or accessibility options.

  2. Adjust the color theme and contrast settings to see how text and background colors appear.

  3. Measure the contrast ratios between text and background colors using an accessibility tool or online checker to ensure compliance with WCAG guidelines.

  4. Adjust font sizes using the available options.

  5. Observe the readability of text at different font sizes.

* **Postconditions:**

  * Color contrast and font size adjustments are assessed.

* **Expected Result:** The platform offers an accessible color theme with sufficient contrast for users with visual impairments. Adequate font size options are available to ensure readability.

* **Severity:** Major

* **Type of Testing:** Functional, Accessibility

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** AC_AD1

* **Test Case Title:** Verify accessibility features documentation.

* **Test Case Description:** This test case evaluates the clarity and completeness of documentation related to accessibility features.

* **Test Suite:** Accessibility Features

* **Test Priority:** Medium

* **Preconditions:**

    * User has access to the platform's help or documentation section.

* **Test Data:** Platform's accessibility-related documentation.

* **Test Steps:**

    1. Locate the platform's help or documentation section.

    2. Search for information related to accessibility features.

    3. Review the documentation for clarity, accuracy, and comprehensiveness.

    4. Evaluate if it effectively guides users on how to enable and utilize accessibility features.

* **Postconditions:**

    * The accessibility documentation is reviewed.

* **Expected Result:** The documentation provides clear, concise, and accurate information about the platform's accessibility features, enabling users to easily understand and use them.

* **Severity:** Major

* **Type of Testing:** Usability

* **Test Case Approach:** Positive

Data - 17

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Safety and Moderation Tools

Feature description -

Safety and moderation tools are designed to protect users from harmful or inappropriate content and interactions on the platform. This may include features such as content moderation algorithms, reporting tools, blocking capabilities, comment filters, and community guidelines enforcement, fostering a safer and more positive online environment.

Userstory title -

As a user, I want safety and moderation tools so that I can have a safe and positive experience on the platform.

Acceptance criteria -

['Content moderation algorithms filter inappropriate content.', 'Users can report harmful or inappropriate content.', 'Blocking capabilities are available to users.', 'Comment filters prevent offensive language.', 'Community guidelines are enforced consistently.']

Generated Test Cases:

Here are 5 detailed test cases for the Social Media Platform's Safety and Moderation Tools, ensuring each acceptance criterion is covered:

Test Case 1

Test Case ID: SM-MOD-001

Test Case Title: Verify that content moderation algorithms filter inappropriate content.

Test Case Description: This test case verifies the effectiveness of the content moderation algorithms in identifying and flagging inappropriate content such as hate speech, profanity, and violence-inducing content.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

* A new user account is created.
* The platform is configured with predefined rules and filters for content moderation.
* Test content containing various forms of inappropriate language and themes is prepared.

Test Data:

* Pre-written test content: Posts containing hate speech, profanity, and violent imagery.

Test Steps:

1. Log in to the platform using the new user account.
2. Post the prepared test content onto the platform.
3. Observe the system's response to the posted content.
4. Verify that the inappropriate content is flagged or removed according to the platform's moderation rules.
5. Check if the user receives any notifications regarding the action taken on their content.

Postconditions:

* Flagged or removed content is no longer visible to other users.
* Appropriate actions (flagging, removal, or notification) are taken by the system.

Expected Result: The content moderation algorithms successfully identify and take appropriate action against the inappropriate content posted.

Severity: Critical

Type of Testing: Functional, System

Test Case Approach: Negative

Test Case 2

Test Case ID: SM-MOD-002

Test Case Title: Verify users can report harmful or inappropriate content.

Test Case Description: This test case ensures that users have a functioning mechanism to report content they deem harmful or inappropriate.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

* A new user account is created.

* Test content containing a violation of platform rules is uploaded.

Test Data:

* Test content: A post containing potentially sensitive personal information (e.g., someone's address)

Test Steps:

1. Log in to the platform using the new user account.

2. Locate the test content containing the violation.

3. Utilize the reporting mechanism (e.g., "Report" button or link) to flag the content.

4. Select the appropriate reason for reporting from the available options.

5. Optionally, provide additional details or context in a comment field.

6. Submit the report.

Postconditions:

* The content marked as reported is reviewed by platform moderators.

Expected Result: The user successfully reports the content, and the system forwards the report to the appropriate moderation team for review and action.

Severity: Major

Type of Testing: Functional, User Interface

Test Case Approach: Positive

Test Case 3

Test Case ID: SM-MOD-003

Test Case Title: Verify blocking capabilities are accessible to users.

Test Case Description: This test case confirms that users can block other users to prevent unwanted interactions and content from appearing in their feeds.

Test Suite: Safety and Moderation

Test Priority: Medium

Preconditions:

* Two user accounts are created: the "testing" user and the "blocked" user.

* The "blocked" user creates some content (posts, comments)

Test Data:

* Minimum two posts created by the "blocked" user.

Test Steps:

1. Log in with the "testing" user.

2. Navigate to the profile of the "blocked" user.

3. Locate the "Block" button or option.

4. Click on the "Block" button to initiate the blocking process.

5. Confirm the blocking action.

Postconditions:

* The "testing" user is now blocked from interacting with the "blocked" user.

Expected Result: The system successfully blocks the "blocked" user, preventing the "testing" user from seeing their content, receiving notifications from them, or engaging with their posts.

Severity: Major

Type of Testing: Functional, User Interface

Test Case Approach: Positive

Test Case 4

Test Case ID: SM-MOD-004

Test Case Title: Verify comment filters prevent offensive language.

Test Case Description: This test case examines the effectiveness of the comment filters in preventing the posting of offensive language.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

* A new user account is created.

Test Data:

* A list of common offensive words and phrases.

Test Steps:

1. Log in with the new user account.
2. Record the initial comment section of a post (to establish baseline).
3. Post each offensive word/phrase from the list as comments on the post.
4. Observe the system's response to each comment.
5. Observe if any comments are prevented from being posted.

Postconditions:

* Offensive comments are either prevented from posting or are automatically edited.

Expected Result: The comment filters effectively block or modify comments containing predefined offensive language, preventing the display of inappropriate content.

Severity: Critical

Type of Testing: Functional, System

Test Case Approach: Negative

Test Case 5

Test Case ID: SM-MOD-005

Test Case Title: Verify community guidelines are enforced consistently.

Test Case Description: This test case evaluates whether the platform enforces its community guidelines consistently across all types of user content and interactions.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

* Diverse set of user content (text posts, images, videos) is created to simulate various potential guideline violations.
* Content samples include: harassment, spam, misinformation, impersonation, etc.

Test Data:

* A list of content samples that clearly violate different community guidelines.

Test Steps:

1. Upload the prepared content samples onto the platform.
2. Observe the system's response to each content sample.

3. Review any actions taken by the platform (flagging, removal, notifications).

Postconditions:

* Violations are addressed according to the community guidelines.

Expected Result: The platform proactively identifies and takes action against content violating the community guidelines, ensuring a consistent and fair application of rules for all users.

Severity: Major

Type of Testing: Functional, System

Test Case Approach: Negative

These test cases provide a starting point for ensuring the safety and moderation tools on the Social Media Platform function effectively in protecting users and maintaining a positive online experience.

Data - 18

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Multi-Language Support

Feature description -

Multi-language support allows the platform to cater to users from diverse linguistic backgrounds by offering interface localization and content translation features. Users can select their preferred language for the interface, and posts or messages can be translated into different languages, enhancing accessibility and user experience across global audiences.

Userstory title -

As a user, I want multi-language support so that I can use the platform in my preferred language.

Acceptance criteria -

['Users can select their preferred interface language.', 'The platform offers content translation features.', 'Posts and messages can be translated into different languages.', 'Language preferences are saved and applied consistently.', 'The platform supports a wide range of languages.']

Generated Test Cases:

Test Cases for Multi-Language Support

Following the provided template and considering the project context and acceptance criteria:

Test Case 1

Test Case ID: MLT-001

Test Case Title: Verify that users can select their preferred interface language

Test Case Description: This test case verifies the functionality of selecting a preferred language from the available options in the user interface and confirms its successful application.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged into the platform
- Multiple languages are supported on the platform

Test Data: List of supported languages

Test Steps:

1. Navigate to user profile settings.
2. Locate the language selection option.
3. Choose a language different from the default.
4. Observe the UI elements and text updating to the selected language.

Postconditions:

- Platform settings are saved.
- User interface reflects the selected language.

Expected Result: Upon selecting a language, all UI elements and text are displayed in the chosen language without errors.

Severity: Major

Type of Testing: Functional, UI

Test Case Approach: Positive

Test Case 2

Test Case ID: MLT-002

Test Case Title: Verify that the platform offers content translation features

Test Case Description: This test case validates the system's ability to translate content posted by other users, ensuring cross-linguistic comprehension.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged into the platform.

- User has access to different language settings.

- Another user with a different language preference has posted content.

Test Data: Posts in a language different from the logged-in user's preference.

Test Steps:

1. Navigate to the news feed or any section where user-generated content is displayed.

2. Identify a post written in a different language.

3. Observe the platform's attempt to translate the content into the user's preferred language.

4. Verify the accuracy of the translation.

Postconditions:

- Content translation is applied.

- The translated content is displayed alongside the original.

Expected Result: The system successfully translates the content into the user's selected language, ensuring comprehension.

Severity: Major

Type of Testing: Functional, Translation

Test Case Approach: Positive

Test Case 3

Test Case ID: MLT-003

Test Case Title: Verify that posts and messages can be translated into different languages

Test Case Description: This test case examines the functionality of translating individual posts and messages into different languages, similar to testing cross-post translation but focused on user-initiated interaction.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged into the platform;

- User has access to the messaging feature.

- User has access to different language settings.

Test Data: A sample message to be translated.

Test Steps:

1. Compose a new message in English.

2. Select the "translate" option for the message.

3. Choose a different target language.

4. Send the translated message to another user.

Postconditions:

- User can select the translation target language

- Platform saves and displays the translated message correctly.

Expected Result: The message gets translated to the selected language, and the recipient sees it in their preferred language.

Severity: Major

Type of Testing: Functional, Translation

Test Case Approach: Positive

Test Case 4

Test Case ID: MLT-004

Test Case Title: Verify that language preferences are saved and applied consistently

Test Case Description: This test case ensures that the selected language preference is persisted and correctly applied across various platform functionalities and sessions.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

- User is logged into the platform.
- User selects a language preference.

Test Data: User's selected language preference.

Test Steps:

1. Select and save a language preference from the settings.
2. Log out of the platform.
3. Log back into the platform.
4. Observe the platform interface and any user-generated content displayed.

Postconditions:

- Platform interface reflects the saved language preference.
- Translations are consistent across various platform sections.

Expected Result: Upon logging back in, the platform remembers the user's language preference and displays the entire interface accordingly.

Severity: Major

Type of Testing: Functional, Recall

Test Case Approach: Positive

Test Case 5

Test Case ID: MLT-005

Test Case Title: Verify that the platform supports a wide range of languages.

Test Case Description: This test case validates the comprehensiveness of supported languages, covering major global languages and potentially regional ones based on the platform's target audience.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged into the platform.

Test Data: List of target languages to be tested.

Test Steps:

1. Navigate to user profile settings or language selection options.
2. Observe the available language choices.
3. Select one language from the list and confirm successful application.
4. Repeat steps 2 and 3 for a diverse set of target languages.

Postconditions:

- All target languages are successfully selectable.
- The platform displays the selected languages accurately.

Expected Result: The platform offers a diverse range of languages, allowing users to access and utilize the platform in their preferred language.

Severity: Major

Type of Testing: Functional, Completeness

Test Case Approach: Positive

Data - 19

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Cross-Platform Integration

Feature description -

Cross-platform integration allows users to connect and share content across different social media platforms and external services. This includes features like single sign-on (SSO) capabilities, cross-posting of content, integration with external media sources, and the ability to share platform content on other social networks, enhancing user convenience and content reach.

Userstory title -

As a user, I want cross-platform integration so that I can connect and share content across multiple platforms.

Acceptance criteria -

['Users can link their accounts with other social media platforms.', 'Content can be cross-posted to other platforms.', 'Users can sign in using single sign-on (SSO).', 'Integration with external media sources is available.', 'Content sharing across platforms is seamless.']

Generated Test Cases:

Here are 5 test cases for the provided input data, each covering a different acceptance criterion:

Test Case 1

Test Case ID: XPT_001

Test Case Title: Verify that users can link their accounts with other social media platforms.

Test Case Description: This test case ensures that users can successfully connect their existing accounts from platforms like Facebook, Twitter, or Google to their user profiles on the social media platform.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User has created an account on the social media platform.

* User has an active account on a supported third-party social media platform (e.g., Facebook).

Test Data: User credentials for both the social media platform and the chosen third-party platform.

Test Steps:

1. Navigate to the account settings on the social media platform.

2. Locate the "Connect with other platforms" or similar option.

3. Select the desired third-party platform (e.g., Facebook).

4. Authorize access to the social media platform's account using the third-party platform's login credentials.

5. Verify that the accounts are successfully linked.

Postconditions:

* Connected account information is displayed in the user's profile settings.

* User can access features that leverage the linked account (e.g., posting to Facebook from the platform).

Expected Result: The user's account is successfully linked with the chosen third-party platform, and the connection is reflected in the platform's user interface.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: XPT_002

Test Case Title: Verify that content can be cross-posted to other platforms.

Test Case Description: This test case aims to confirm that users can share their content from the social media platform to compatible external platforms like Twitter or Facebook.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User has created and posted content (text, photo, video) on the social media platform.

* User has successfully linked their social media account with a supported external platform (e.g., Twitter).

Test Data: Original content created by the user on the social media platform.

Test Steps:

1. Locate a piece of content on the social media platform (post, photo, video).

2. Click on the "Share" or "Cross-post" button.

3. Select the desired external platform (e.g., Twitter).

4. Preview the cross-posted content to ensure accuracy.

5. Confirm the cross-posting action.

Postconditions:

* The content is visible on the selected external platform (e.g., Twitter).

* The cross-posted content is attributed correctly to the user's social media platform account.

* Appropriate sharing mechanisms are available (e.g., customize caption, target audience).

Expected Result: The user's content successfully cross-posts to the intended external platform with proper attribution.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: XPT_003

Test Case Title: Verify that users can sign in using single sign-on (SSO).

Test Case Description: This test case ensures users can log into the social media platform seamlessly using their credentials from a linked external platform (e.g., Google, Facebook).

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

 * User has created an account on the social media platform.
 * User has linked their account with a supported SSO platform (e.g., Google).

Test Data: User credentials for the linked SSO platform (e.g., Google).

Test Steps:

1. Navigate to the login page of the social media platform.
2. Locate the SSO login option (e.g., "Login with Google").
3. Click on the SSO login button.
4. Authorize access to the social media platform using the SSO provider's credentials (e.g., Google account).
5. Verify that the user is successfully logged in to the social media platform.

Postconditions:

* User dashboard or profile page is loaded, indicating successful login.

Expected Result: The user is authenticated and logged into the social media platform using their SSO credentials without needing to enter separate email and password information.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: XPT_004

Test Case Title: Verify integration with external media sources.

Test Case Description: This test case aims to confirm the user's ability to incorporate content from external media sources (e.g., YouTube, Instagram, Vimeo) directly into their posts and interactions on the social media platform.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

* User account is authenticated on the social media platform.
* Platform supports embedding content from selected external media sources (e.g., YouTube).

Test Data: Links or URLs to content from supported external media sources.

Test Steps:

1. Navigate to the post composition or comment section on the platform.
2. Locate the option to embed external media (e.g., "Insert Link," "Embed Media").
3. Paste a link from a supported external media source into the designated field.
4. Review the preview of the embedded content to ensure it is displayed correctly.
5. Confirm the embedding action and view the final post or comment.

Postconditions:

* Embedded media content is displayed within the post or comment.
* Embedded content retains basic functionality (e.g., play button for videos, image zoom for photos).

Expected Result: The user can successfully embed media content from supported external sources into their posts and comments, enriching their interactions and providing added value.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: XPT_005

Test Case Title: Verify seamless content sharing across platforms.

Test Case Description: This test case examines the ease and functionality of sharing content from the platform to other social networks, focusing on the user experience and accuracy of data transfer.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

* User has posted content on the social media platform.
* User has linked their account with at least one other social media platform (e.g., Twitter).

Test Data: Existing post content on the social media platform.

Test Steps:

1. Locate a post on the social media platform.

2. Click on the "Share" or "Cross-post" button for the specific social network platform.

3. Review any sharing options or customization available (e.g., caption, privacy settings).

4. Confirm the sharing action.

5. Navigate to the target social media platform to verify the shared post.

6. Analyze the shared post for accuracy of information (e.g., text, images, links).

Postconditions:

* User can select the desired social network directly.

* Shared content appears correctly on the target platform.

Expected Result: Users can effortlessly share their content to different social media platforms with accurate information transfer, promoting wider audience reach and engagement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 20

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Gamification and Rewards

Feature description -

Gamification and rewards enhance user engagement by incorporating game-like elements such as points, badges, leaderboards, and challenges. Users can earn rewards for their activities, achievements, and contributions on the platform, fostering a sense of accomplishment and encouraging continued interaction and participation.

Userstory title -

As a user, I want gamification and rewards so that I can earn recognition and incentives for my activities.

Acceptance criteria -

['Users can earn points and badges for their activities.', 'Leaderboards display top users based on points or achievements.', 'Challenges are available for users to complete and earn rewards.', 'Rewards can include virtual goods, discounts, or exclusive content.', 'Users can view their progress and achievements in a dedicated section.']

Generated Test Cases:

Social Media Platform - Gamification and Rewards Test Cases

Test Case ID: Gm1234a

Test Case Title: Verify that users can earn points and badges for activities

Test Case Description: This test case aims to verify the functionality of earning points and badges for various user actions on the platform.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User account is created and logged in.
- Feature "Gamification" is enabled for the user account.

Test Data: User account, Social media content (posts, comments, shares), Simulated challenges

Test Steps:

1. Post a new text update on the user's timeline.
2. Comment on another user's post.
3. Share a user's post to the user's story.
4. Complete a designated challenge within the platform.
5. Navigate to the user profile section and review the earned points and badges.

Postconditions:

- Earned points should reflect in the user profile.
- Acknowledging notification of earned points.
- Awarded badges should be displayed in the user profile.

Expected Result: The system successfully records and displays earned points and badges for the user's activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: Gm4567b

Test Case Title: Verify that leaderboards display top users based on points or achievements

Test Case Description: This test case verifies the functionality of leaderboards, ensuring they accurately display top users sorted by points or achievements.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- Multiple user accounts are created and logged in.
- Users have actively participated in platform activities to earn points and badges.

Test Data: Simulated user activity data (points and achievements), leaderboard categories (points, badges)

Test Steps:

1. Access the platform's leaderboard section.
2. Select different leaderboard categories (e.g., points, badges).
3. Observe the ranking of users on each leaderboard.

Postconditions:

- Leaderboard displays correctly sort users by selected criteria (points or achievements).
-

Expected Result: Users are accurately ranked on leaderboards based on their earned points or achievements.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive, Negative (test cases with edge cases and user absence)

Test Case ID: Gm8901c

Test Case Title: Verify that challenges are available for users to complete and earn rewards

Test Case Description: This test case verifies the availability and functionality of challenges.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User account is created and logged in.
- "Gamification" feature is enabled for the user account.

Test Data: Example challenges with specified actions and rewards

Test Steps:

1. Navigate to the platform's "Challenges" section.

2. Select and review the available challenges.

3. Initiate and complete a selected challenge.

Postconditions:

- The challenge is successfully completed and marked as finished.

- The user receives the designated reward for completing the challenge.

Expected Result: Users can access challenges, complete them successfully, earn rewards, and the system acknowledges their accomplishment.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive, Negative (test cases with incomplete challenges and invalid actions)

Test Case ID: Gm1234d

Test Case Title: Verify that rewards include virtual goods, discounts, or exclusive content

Test Case Description: This test case verifies the nature and variety of rewards offered for user participation.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- User account is created and logged in.
- User earns rewards through completed challenges or activities.

Test Data: Different reward types (virtual goods, discounts, exclusive content)

Test Steps:

1. Redeem earned rewards from the user profile section.
2. Inspect the redeemed rewards and their functionality.

Postconditions:

- Rewards are successfully redeemed by the user.

Expected Result: The system offers a range of rewards, including virtual goods, platform discounts, or exclusive content access, accessible through redemption.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

"

Data - 21

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Creation

Feature description -

Users can create tasks by providing detailed information such as the task title, description, due date, priority level, and assigning it to specific team members. This feature serves as the foundation for organizing and managing tasks within the tool.

Userstory title -

As a user, I want to create tasks so that I can organize and manage my work effectively.

Acceptance criteria -

['Users can enter a task title, description, due date, and priority level.', 'Tasks can be assigned to specific team members.', 'The system should save and display created tasks.', 'Users can edit task details after creation.', 'The system should validate required fields (e.g., task title).']

Generated Test Cases:

Here are 5 test cases for the Task Management Tool feature "Task Creation," addressing all acceptance criteria:

Test Case 1:

* **Test Case ID:** TMS-001
* **Test Case Title:** Verify that users can enter required task details
* **Test Case Description:** This test case aims to ensure that users can successfully create tasks by providing mandatory information like title, description, due date, and priority level. It will validate the system's ability to store and display these details accurately.
* **Test Suite:** Task Creation
* **Test Priority:** High
* **Preconditions:**
    * User is logged in to the Task Management Tool application.
* **Test Data:**
    * Task Title: Meeting with John
    * Task Description: Discuss project progress and next steps.
    * Due Date: 2024-03-15
    * Priority Level: High
* **Test Steps:**
    1. Navigate to the "Create New Task" page.
    2. Enter the task title, description, due date, and priority level in the respective fields.
    3. Click the "Create Task" button.
* **Postconditions:**
    * A new task is created and displayed on the task list.
* **Expected Result:** The system successfully creates the task and displays it with the entered details.
* **Severity:** Major

* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TMS-002
* **Test Case Title:** Verify that tasks can be assigned to team members
* **Test Case Description:** This test case verifies the functionality of assigning tasks to specific team members during creation.
* **Test Suite:** Task Creation
* **Test Priority:** High
* **Preconditions:**
    * User is logged in as a project manager with access to team member details.
    * At least two team members are added to the project.
* **Test Data:**
    * Task Title: Design wireframes for new feature
    * Task Description: Create initial wireframes for the user interface.
    * Due Date: 2024-03-22
    * Priority Level: Medium
    * Team Member: John Doe
* **Test Steps:**
    1. Go to the "Create New Task" page.
    2. Enter the task title, description, due date, and priority level.
    3. In the "Assigned to" field, select John Doe from the team member dropdown.
    4. Click the "Create Task" button.
* **Postconditions:**
    * A new task is created and assigned to John Doe.

    * John Doe receives a notification about the assigned task.
* **Expected Result:** The system assigns the task to John Doe and notifies him accordingly.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TMS-003
* **Test Case Title:** Verify that the system saves and displays created tasks
* **Test Case Description:** This test case ensures that the system correctly saves created tasks and displays them accurately in the task list.
* **Test Suite:** Task Creation
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the Task Management Tool application.
* **Test Data:**
    * Task Title1: Prepare presentation for client meeting
    * Task Title2: Follow up with vendor regarding invoice
* **Test Steps:**
    1. Create two tasks with different titles using the "Create New Task" feature.
    2. Log out of the application.
    3. Log back into the application.
* **Postconditions:**
    * User's previously created tasks are visible in the task list.
* **Expected Result:** Both created tasks are displayed in the task list even after logging out and back in.
* **Severity:** Major

* **Type of Testing:** Functional/Data Storage

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TMS-004

* **Test Case Title:** Verify that users can edit task details after creation

* **Test Case Description:** This test case checks the ability to modify task information post-creation, ensuring tasks remain accurate and up-to-date.

* **Test Suite:** Task Creation

* **Test Priority:** Medium

* **Preconditions:**

    * User has created a task. The task exists in the system.

* **Test Data:**

    * Original Task Title: Write blog post

    * Updated Task Title: Complete Blog post Edit

    * Original Due Date: 2024-03-29

    * Updated Due Date: 2024-03-31

* **Test Steps:**

    1. Navigate to the task list and locate the previously created task.

    2. Select the task to open its details.

    3. Modify the task title and due date in the appropriate fields.

    4. Click the "Update Task" button.

* **Postconditions:**

    * The task details are updated in the system.

    * The updated task attributes are reflected in the task list.

* **Expected Result:** The system successfully updates the task title and due date according to the user's

changes.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TMS-005

* **Test Case Title:** Verify that the system validates required fields

* **Test Case Description:** This test case ensures the system enforces required field input for task creation. It covers scenarios where essential fields are left blank.

* **Test Suite:** Task Creation

* **Test Priority:** High

* **Preconditions:**

    * User is logged in to the Task Management Tool application.

* **Test Data:**

    * Incomplete task information (e.g., missing task title, due date)

* **Test Steps:**

    1. Attempt to create a new task without entering a valid title.

    2. Attempt to create a new task without setting a due date.

* **Postconditions:**

    * The system displays an error message indicating the missing field.

* **Expected Result:** Successful creation requires all required fields to be filled. The system should display clear error messages when mandatory fields are left blank, preventing incomplete tasks from being saved.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Negative

Data - 22

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Assignment

Feature description -

Tasks can be assigned to specific team members, delegating responsibility and ensuring clarity about who is accountable for each task. Assignees receive notifications about their assigned tasks, helping them stay informed and focused on their responsibilities.

Userstory title -

As a user, I want to assign tasks to specific team members so that responsibilities are clear and tasks are managed effectively.

Acceptance criteria -

['Users can select team members to assign tasks.', 'Assignees receive notifications about their tasks.', "The system should display assigned tasks in the assignee's task list.", 'Users can reassign tasks to different team members.', 'Task assignment history should be tracked.']

Generated Test Cases:

Here are 5 test cases designed to cover the acceptance criteria for the Task Assignment feature in your Task Management Tool project:

Test Case 1:

* **Test Case ID:** TM-ASSIGN-001
* **Test Case Title:** Verify that users can select team members to assign tasks.
* **Test Case Description:** This test case will confirm that users can successfully choose team members from a list of available users to assign to a task.
* **Test Suite:** Task Assignment
* **Test Priority:** High
* **Preconditions:**
    * A user with permissions to create and assign tasks is logged in.
    * At least two team members exist in the system.
* **Test Data:** Task details (title, description), list of team members
* **Test Steps:**
    1. Create a new task.
    2. Navigate to the "Assign Task" section.
    3. Select a team member from the available list.
    4. Verify that the selected team member is now displayed as the assignee for the task.
* **Postconditions:**
    * The selected team member is assigned to the task.
    * The task is saved successfully.
* **Expected Result:** The system correctly assigns the task to the chosen team member.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TM-ASSIGN-002
* **Test Case Title:** Verify that assignees receive notifications about their tasks.
* **Test Case Description:** This test case will ensure that notifications are delivered to the assignee when a task is assigned to them.
* **Test Suite:** Task Assignment
* **Test Priority:** High
* **Preconditions:**
   * A user with permissions to create and assign tasks is logged in.
   * A task assigned to a test user who has notification settings enabled.
* **Test Data:** Task details, notification settings
* **Test Steps:**
   1. Create a new task.
   2. Assign the task to a test user known to have notifications enabled.
   3. Observe the test user's notification panel (Inbox, system email, or designated notification app).
* **Postconditions:**
   * The task is successfully assigned.
* **Expected Result:** The assigned test user receives a notification about the assigned task.
* **Severity:** Major
* **Type of Testing:** Functional Testing, Integration Testing (if notifications are handled by a separate system)
* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TM-ASSIGN-003
* **Test Case Title:** Verify that assigned tasks are displayed in the assignee's task list.
* **Test Case Description:** This test case confirms that tasks assigned to a user are correctly presented in their personal task list.
* **Test Suite:** Task Assignment
* **Test Priority:** High
* **Preconditions:**
    * A user with permissions to create and assign tasks is logged in.
    * At least one task has been assigned to a test user.
* **Test Data:** Assigned task details
* **Test Steps:**
    1. Log in as the test user who has tasks assigned.
    2. Go to the user's task list or designated view for assigned tasks.
    3. Verify that the assigned tasks are present in the list.
* **Postconditions:**
    * The user's task list is displayed correctly.
* **Expected Result:** The assigned tasks are accurately reflected in the test user's task list.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TM-ASSIGN-004
* **Test Case Title:** Verify that users can reassign tasks to different team members.
* **Test Case Description:** This test case ensures that an initially assigned task can be successfully reassigned to another user.

* **Test Suite:** Task Assignment
* **Test Priority:** Medium
* **Preconditions:**
    * A user with permissions to create, assign, and reassign tasks is logged in.
    * A task has been previously assigned to a test user.
* **Test Data:** Task detail, list of available team members
* **Test Steps:**
    1. Find the task assigned to the test user.
    2. Select the option to reassign the task.
    3. Choose a different team member from the available list.
    4. Verify that the task is now assigned to the newly selected team member.
* **Postconditions:**
    * The task assignee is updated.
* **Expected Result:** The task is successfully reassigned to the chosen team member.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TM-ASSIGN-005
* **Test Case Title:** Verify that task assignment history is tracked.
* **Test Case Description:** This test case confirms that the system logs and maintains a record of who previously had tasks assigned to them.
* **Test Suite:** Task Assignment
* **Test Priority:** Medium
* **Preconditions:**

* A user with appropriate permissions is logged in.

* At least two separate assignments have been made to a specific task.

* **Test Data:** Task details

* **Test Steps:**

1. Find the task that has had multiple assignments.

2. Navigate to the task's history or audit log.

3. Verify that the list of past assignments, including the assignees at each point in time, is accurately recorded.

* **Postconditions:**

* Task history is displayed

* **Expected Result:** The history of task assignments is correctly documented within the system.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 23

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Deadline Reminders

Feature description -

Users receive reminders and notifications for upcoming task deadlines, ensuring that tasks are completed on time and preventing delays. Reminders can be customized based on users' preferences and urgency levels.

Userstory title -

As a user, I want to receive reminders for upcoming task deadlines so that I can complete tasks on time.

Acceptance criteria -

['Users can set reminders for tasks based on deadlines.', 'Reminders can be customized by urgency level.', 'The system sends notifications for upcoming deadlines.', 'Users can choose the delivery method for reminders (e.g., email, in-app).', 'Reminder settings can be adjusted or disabled.']

Generated Test Cases:

Here are 5 test cases for the "Task Deadline Reminders" feature, following your guidelines:

Test Case 1:

*   **Test Case ID:** R7J2P5
*   **Test Case Title:** Verify that users can set reminders for tasks based on deadlines.
*   **Test Case Description:** This test case aims to ensure that users have the functionality to set reminders for tasks associated with specific deadlines.
*   **Test Suite:** Task Deadline Reminders
*   **Test Priority:** High
*   **Preconditions:**
    *   A user is logged into the Task Management Tool.
    *   A new task has been created.
*   **Test Data:** A task with a future deadline date and time.

* **Test Steps:**
    1. Navigate to the task details page.
    2. Locate the "Reminder" section.
    3. Input the desired reminder time prior to the deadline.
    4. Select "Save Reminder."
* **Postconditions:** The reminder is scheduled for the specified time.
* **Expected Result:** The system successfully sets a reminder for the task based on the entered deadline and time.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** Yx4T9S
* **Test Case Title:** Verify that reminders can be customized by urgency level.
* **Test Case Description:** This test case verifies that users can differentiate the urgency of reminders and apply appropriate settings.
* **Test Suite:** Task Deadline Reminders
* **Test Priority:** Medium
* **Preconditions:**
    * A user is logged into the Task Management Tool.
    * A task with a future deadline exists.
* **Test Data:** Options for different urgency levels (e.g., Low, Medium, High).
* **Test Steps:**
    1. Navigate to the task details page.
    2. Locate the "Reminder" section.

    3. Select a desired urgency level for the reminder.

    4. Save the reminder settings.

*   **Postconditions:** The reminder is assigned the chosen urgency level.

*   **Expected Result:** The system successfully sets different reminder behaviors based on the selected urgency level.

*   **Severity:** Major

*   **Type of Testing:** Functional

*   **Test Case Approach:** Positive

Test Case 3:

*   **Test Case ID:** Z1N6B3

*   **Test Case Title:** Verify that the system sends notifications for upcoming deadlines.

*   **Test Case Description:** This test case confirms that the system sends timely notifications to users about approaching deadlines.

*   **Test Suite:** Task Deadline Reminders

*   **Test Priority:** High

*   **Preconditions:**

    *   A user is logged into the Task Management Tool.

    *   A task with a future deadline and a reminder set is created.

*   **Test Data:** A task deadline within a reasonable timeframe.

*   **Test Steps:**

    1. Set a reminder for the task with a specific time.

    2. Allow sufficient time before the reminder trigger point.

    3. Verify that the user receives a notification delivered via the chosen method (e.g., email, in-app).

*   **Postconditions:** The reminder notification is sent to the user.

*   **Expected Result:** The system successfully delivers a notification to the user close to the reminder time.

*   **Severity:** Critical

*   **Type of Testing:** Functional, Integration Testing

*   **Test Case Approach:** Positive

Test Case 4:

*   **Test Case ID:** G8Q2Y7

*   **Test Case Title:** Verify that users can choose the delivery method for reminders (e.g., email, in-app).

*   **Test Case Description:** This test case ensures the system allows users to decide how they receive reminders.

*   **Test Suite:** Task Deadline Reminders

*   **Test Priority:** Medium

*   **Preconditions:**

    *   A user is logged into the Task Management Tool.

    *   A task with a future deadline exists.

*   **Test Data:** Multiple reminder delivery options (e.g., Email, In-App notification, SMS).

*   **Test Steps:**

    1. Navigate to the task details page.

    2. Access the "Reminder" section.

    3. Select a desired delivery method from the available options.

    4. Save the reminder settings.

*   **Postconditions:** The reminder is configured to use the selected delivery method.

*   **Expected Result:** The system allows users to choose their preferred reminder delivery method and respects those settings.

*   **Severity:** Major

*   **Type of Testing:** Functional

*   **Test Case Approach:** Positive

Test Case 5:

*   **Test Case ID:** L5P9N2
*   **Test Case Title:** Verify that reminder settings can be adjusted or disabled.
*   **Test Case Description:** This test case validates that users can modify existing reminders or turn them off as needed.
*   **Test Suite:** Task Deadline Reminders
*   **Test Priority:** Medium
*   **Preconditions:**
    *   A user is logged into the Task Management Tool.
    *   A task with an existing reminder is created.
*   **Test Data:** An existing task reminder.
*   **Test Steps:**
    1. Navigate to the task details page.
    2. Open the "Reminder" section.
    3. Adjust the reminder time, urgency level, or delivery method.
    4. Disable the reminder entirely.
*   **Postconditions:** The reminder settings are updated based on user changes.
*   **Expected Result:** The system permits users to dynamically change reminder settings for individual tasks or switch reminders off completely.
*   **Severity:** Major
*   **Type of Testing:** Functional
*   **Test Case Approach:** Positive, Negative

Data - 24

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Prioritization

Feature description -

Users can prioritize tasks based on their urgency and importance, allowing them to focus on high-priority tasks and manage their workload effectively. Tasks can be categorized into different priority levels, such as high, medium, or low, based on their importance to the project or team.

Userstory title -

As a user, I want to prioritize tasks so that I can manage my workload effectively.

Acceptance criteria -

['Users can set priority levels for tasks (High, Medium, Low).', 'Tasks can be sorted based on priority levels.', 'Users can change the priority level of a task.', 'The system visually distinguishes tasks based on priority.', 'Tasks without a specified priority default to Medium.']

Generated Test Cases:

Here are 5 test cases for the Task Prioritization feature, following the provided template:

Test Case 1

Test Case ID: TKT-001

Test Case Title: Verify that users can set priority levels for tasks (High, Medium, Low)

Test Case Description: This test case verifies the ability to assign priority levels (High, Medium, Low) to newly created tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.

- A project is created.

Test Data: New task details including priority level (High, Medium, Low)

Test Steps:

1. Navigate to the project.

2. Click the "Add Task" button.

3. Fill in the task details, including a priority level selection.

4. Click "Save" to create the task.

Postconditions:

- Task is added to the project.

Expected Result: The task is saved with the selected priority level (High, Medium or Low).

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TKT-002

Test Case Title: Verify that tasks can be sorted based on priority levels

Test Case Description: This test case checks if the task list can be dynamically sorted based on priority levels (High, Medium, Low).

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.
- A project with multiple tasks of different priority levels exists.

Test Data: Project with multiple tasks of varying priorities.

Test Steps:

1. Open the project with tasks.
2. Click the "Priority" sorting option in the task list header.
3. Observe the order of tasks in the list.

Postconditions:

- Tasks are sorted based on priority.

Expected Result: Tasks appear in the list sorted from Highest to Lowest priority (High, Medium, Low).

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TKT-003

Test Case Title: Verify that users can change the priority level of a task

Test Case Description: This test case confirms the ability to modify the priority level of an existing task.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.

- A project with a task having a priority level exists.

Test Data: Existing task to modify.

Test Steps:

1. Open the project containing the task.

2. Locate the task and click the "Edit" option.

3. Change the existing priority level.

4. Click "Save" to update the task.

Postconditions:

- Task priority is updated.

Expected Result: The task's priority level is successfully changed to the new selection.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: TKT-004

Test Case Title: Verify visual distinction of tasks based on priority

Test Case Description: This test case assesses if task cards visually differentiate themselves based on assigned priority levels.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.

- A project with tasks of different priorities exists.

Test Data: Project with tasks having High, Medium, and Low priorities.

Test Steps:

1. Access the project containing the task with different priority levels.

2. Observe the task cards visually for any distinct colors, icons, or markers based on priority.

Postconditions:

- Task visual representation reflects priority levels.

Expected Result: Task cards clearly differentiate themselves based on priority (High, Medium, Low) through visual cues (e.g., color coding, icons).

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TKT-005

Test Case Title: Verify default priority for tasks without a specified level

Test Case Description: This test case confirms that tasks without explicitly assigned priority levels default to "Medium."

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in.

- A project exists.

Test Data: New task details without a specified priority level.

Test Steps:

1. Navigate to the project and create a new task.

2. Leave the priority level field empty.

3. Click "Save" to create the task.

Postconditions:

- Task is added to the project.

Expected Result: The task is saved with a default priority level of "Medium."

Data - 25

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Subtasks

Feature description -

Tasks can be broken down into smaller subtasks, enabling users to divide complex tasks into manageable components. Subtasks can have their own due dates, assignees, and dependencies, providing granularity and clarity in task execution.

Userstory title -

As a user, I want to create subtasks so that I can break down complex tasks into manageable parts.

Acceptance criteria -

['Users can create subtasks under main tasks.', 'Subtasks can have their own due dates and assignees.', 'The system tracks the progress of each subtask.', 'Subtasks can be marked as complete independently of the main task.', 'Dependencies between subtasks can be set.']

Generated Test Cases:

Test Cases for Subtasks Feature

Test Case ID: st1

Test Case Title: Verify that users can create subtasks under main tasks.

Test Case Description: This test case aims to validate the user's ability to create subtasks within an existing task.

Test Suite: Subtasks Feature

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.
- A new task has been created.

Test Data: New subtask details (title, description, due date)

Test Steps:

1. Navigate to the created task.

2. Click on the "Add Subtask" button.

3. Enter the subtask details (title, description, due date).

4. Save the subtask.

Postconditions:

- A new subtask is added under the main task.

Expected Result: The system successfully creates and displays the subtask under the main task, allowing hierarchical task breakdown.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: st2

Test Case Title: Verify that subtasks can have their own due dates and assignees.

Test Case Description: This test case verifies the functionality of setting unique due dates and assigning specific users to subtasks.

Test Suite: Subtasks Feature

Test Priority: High

Preconditions:

- Subtask created in Test Case ID: st1 exists.

Test Data: New date and assignee for the subtask.

Test Steps:

1. Click on the subtask created in the previous test case.

2. Modify the due date field.

3. Change the assigned user for the subtask.

4. Save the changes.

Postconditions:

- The subtask's due date and assignee are updated.

Expected Result: The system allows changing the due date and assignee of the subtask independently of the main task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: st3

Test Case Title: Verify the system tracks the progress of each subtask.

Test Case Description: This test case ensures the system accurately reflects the completion status of individual subtasks.

Test Suite: Subtasks Feature

Test Priority: High

Preconditions:

- Subtask created in Test Case ID: st1 exists.

- User is assigned as the assignee of the subtask.

Test Data: No specific test data required.

Test Steps:

1. Mark the subtask as "In progress".

2. Complete the subtask.

3. Observe the update in the subtask status.

Postconditions:

- The subtask's status is updated to "In progress" and then "Completed".

Expected Result: The system accurately updates the subtask's progress status as determined by user actions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: st4

Test Case Title: Verify that subtasks can be marked as complete independently of the main task.

Test Case Description: This test case verifies if a subtask can be marked as complete even if the main task is not finished.

Test Suite: Subtasks Feature

Test Priority: Medium

Preconditions:
- Subtask created in Test Case ID: st1 exists.

Test Data: No specific test data required.

Test Steps:
1. Complete the subtask and click "Mark Complete".
2. Observe the main task's status.

Postconditions:
- The subtask is marked as completed, but the main task's status remains unchanged.

Expected Result: Marking a subtask as complete does not automatically affect the completion status of the main task.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: st5
Test Case Title: Verify the functionality of setting dependencies between subtasks.
Test Case Description: This test case examines if subtasks can be linked with dependencies to ensure sequential execution.
Test Suite: Subtasks Feature
Test Priority: Medium

Preconditions:

- Two subtasks under the same main task exist.

Test Data:

- Subtask 1: Description: "Write a first draft"
- Subtask 2: Description: "Review and edit the draft"
- Dependency: Set "Review and edit the draft" (Subtask 2) as dependent on "Write a first draft" (Subtask 1)

Test Steps:

1. Attempt to mark "Review and edit the draft" (Subtask 2) as complete before "Write a first draft" (Subtask 1) is marked as complete.
2. Observe the system's response.

Postconditions:

- The system prevents marking Subtask 2 as complete before Subtask 1 is finished.

Expected Result: The system enforces dependency constraints, ensuring that subtasks are completed in the specified order.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Negative

Data - 26

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Dependencies

Feature description -

Users can define dependencies between tasks, specifying the order in which tasks should be completed to ensure smooth workflow and project progression. Task dependencies help in identifying critical paths and managing interdependencies between tasks.

Userstory title -

As a user, I want to set task dependencies so that tasks are completed in the correct sequence.

Acceptance criteria -

['Users can set dependencies between tasks.', 'The system displays dependencies in the task details.', 'Tasks cannot be marked as complete if dependent tasks are not completed.', 'Users can update or remove dependencies as needed.', 'The system provides alerts or warnings if a dependent task is not completed.']

Generated Test Cases:

Here are 5 test cases for the Task Dependencies feature of the Task Management tool, each covering a different acceptance criterion:

Test Case ID: TM-DEP-001

Test Case Title: Verify that users can set dependencies between tasks.

Test Case Description: This test case confirms that users can establish dependencies between tasks, linking them in a logical sequence.

Test Suite: Task Management - Task Dependencies

Test Priority: High

Preconditions:

* A user is logged into the system.
* Two tasks exist within a project.

Test Data: Task details like names, descriptions, and assigned users.

Test Steps:

1. Navigate to the project containing the two tasks.
2. Select the first task.
3. Open the task details view.
4. Locate the "Dependencies" section.
5. Choose the second task as a dependent task.
6. Save the changes.

Postconditions:

* The dependency link between the tasks is displayed in the task details.

Expected Result: The system successfully establishes a dependency between the tasks, ensuring the second task cannot be marked complete until the first task is finished.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TM-DEP-002

Test Case Title: Verify that the system displays dependencies in the task details.

Test Case Description: This test case ensures that the system visually represents task dependencies within the task details view for clear understanding.

Test Suite: Task Management - Task Dependencies

Test Priority: Medium

Preconditions:

* A user is logged into the system.
* A task has a dependency on another task.

Test Data: Task with a dependency relationship.

Test Steps:

1. Open the task details view of the task with the dependency.
2. Locate the "Dependencies" section.
3. Verify that the dependent task is displayed with a clear visual indicator (e.g., arrow, line) representing the relationship.

Postconditions:

* The user can see the dependent task.

Expected Result: The system displays the dependent task in the task details view, visually indicating the dependency.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TM-DEP-003

Test Case Title: Verify that tasks cannot be marked as complete if dependent tasks are not completed.

Test Case Description: This test case confirms that the system enforces the dependency relationship, preventing the completion of a task until its dependent tasks are finished.

Test Suite: Task Management - Task Dependencies

Test Priority: High

Preconditions:

* A user is logged into the system.
* A task has a dependency on another task.
* The dependent task is not yet completed.

Test Data: Tasks with a dependency relationship.

Test Steps:

1. Open the details view of the task that is dependent on another task.
2. Attempt to mark the dependent task as "Complete."
3. Observe the system's response.

Postconditions:

* The dependent task remains incomplete.

Expected Result: The system prevents the marking of the dependent task as "Complete" until its dependent task is marked as finished. An error message or warning should be displayed.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Negative

Test Case ID: TM-DEP-004

Test Case Title: Verify that users can update or remove dependencies as needed.

Test Case Description: This test case ensures that the system allows for flexibility in managing task dependencies, enabling users to update or remove them when required.

Test Suite: Task Management - Task Dependencies

Test Priority: Medium

Preconditions:

* A user is logged into the system.

* Two tasks exist with a defined dependency between them.

Test Data: Tasks with an existing dependency relationship.

Test Steps:

1. Open the details view of the task that has a dependency.

2. Locate the "Dependencies" section.

3. Modify the existing dependency to link it to a different task.

4. Remove the dependency entirely.

5. Save the changes.

Postconditions:

* The dependency is updated or removed as intended.

Expected Result: The system successfully updates or removes the dependency between the tasks according to the user's actions.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive/Modifying

Test Case ID: TM-DEP-005
Test Case Title: Verify that the system provides alerts or warnings if a dependent task is not completed.
Test Case Description: This test case confirms that the system notifies users when a dependent task falls behind in completion, highlighting potential workflow disruptions.
Test Suite: Task Management - Task Dependencies
Test Priority: High
Preconditions:
* A user is logged into the system.
* A task has a dependency on another task.
* The dependent task is not completed past its due date.

Test Data: Tasks with a dependency relationship and a past-due date on the dependent task.
Test Steps:
1. Login as the user who is assigned the task with the dependency.
2. Open the details view of the Task with the dependency.
3. Check if the system displays an alert or warning regarding the incomplete dependent task.

Postconditions:
* The system may offer a way to manage the dependencies based on the user's actions.

4.

Expected Result: The system provides a clear alert or warning message indicating that the dependent task is not completed, potentially impacting the progress of the task.

Severity: Major
Type of Testing: Functional, User Interface Testing
Test Case Approach: Positive

These test cases cover all the acceptance criteria outlined for the Task Dependencies feature and provide a structured approach for QA engineers to thoroughly evaluate this functionality during software testing.

Data - 27

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Comments and Collaboration

Feature description -

Users can collaborate on tasks by leaving comments, attaching files, and engaging in discussions directly within the task interface. This feature facilitates communication and teamwork, allowing team members to provide updates, share feedback, and ask questions related to the task.

Userstory title -

As a user, I want to comment on tasks so that I can collaborate with my team.

Acceptance criteria -

['Users can add comments to tasks.', 'Comments can include text, attachments, and links.', 'Users can reply to and edit their comments.', 'The system notifies users of new comments.', 'Comments are displayed in chronological order.']

Generated Test Cases:

Here are 5 detailed test cases for the provided user story and acceptance criteria:

Test Case 1:

Test Case ID: TC_TaskComment_01

Test Case Title: Verify that users can add comments to tasks

Test Case Description: This test case verifies the functionality of adding comments to a task. It ensures users can input text and add optional attachments to the comment.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is authenticated and logged in

- A task exists within a project

Test Data:

- Sample text comment: "Task progress update: 75% complete"
- A small image file (e.g., .jpg)

Test Steps:

1. Navigate to the task page.
2. Click on the "Add Comment" button.
3. Type the sample text comment in the comment box.
4. Upload the image file to the comment.
5. Click the "Post Comment" button.

Postconditions:

- Comment is displayed on the task page.

Expected Result: The comment with the text and attached image is successfully added to the task page and is visible to all users with access to the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_TaskComment_02

Test Case Title: Verify that users can reply to comments

Test Case Description: This test case verifies the functionality of replying to an existing comment on a task.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is authenticated and logged in
- An existing comment has been added to a task.

Test Data:

- Sample reply comment: "Thanks for the update! Can I get a heads-up when it's finished?"

Test Steps:

1. Navigate to the task page.
2. Locate the existing comment.
3. Click the "Reply" button below the existing comment.
4. Type the sample reply comment in the reply box.
5. Click the "Post Reply" button.

Postconditions:

- Reply is displayed under the corresponding comment.

Expected Result: The reply comment is successfully added below the selected comment and is visible to all users with access to the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_TaskComment_03

Test Case Title: Verify that users can edit their own comments

Test Case Description: This test case verifies the ability for users to edit their own comments after they have been posted.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is authenticated and logged in
- The user has created a comment on a task.

Test Data:

- Original comment text
- Modified comment text

Test Steps:

1. Navigate to the task page.
2. Locate the comment created by the user.
3. Click the "Edit" button next to the comment.
4. Change the comment text to the modified text.
5. Click the "Save Changes" button.

Postconditions:

- The comment is updated with the new text.

Expected Result: The comment is successfully edited, and the updated text is displayed on the task page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_TaskComment_04

Test Case Title: Verify comment display order

Test Case Description: This test case verifies that comments are displayed on the task page in chronological order, from oldest to newest.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is authenticated and logged in
- Multiple comments have been added to a task.

Test Data:

- TIMESTAMP1 (earliest comment time)

- TIMESTAMP2 (subsequent comment time)

- TIMESTAMP3 (latest comment time)

Test Steps:

1. Navigate to the task page.

2. Observe the order of the comments listed.

Postconditions:

- No new comments are added.

Expected Result: Comments are displayed in chronological order based on the timestamps of when they were created (TIMESTAMP1, TIMESTAMP2, TIMESTAMP3).

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_TaskComment_05

Test Case Title: Verify notification for new comments

Test Case Description: This test case verifies that the system notifies users when new comments are added to a task they are following or collaborating on.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User A is logged in and has access to a task.

- User B is logged in and also has access to the same task.

- A notification system is functioning

Test Data:

- A new comment text by User B

Test Steps:

1. User B adds a comment to the task while User A is logged in and active on another page.

2. Check User A's notification inbox.

Postconditions:

- User A is not actively on the task page.

Expected Result: User A receives a notification about the new comment added by User B.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 28

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

File Attachments

Feature description -

Users can attach relevant files, documents, and resources to tasks, ensuring that all necessary information is accessible in one place. File attachments can include documents, spreadsheets, images, links, and other resources needed for task execution.

Userstory title -

As a user, I want to attach files to tasks so that all relevant information is easily accessible.

Acceptance criteria -

['Users can attach multiple files to a task.', 'Attachments can include various file types (documents, images, etc.).', 'Users can view and download attachments.', 'The system limits the size of attachments to prevent data overload.', 'Users can remove attachments if needed.']

Generated Test Cases:

Test Cases for File Attachments Feature

User Story: As a user, I want to attach files to tasks so that all relevant information is easily accessible.

Acceptance Criteria:

* Users can attach multiple files to a task.

* Attachments can include various file types (documents, images, etc.).

* Users can view and download attachments.

* The system limits the size of attachments to prevent data overload.

* Users can remove attachments if needed.

Test Case 1

Test Case ID: fhgnR5-file-01

Test Case Title: Verify that users can attach multiple files to a task

Test Case Description: This test case ensures that users can attach more than one file to a single task, simulating real-world scenarios where multiple resources might be needed.

Test Suite: File Attachments

Test Priority: High

Preconditions:

  * User is logged in.

  * A task is created.

Test Data: 3 sample files of different types (e.g., .docx, .jpg, .pdf)

Test Steps:

  1. Navigate to the created task.

  2. Click on the "Attach File" button.

  3. Select the first file (e.g., .docx) and upload it.

  4. Click on the "Attach File" button again.

  5. Select the second file (e.g., .jpg) and upload it.

  6. Click on the "Attach File" button once more.

7. Select the third file (e.g., .pdf) and upload it.

Postconditions:

* All three files are listed as attachments to the task.

Expected Result: The system successfully uploads and lists all three selected files as attachments to the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: fdF8Ew-file-02

Test Case Title: Verify support for various file types

Test Case Description: This test case verifies if the system accepts common file types for attachment, ensuring broad compatibility.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

* User is logged in.
* A task is created.

Test Data: Sample files of different types (.docx, .pdf, .jpg, .png, .zip, .txt, etc.)

Test Steps:

1. Navigate to the created task.
2. Click on the "Attach File" button.
3. Select each sample file type in turn and upload it.

Postconditions:

* All files are listed as attachments to the task.

Expected Result: The system successfully uploads all different file types as attachments to the task. Potential errors with unsupported file types should be logged and user-friendly messages should be displayed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: Thf4w-file-03

Test Case Title: Verify users can view and download attachments

Test Case Description: This test case checks if users can successfully view and download the attached files, confirming proper functionality.

Test Suite: File Attachments

Test Priority: High

Preconditions:

* User is logged in.
* A task is created with at least one file attachment.

Test Data: Attached files from Test Case 2

Test Steps:

1. Navigate to the task with attachments.
2. Click on the thumbnail of one of the attached files.
3. Verify the file opens in the appropriate viewer or format.
4. Click on the download icon of the same file.
5. Verify the file downloads to the user's device successfully.

Postconditions:

* Attachment is viewed and downloaded successfully.

Expected Result: The system correctly displays and allows downloading of the attached files.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: fhgnR5-file-04

Test Case Title: Verify the system limits attachment file size

Test Case Description: This test case simulations exceeding allowed file size limits to ensure the system handles it appropriately and prevents data overload.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

* User is logged in.
* A task is created.

Test Data:

* A small file (under the allowed limit).
* A large file significantly exceeding the allowed limit (e.g., 100 MB if the limit is 10 MB).

Test Steps:

1. Navigate to the created task.
2. Click on the "Attach File" button.
3. Select the small file and upload it ? verify it uploads successfully.
4. Repeat step 3 with the large file ? verify the system prevents upload and displays an informative error message.

Postconditions:

* Small file is attached.

* Large file upload is blocked.

Expected Result: The system successfully uploads files within the allowed size limit and prevents uploads exceeding the limit with an appropriate error message.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive/Negative

Test Case 5

Test Case ID: fdF8Ew-file-05

Test Case Title: Verify users can remove attachments

Test Case Description: This test case verifies if users can easily remove unnecessary attachments from tasks, ensuring flexibility.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

* User is logged in.

* A task is created with at least one file attachment.

Test Data: An attached file.

Test Steps:

1. Navigate to the task with attachments.

2. Hover over the attachment you want to remove.

3. Click the "Delete" button or icon associated with the file.

4. Confirm the deletion in the prompt.

Postconditions:

* Attachment is removed from the task.

Expected Result: The system successfully removes the selected file attachment from the task. The task should no longer display the deleted file.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 29

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Recurring Tasks

Feature description -

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

Userstory title -

As a user, I want to set recurring tasks so that regular activities are automated.

Acceptance criteria -

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

Generated Test Cases:

Task Management Tool Test Cases - Recurring Tasks

Test Case ID: TMT-RCC-001

Test Case Title: Verify that users can set tasks to recur daily.

Test Case Description: This test case verifies the ability to set a task to recur daily and ensure the system generates new instances of the task accordingly.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

* User is logged in.

* A project is created.

Test Data:

* Task details (title, description, due date).

* Recurrence setting: Daily

Test Steps:

1. Create a new task within the project.

2. In the task properties, set the recurrence to "Daily."

3. Save the task.

4. Observe and verify if the system automatically generates a new instance of the task for the next day.

Postconditions:

* New daily task instances are generated.

Expected Result: The system successfully creates a daily recurring task and generates subsequent instances for each day.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RCC-002

Test Case Title: Verify that users can modify the recurrence pattern of a recurring task.

Test Case Description: This test case verifies the ability to change the recurrence pattern of an existing recurring task.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

* A recurring task with a daily recurrence pattern is created (refer to Test Case TMT-RCC-001).

Test Data:

* Existing daily recurring task

Test Steps:

1. Navigate to the details of the existing daily recurring task.

2. Edit the recurrence setting to a weekly pattern.

3. Save the changes.

4. Verify that the recurrence pattern for the task changes to weekly.

Postconditions:

* Recurrence pattern for the task is updated

Expected Result: The system successfully modifies the recurrence pattern of the recurring task to weekly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RCC-003

Test Case Title: Verify the system generates instances of recurring tasks automatically.

Test Case Description: This test case ensures the system automatically creates new instances of recurring tasks based on the configured recurrence pattern.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

* A weekly recurring task is created (see Test Case TMT-RCC-002).

Test Data:

* Existing weekly recurring task

Test Steps:

1. Log out and log back in to simulate time passing.

2. Observe and verify if new instances of the weekly recurring task are generated for subsequent weeks.

Postconditions:

* New instances are generated based on the configured weekly recurrence pattern.

Expected Result: The system automatically creates new instances of the weekly recurring task for each subsequent week according to the set recurrence pattern.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RCC-004

Test Case Title: Verify users can cancel the recurrence of a recurring task.

Test Case Description: This test case confirms that users have the option to stop the automatic creation of instances for a recurring task.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

* A monthly recurring task is created.

Test Data:

* Existing monthly recurring task

Test Steps:

1. Navigate to the details of the monthly recurring task.

2. Select the option to cancel the recurrence.

3. Save the changes.

4. Verify that no new instances of the task are generated.

Postconditions:

* Recurrence for the task is stopped.

Expected Result: The system successfully cancels the recurrence of the task, preventing the generation of new instances.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RCC-005

Test Case Title: Verify the system notifies users of upcoming instances of recurring tasks.

Test Case Description: This test case confirms that users receive notifications for upcoming instances of recurring tasks.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

* A daily recurring task with a due date in the near future is created.

* User has notification preferences enabled.

Test Data:

* Existing daily recurring task with a specific due date.

Test Steps:

1. Simulate passage of time until the upcoming instance of the recurring task is due.

2. Observe and verify if the user receives a notification for the upcoming instance.

Postconditions:

* User receives notification about the recurring task.

Expected Result: The system successfully notifies the user about the upcoming instance of the recurring task based on the configured recurrence pattern and user notification preferences.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 30

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Tags and Labels

Feature description -

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

Userstory title -

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

Acceptance criteria -

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

Generated Test Cases:

Here are 5 test cases for the Task Management Tool, based on the provided acceptance criteria:

Test Case 1

Test Case ID: tagMNP1

Test Case Title: Verify that users can add tags to tasks

Test Case Description: This test case verifies the functionality of adding various tags to a task. It ensures the tag field is accessible and allows for inputting diverse tags, covering both single and multiple tag assignments.

Test Suite: Task Tag and Label Management

Test Priority: High

Preconditions:

- User is logged in
- User has access to create new tasks

Test Data:

- A list of diverse tags (e.g., #projectA, #urgent, #lowPriority, #clientX, #design)

Test Steps:

1. Navigate to the task creation page.
2. Create a new task with a description.
3. In the task tagging field, enter one tag and save the task.
4. Add another tag to the same task and verify it is displayed.
5. Repeat steps 3 and 4 with multiple tags encompassing different categories.

Postconditions:

- The task is saved with the assigned tags

Expected Result: Multiple tags can be added to a task, and they are displayed correctly on the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: tagEditABC

Test Case Title: Verify that tags can be customized and edited

Test Case Description: This test case ensures the user can modify existing tags on a task, allowing for flexibility and correctness in task categorization.

Test Suite: Task Tag and Label Management

Test Priority: High

Preconditions:

- User is logged in
- User has created a task with tags

Test Data:

- Existing tag on the task
- Modified tag

Test Steps:

1. Open the task page with existing tags.
2. Click on the edit icon next to one of the tags.
3. Modify the tag to a new value.
4. Click "Save" to apply the change.

Postconditions:

- The tag is updated

Expected Result: The existing tag is replaced with the edited tag on the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: tagFilterGHI

Test Case Title: Verify that tasks can be filtered and sorted based on tags

Test Case Description: This test case confirms the functionality of filtering and sorting tasks based on assigned tags, allowing users to focus on specific task sets efficiently.

Test Suite: Task Tag and Label Management

Test Priority: High

Preconditions:

- User is logged in
- The system has multiple tasks with different tags

Test Data:

- A specific tag used on multiple tasks

Test Steps:

1. Navigate to the task list page.

2. Select the chosen tag from the tag filter dropdown.
3. Verify that only tasks with that specific tag are displayed.
4. Use the sort functionality (ascending/descending) on the tag filter.
5. Verify that tasks are sorted accordingly.

Postconditions:

- The task list is filtered based on the selected tag

Expected Result: The task list accurately displays only tasks associated with the chosen tag and allows sorting based on the tag.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: tagRemoveJLM

Test Case Title: Verify that users can remove tags from tasks

Test Case Description: This test case ensures that users possess the ability to delete unwanted tags from tasks, maintaining task organization and accuracy.

Test Suite: Task Tag and Label Management

Test Priority: High

Preconditions:

- User is logged in
- User has created a task with tags

Test Data:

- An existing tag on the task

Test Steps:

1. Open the task page with assigned tags.

2. Identify the tag to be removed.
3. Click the "remove" option (icon or button) next to the chosen tag.
4. Confirm the tag removal.

Postconditions:

- The tag is deleted from the task

Expected Result: The selected tag is successfully removed from the task, updating its tag assignment.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: tagDisplayNPO

Test Case Title: Verify that the system prominently displays tags and labels in the task list

Test Case Description: This test case ensures that the displayed tags for each task are clear, visible, and easy to identify within the task list, aiding user comprehension and navigability.

Test Suite: Task Tag and Label Management

Test Priority: High

Preconditions:

- User is logged in
- Tasks with tags are present in the task list

Test Data:

- Tasks with various tags

Test Steps:

1. Navigate to the task list view.
2. Observe the display of tags for each task.
3. Verify that tags are displayed prominently alongside relevant task information.

4. Ensure that tags are easily distinguishable from other task elements.

Postconditions:

- Task list remains visible and accessible

Expected Result:

The system displays tags for each task clearly and prominently, facilitating quick identification and categorization.

Severity: High

Type of Testing: Usability

Test Case Approach: Positive

Data - 31

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Status Tracking

Feature description -

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

Userstory title -

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

Acceptance criteria -

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

Generated Test Cases:

Here are 5 test cases for the Task Status Tracking feature, based on the provided acceptance criteria:

Test Case 1

* **Test Case ID:** TST-TASK-001
* **Test Case Title:** Verify that users can update task status to "To Do", "In Progress", and "Completed"
* **Test Case Description:** This test case aims to ensure that users can effectively change the status of a task within the system, validating the core functionality of task status updates.
* **Test Suite:** Task Management
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the system.
    * A new task has been created.
* **Test Data:** Task details (title, description, assignee, deadline, etc.)
* **Test Steps:**
    1. Navigate to the task list.
    2. Select the created task.
    3. Click on the "Status" dropdown menu.

   4. Change the status to "To Do".

   5. Change the status to "In Progress."

   6. Change the status to "Completed."

* **Postconditions:**

   * The task status is updated accordingly in the task list.

* **Expected Result:** The task status successfully changes between "To Do", "In Progress," and "Completed" in the task list.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TST-TASK-002

* **Test Case Title:** Verify that the system displays task status prominently in the task list.

* **Test Case Description:** This test case ensures that the task status is clearly displayed within the task list, enabling users to quickly identify the current state of each task.

* **Test Suite:** Task Management

* **Test Priority:** High

* **Preconditions:**

   * User is logged into the system.

   * Multiple tasks with different statuses are created.

* **Test Data:** Task details (title, description, assignee, deadline, status, etc.)

* **Test Steps:**

   1. Access the task list view.

   2. Observe the visual representation of task status for each task (e.g., icons, colors, labels).

* **Postconditions:**

    * All tasks display their respective statuses.

* **Expected Result:** Each task in the list clearly displays its status using a visually distinct format, allowing users to easily understand their current state.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TST-TASK-003

* **Test Case Title:** Verify that users can view the progress percentage of tasks.

* **Test Case Description:** This test case confirms that the system accurately calculates and displays the progress percentage for tasks, providing a quantitative measure of completion.

* **Test Suite:** Task Management

* **Test Priority:** High

* **Preconditions:**

    * User is logged into the system.

    * A task with defined subtasks is created.

* **Test Data:** Task details (title, description, assignee, deadline, subtasks, completion status)

* **Test Steps:**

    1. Navigate to the task with subtasks.

    2. Observe the displayed progress percentage for the task.

    3. Individually mark subtasks as completed.

    4. Observe the updates in the task's progress percentage.

* **Postconditions:**

    * The task's progress percentage reflects the completion status of its subtasks accurately.

* **Expected Result:** The progress percentage dynamically updates based on the completion status of

subtasks, providing a clear indication of the task's overall progress.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive, Regression

Test Case 4

* **Test Case ID:** TST-TASK-004

* **Test Case Title:** Verify that the system provides notifications for status changes.

* **Test Case Description:** This test case confirms whether the system notifies users about status updates for tasks they are assigned to or involved with, ensuring timely communication and awareness.

* **Test Suite:** Task Management, Notification System

* **Test Priority:** Medium

* **Preconditions:**

    * User A is assigned to a task.

    * User B is a watcher of the same task (can see updates).

    * Notification settings are enabled for both users.

* **Test Data:** Task details, notification settings

* **Test Steps:**

    1. User A changes the task status to "In Progress".

    2. Observe User A's notification feed.

    3. Observe User B's notification feed.

* **Postconditions:**

    * Both User A and User B receive notifications about the status change.

* **Expected Result:** System sends timely notifications to all relevant users upon changes in a task's status.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TST-TASK-005

* **Test Case Title:** Verify that task status can be filtered and sorted in the task list.

* **Test Case Description:** This test ensures that users can efficiently filter and sort tasks based on their status, allowing for a tailored view of task progress.

* **Test Suite:** Task Management

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the system.

    * Multiple tasks with different statuses are created.

* **Test Data:** Task details (title, description, assignee, deadline, status, etc.)

* **Test Steps:**

    1. Open the task list view.

    2. Utilize the "Status" filter dropdown to select specific statuses (e.g., "To Do," "In Progress").

    3. Observe how the task list conforms to the selected filter.

    4. Click the header of the "Status" column to sort tasks by status (ascending/descending).

* **Postconditions:**

    * The task list accurately reflects the selected filter and sorting order.

* **Expected Result:** The system enables users to effectively filter and sort tasks by status, providing customized views of task progress.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Data - 32

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Filtering and Sorting

Feature description -

Users can filter and sort tasks based on various criteria such as due date, priority, assignee, or project. Filtering and sorting options help users focus on specific tasks or view tasks from different perspectives, enhancing task management and organization.

Userstory title -

As a user, I want to filter and sort tasks so that I can focus on specific tasks or view tasks from different perspectives.

Acceptance criteria -

['Users can filter tasks based on criteria such as due date, priority, assignee, or project.', 'Tasks can be sorted by various attributes.', 'The system provides multiple filter and sort options in the task list.', 'Users can save and apply custom filter and sort settings.', 'Filters and sorts do not affect the actual data, only the view.']

Generated Test Cases:

Test Cases for Task Filtering & Sorting Feature

Here are 5 test cases for the "Task Filtering and Sorting" feature in the Task Management Tool, covering all acceptance criteria:

Test Case 1:

* **Test Case ID:** TMT-FS-001
* **Test Case Title:** Verify that users can filter tasks by due date
* **Test Case Description:** This test case ensures the system allows filtering tasks based on their due date, including past, present, and future dates.
* **Test Suite:** Task Filtering and Sorting
* **Test Priority:** High
* **Preconditions:**
    * There are tasks with different due dates created in the system.
    * User is logged in to the task management tool.
* **Test Data:**
        * Tasks with due dates in past, present, and future.
* **Test Steps:**
    1. Navigate to the task list view.
    2. Select the "Due Date" filter option.
    3. Choose "Past Due" filter.
    4. Verify that only tasks with due dates in the past are displayed.
    5. Repeat steps 3 and 4 for "Today" and "Upcoming" filter options.
* **Postconditions:**

  * User returns to the complete task list view.
* **Expected Result:** The system successfully filters tasks based on selected due date criteria. Task list displays the corresponding tasks only.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TMT-FS-002
* **Test Case Title:** Verify that users can sort tasks by priority
* **Test Case Description:** This test case ensures the system allows sorting tasks by priority, whether in ascending or descending order.
* **Test Suite:** Task Filtering and Sorting
* **Test Priority:** High
* **Preconditions:**
    * There are tasks with different priority levels created in the system.
    * User is logged in to the task management tool.
* **Test Data:** Tasks with different priority levels (e.g., High, Medium, Low).
* **Test Steps:**
    1. Navigate to the task list view.
    2. Click on the "Priority" column header.
    3. Verify that tasks are sorted in ascending order (Low to High).
    4. Click on the "Priority" column header again.
    5. Verify that tasks are now sorted in descending order (High to Low).
* **Postconditions:**
    * User returns to the default task list view.

* **Expected Result:** The system successfully sorts tasks based on priority level in both ascending and descending order.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TMT-FS-003

* **Test Case Title:** Verify that users can filter tasks by assigned user

* **Test Case Description:** This test case verifies the system allows filtering tasks based on the assigned user, showcasing individual task ownership.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** High

* **Preconditions:**

    * Multiple users are created in the system.

    * Tasks are assigned to different users.

    * User is logged in to the task management tool.

* **Test Data:**

    * Tasks assigned to different users.

* **Test Steps:**

    1. Navigate to the task list view.

    2. Select the "Assigned To" filter option.

    3. Choose a specific user from the dropdown list.

    4. Verify that only tasks assigned to the selected user are displayed.

* **Postconditions:**

    * User returns to the complete task list view.

* **Expected Result:** The system accurately filters tasks based on the designated assignee.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TMT-FS-004
* **Test Case Title:** Verify that users can save and apply custom filter settings
* **Test Case Description:** This test case checks if the system allows users to save custom filter combinations and apply them later for efficient task management.
* **Test Suite:** Task Filtering and Sorting
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged in to the task management tool.
* **Test Data:** Combination of filters like due date (Upcoming), Priority (High), and Project (Project A).
* **Test Steps:**
    1. Apply the desired filters (due date, priority, project, etc.).
    2. Locate the option to save the filter combination as a custom filter.
    3. Provide a name for the custom filter.
    4. Save the custom filter setting.
    5. Later, choose the saved custom filter from the available list.
    6. Verify that the system displays only tasks matching the saved filter criteria.
* **Postconditions:**
    * Custom filter is saved.
* **Expected Result:** The system successfully saves and applies custom filter settings, allowing users to quickly access specific task sets.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TMT-FS-005

* **Test Case Title:** Verify that filters and sorts do not affect the actual data

* **Test Case Description:** This test case checks that applying filters or sorting does not modify the underlying task data, ensuring data integrity.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Medium

* **Preconditions:**

    * A set of tasks with various properties (due date, priority, assignee, etc.) is created in the system.

    * User is logged in to the task management tool.

* **Test Data:**

    * Tasks with diverse attributes.

* **Test Steps:**

    1. View the full list of tasks.

    2. Apply filter(s) and sort(s) using different criteria.

    3. Verify that the displayed tasks match the applied filter/sort criteria.

    4. Remove filter(s) and sort(s).

5. Verify that the task list returns to displaying all original tasks.

* **Postconditions:**

    * User returns to the full task list view.

* **Expected Result:** Applying filters/sorts does not alter the original task data. The changes are purely visual and data remains unchanged.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive/Negative

Data - 33

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Time Tracking

Feature description -

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

Userstory title -

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Acceptance criteria -

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

Generated Test Cases:

Here are five test cases for the provided user story and acceptance criteria:

Test Case 1

Test Case ID: TMT-TC001

Test Case Title: Verify that users can start and stop a timer for each task.

Test Case Description: This test case aims to verify the functionality of the timer feature within the task management tool.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

* User is logged in to the task management tool.

* A new task has been created.

Test Data:

No test data needed.

Test Steps:

1. Navigate to the created task.

2. Click on the "Start Timer" button.

3. Observe that the timer starts counting.

4. After a specified duration, click on the "Stop Timer" button.

5. Verify that the timer stops counting and the total time spent on the task is recorded.

Postconditions:

* The timer is stopped.

* The total time spent on the task is displayed accurately.

Expected Result: The system successfully starts and stops the timer for the specified task, accurately recording the time spent.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TMT-TC002

Test Case Title: Verify that the system records and displays total time spent on tasks.

Test Case Description: This test case verifies the system's ability to record and display the total time spent on a task across multiple timer starts and stops.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

* User is logged in to the task management tool.

* A new task has been created.

* The timer has been started and stopped multiple times.

Test Data:

* Data related to multiple timer starts and stops for the task

Test Steps:

1. Access the task with recorded timer data.

2. Review the displayed total time spent on the task.

3. Validate that the total time accurately reflects the sum of all timer intervals.

Postconditions:

* The total time spent on the task is displayed accurately.

Expected Result: The system correctly calculates and displays the total time spent on the task, taking into account all timer sessions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TMT-TC003

Test Case Title: Verify that users can manually enter or adjust time entries for tasks.

Test Case Description: This test case verifies if users can manually modify time entries associated with tasks

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* User is logged in to the task management tool.
* A task exists with recorded timer data.

Test Data:

* Manually entered time values.

Test Steps:

1. Open the task with recorded timer data.
2. Locate the time tracking section for the task.
3. Manually enter a new time value for the task.
4. Confirm if the system allows adjustments to existing time entries.

Postconditions:

* Time entry is updated.

Expected Result: User can successfully enter new time values and edit existing time entries for tasks, allowing for manual adjustment of time logs.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Both Positive and Negative (Attempting to enter invalid data)

Test Case 4

Test Case ID: TMT-TC004

Test Case Title: Verify that time tracking data is stored and can be exported.

Test Case Description: This test case confirms if the time tracking data is persistently stored and can be exported in a usable format.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* User is logged in to the task management tool.
* Time has been tracked on multiple tasks.

Test Data:

No test data needed.

Test Steps:

1. Navigate to the time tracking reports section.
2. Select the desired export format (e.g., CSV, Excel).

3. Initiate the export process.

4. Verify if a file with the exported data is generated.

Postconditions:

* Exported time tracking data file is created successfully.

Expected Result: The system stores time tracking data persistently and allows its export in the selected format.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TMT-TC005

Test Case Title: Verify that the system provides reports on time spent per task or project.

Test Case Description: This test case examines the functionality of generating reports summarizing time spent on tasks or projects.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* User is logged in to the task management tool.

* Time has been tracked on multiple tasks within projects.

Test Data:

No test data needed.

Test Steps:

1. Access the reporting section of the time tracking module.
2. Select the desired report type (e.g., time spent per task, time spent per project).
3. Generate the report for a specific date range or period.
4. Analyze the generated report for accuracy.

Postconditions:

* Report is generated successfully.

Expected Result: The system generates reports displaying the time spent on tasks or projects according to the selected criteria, providing actionable insights into time allocation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 34

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Templates

Feature description -

Users can create and use templates for common task types or project workflows, streamlining task creation and standardizing processes across projects. Task templates can include predefined task structures, descriptions, assignees, and due dates, saving time and ensuring consistency in task execution.

Userstory title -

As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

Acceptance criteria -

['Users can create and save task templates.', 'Templates can include predefined task structures, descriptions, assignees, and due dates.', 'Users can apply templates to new tasks or projects.', 'Templates can be edited or deleted as needed.', 'The system displays a list of available templates for selection.']

Generated Test Cases:

Here are 5 test cases for the provided User Story, adhering to the Test Case Creation Template and covering all acceptance criteria:

Test Case 1

Test Case ID: TKT-001

Test Case Title: Verify that users can create and save task templates

Test Case Description: This test case ensures that users can define new task templates with predefined structures, descriptions, assignees, and due dates, and successfully save them for future use.

Test Suite: Task Management

Test Priority: High

Preconditions:

* User is logged in
* User has the necessary permissions to create templates

Test Data:

* Template Name: Project Kick-off
* Description: A template for initial project setup tasks
* Tasks:
  * Task 1: Define project scope (Assigned to: Project Manager, Due Date: Today)
  * Task 2: Create project timeline (Assigned to: Content Team, Due Date: Tomorrow)
* Template Category: Project Management

Test Steps:

1. Navigate to the Task Templates section.
2. Click on the "Create Template" button.
3. Enter the template name "Project Kick-off".
4. Add a description for the template.
5. Define the tasks within the template, including their assignments and due dates.
6. Select the desired category for the template (e.g., Project Management).
7. Click on the "Save Template" button.

Postconditions:

* The template "Project Kick-off" is displayed in the list of available templates.

Expected Result: The system successfully creates and saves the "Project Kick-off" task template.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TKT-002

Test Case Title: Verify that templates include predefined structures, descriptions, and assigned users

Test Case Description: This test case validates that task templates encompass predefined task structures, descriptions, and assigned users, ensuring consistency when applying templates to new tasks or projects.

Test Suite: Task Management

Test Priority: High

Preconditions:

  * User is logged in

  * A task template with predefined elements is available

Test Data:

  * Existing Task Template: "Project Meeting Preparation"

Test Steps:

  1. Go to the Task Templates section.

  2. Select the "Project Meeting Preparation" template.

  3. Apply the template to a new task or project.

  4. Examine the newly created task or project.

Postconditions:

  * A new task or project has been created.

Expected Result: The newly created task or project inherits the predefined structure, description, and assigned user(s) from the "Project Meeting Preparation" template.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TKT-003

Test Case Title: Verify that users can apply task templates to new tasks or projects

Test Case Description: This test case aims to confirm that users can effectively apply existing task templates to either new individual tasks or entire projects, simplifying task setup across different workflows.

Test Suite: Task Management

Test Priority: High

Preconditions:

  * User is logged in

  * One or more task templates are saved in the system.

Test Data:

  * Existing Task Template: "Content Creation Workflow"

Test Steps:

  1. Navigate to the Create New Task section.

  2. Locate and select the "Content Creation Workflow" template.

  3. Click the "Apply Template" button.

  4. (To test applying to a project) Navigate to the Create New Project section.

  5. Select the "Content Creation Workflow" template.

  6. Click the "Create Project" button.

Postconditions:

  * A new task (or project) has been created based on the template.

Expected Result: The system successfully applies the "Content Creation Workflow" template to the newly created task or project, including its predefined structure and elements.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: TKT-004

Test Case Title: Verify that users can edit or delete existing task templates

Test Case Description: This test case verifies if users have the capabilities to modify existing task templates by changing their structure, descriptions, or assignee information, as well as the ability to delete unwanted templates. *Test Suite:* Task Management

Test Priority: High

Preconditions:

* User is logged in
* A task template has been created and saved.

Test Data:

* Existing Task Template: "Project Kick-off"

Test Steps:

1. Navigate to the Task Templates section.
2. Locate and select the "Project Kick-off" template.
3. Edit the template's description, change the assignee of a task within the template, etc.
4. Click the "Update Template" button to save changes.
5. Click the "Delete" button next to "Project Kick-off."

Postconditions:

* The "Project Kick-off" template is modified or removed from the list.

Expected Result: The system successfully allows editing and deleting the "Project Kick-off" task template, updating or removing it as needed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive/Negative

Test Case 5

Test Case ID: TKT-005

Test Case Title: Verify that the system displays a list of available task templates

Test Case Description: This test case confirms that the system presents a clear and accessible list of available task templates, enabling users to easily find and utilize templates for various tasks or projects. *Test Suite:* Task Management

Test Priority: Medium

Preconditions:

* User is logged in

* At least two task templates exist

Test Data: No test data needed

Test Steps:

1. Navigate to the Task Templates section.

2. Verify that the list displays all available task templates.

Postconditions:

* The user can view the list of task templates.

Expected Result: The system displays a comprehensive list of available task templates, allowing users to easily browse and select the desired template.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 35

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Calendar Integration

Feature description -

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

Userstory title -

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

Acceptance criteria -

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

Generated Test Cases:

Test Cases for Calendar Integration Feature:

Test Case ID: TMG_1

Test Case Title: Verify that users can sync tasks with Google Calendar.

Test Case Description: This test case verifies the functionality of syncing tasks with an external calendar, specifically Google Calendar.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

* User is logged into their account.

* Google Calendar account is configured and accessible.

Test Data: Sample tasks with varying deadlines.

Test Steps:

1. Navigate to the "Settings" section of the application.

2. Under "Calendar Integration," select "Google Calendar" as the desired calendar service.

3. Authorize the application to access the user's Google Calendar account.

4. Create a new task with a specific deadline within the application.

5. Verify that the task is automatically created as an event in the user's Google Calendar with the accurate title, time, and description.

Postconditions:

* Task remains synced with Google Calendar even after logouts and logins.

Expected Result: The task will be successfully synced with the user's Google Calendar, reflecting all relevant details.

Severity: Major

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case ID: TMG_2

Test Case Title: Verify that the system displays task deadlines in the calendar view.

Test Case Description: This test case ensures task deadlines are accurately reflected within the integrated calendar view.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

* User is logged into their account.
* Tasks have been created with various deadlines.

Test Data: Sample tasks with different deadlines close to or on distinct days.

Test Steps:

1. Navigate to the "Calendar" view within the application.
2. Verify that all tasks with set deadlines appear as events on the calendar, displayed on the corresponding date.
3. Click on each task event to confirm the accurate task title, description, and deadline.

Postconditions:

* No task deadlines are missing from the calendar view.

Expected Result: All tasks with assigned deadlines will be displayed correctly in the calendar view, providing a clear overview of upcoming obligations.

Severity: Major

Type of Testing: Functional, User Interface

Test Case Approach: Positive

Test Case ID: TMG_3

Test Case Title: Verify that users can set reminders for tasks via the calendar.

Test Case Description: This test case confirms the functionality of setting reminders for tasks through the integrated calendar feature.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

* User is logged into their account.

* Google Calendar account is configured and accessible.

Test Data: Sample task with a specific deadline.

Test Steps:

1. Navigate to the task within the application.

2. Switch to the "Calendar" tab for the task.

3. Select the "Set Reminder" option and choose a desired reminder time before the task deadline.

4. Verify that a reminder event is created in Google Calendar corresponding to the selected time.

Postconditions:

* Reminder notification is successful when triggered by the calendar.

Expected Result: Task reminders will be successfully set and displayed in the user's Google Calendar, providing timely notifications.

Severity: Major

Type of Testing: Functional, User Interface

Test Case Approach: Positive

Test Case ID: TMG_4

Test Case Title: Verify that calendar events update automatically when tasks are changed.

Test Case Description: This test case checks if the calendar reflects changes to tasks, ensuring real-time synchronization.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

* User is logged into their account.
* Google Calendar account is configured.
* Tasks have been synced with the Google Calendar.

Test Data: Existing tasks with scheduled deadlines.

Test Steps:

1. Modify the deadline of an existing synced task within the application.
2. Observe the corresponding event in the Google Calendar.

3. Verify that the calendar event reflects the updated deadline.

Postconditions:

* Changes to task status and deadlines are reflected accurately.

Expected Result: Any changes made to task details within the application will be automatically reflected in the corresponding calendar event on the user's Google Calendar.

Severity: Major

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case ID: TMG_5

Test Case Title: Verify that users can manage calendar sync settings within the app.

Test Case Description: This test case ensures users can control their calendar sync settings, including disabling sync and switching between calendars.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

* User is logged into their account

* At least one calendar integration is set up (e.g., Google Calendar)

Test Data: Different calendar settings (sync enabled/disabled, calendar selection).

Test Steps:

1. Navigate to the "Settings" section of the application.

2. Go to the "Calendar Integration" settings.

3. Disable sync for the chosen calendar. Verify that no new tasks are added to that calendar.

4. Enable sync again. Verify that new tasks are synced with the calendar.

5. Change the synced calendar to a different account. Confirm the new calendar is active and syncing.

Postconditions:

* Sync status of chosen calendar is properly reflected.

* Chosen calendar remains in sync with the application.

Expected Result: Users should be able to effectively manage their calendar integrations, enabling and disabling sync, switching between different calendars, as needed.

Severity: Major

Type of Testing: Functional, User Interface

Test Case Approach: Positive

Data - 36

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Duplication

Feature description -

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

Userstory title -

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

Acceptance criteria -

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

Generated Test Cases:

Here are 5 test cases for the Task Duplication feature, following the provided template and addressing the acceptance criteria:

Test Case 1: Verify that users can duplicate basic tasks

Test Case Description: This test case verifies the ability to duplicate a basic task containing essential details like assignee, description, and deadline.

Test Suite: Task Management

Test Priority: High

Preconditions:

- A user is logged in with appropriate permissions

- A task with an assignee, description, and deadline exists in a project

Test Data: A sample task with predefined data

Test Steps:

1. Navigate to the project containing the task.
2. Locate the task to be duplicated.
3. Click on the "Duplicate" button associated with the task.
4. Verify that a new task is created with the same assignee, description, and deadline as the original task.
5. Check if the duplicated task has a distinct identifier (e.g., modified task name or ID).

Postconditions:

- The duplicated task appears in the task list.
- The original task remains unchanged.

Expected Result: The system successfully duplicates the task, retaining all original details including assignee, description, and deadline, and the duplicated task is identifiable as a new entry.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that duplicated tasks have a clear indication

Test Case Description: This test case ensures the system provides a clear indication that a task has been duplicated.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- A user is logged in with appropriate permissions
- A task exists in a project

Test Data: A sample task

Test Steps:

1. Duplicate the existing task (refer to Test Case 1 for steps).
2. Examine the duplicated task's display in the task list.
3. Verify the existence of a label or visual cue indicating the task is a duplicate (e.g., "Copy of" followed by the original task name).

Postconditions:

none

Expected Result: The system displays a clear indication, such as "Copy of" followed by the original task name, to differentiate the duplicated task from the original.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify modify ability of duplicated tasks

Test Case Description: This test case verifies the ability to modify the fields of a duplicated task after creation.

Test Suite: Task Management

Test Priority: High

Preconditions:

- A user is logged in with appropriate permissions
- A task has been duplicated (refer to Test Case 1 for steps).

Test Data: Modified data for the duplicated task (e.g., updated assignee, description, or deadline)

Test Steps:

1. Locate the duplicated task in the task list.
2. Click on the task to edit its details.
3. Modify the assignee, description, deadline, or any other field.
4. Save the changes.
5. Verify that the modifications are reflected in the duplicated task.

Postconditions:

- The duplicated task's details are updated

Expected Result: The system successfully allows modification of all fields in the duplicated task, ensuring it can be customized as needed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify duplication of task template

Test Case Description: This test case ensures the ability to duplicate task templates, preserving their pre-defined configurations.

Test Suite: Task Management

Test Priority: High

Preconditions:

- A task template with pre-defined details (assignee, description, deadline, etc.) exists

Test Data: A task template with predefined data

Test Steps:

1. Navigate to the task template section.
2. Locate the desired task template.
3. Click on the "Duplicate" button associated with the template.
4. Verify that a new task template is created with all data copied from the original template.

Postconditions:

- The duplicated task template appears in the task templates list.

Expected Result: The system successfully duplicates the task template, retaining all predefined configurations and making it available for reuse while preserving the original template.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify handling of blank task details during duplication

Test Case Description: This test case verifies that the system handles cases where a task has incomplete or blank details when duplicated.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- A user is logged in with appropriate permissions
- A task exists with some fields blank (e.g., no description or assignee)

Test Data: A task with blank or incomplete fields

Test Steps:

1. Duplicate the task.
2. Examine the details of the duplicated task.

Postconditions:

- The duplicated task appears in the task list.

Expected Result:

- The system handles the blank or incomplete details appropriately (e.g., leaving the fields blank in the duplicated task or prompting the user with appropriate messages)

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 37

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task History

Feature description -

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

Userstory title -

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

Acceptance criteria -

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

Generated Test Cases:

Here are five test cases designed to cover the acceptance criteria for the Task History feature in your Task Management Tool:

Test Case 1

* **Test Case ID:** TMT-HIST-001
* **Test Case Title:** Verify that users can view a detailed history of changes to tasks
* **Test Case Description:** This test case aims to confirm that users can access and view a comprehensive history of modifications made to a task, including changes to its assigned user, priority, descriptions, attachments, and status.
* **Test Suite:** Task History
* **Test Priority:** High
* **Preconditions:**
    * A user is logged into the system.
    * A task has been created and assigned to the user.
    * Some changes have been made to the task (e.g., priority updated, description revised)
* **Test Data:** Task with at least two modifications made
* **Test Steps:**
    1. Navigate to the task details page.

  2. Locate the "Task History" section.

  3. Review the list of changes displayed in the history.

  4. Verify that the history accurately reflects the made changes.

* **Postconditions:**

  * The history section is accessible

  * No user input is required after accessing the history.

* **Expected Result:** The task history section displays a clear and detailed list of changes made to the task, including the date and time stamps, the type of change, and the specific details of the modification.

* **Severity:** Critical

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TMT-HIST-002

* **Test Case Title:** Verify that the history includes modifications, comments, and other relevant activity

* **Test Case Description:** This test case ensures that the task history captures a variety of actions beyond just changes to core task attributes. It includes comments added by team members and other relevant activity.

* **Test Suite:** Task History

* **Test Priority:** High

* **Preconditions:**

  * A user is logged into the system.

  * A task exists with at least one comment added to it.

* **Test Data:** Task with existing comments

* **Test Steps:**

  1. Navigate to the task details page.

  2. Go to the "Task History" section.

  3. Check if the comments added to the task are present in the history.

4. Verify comments are associated with the correct user and timestamp.

* **Postconditions:**

  * No changes to the task are made.

* **Expected Result:** The task history accurately lists all comments made on the task, along with the user who made the comment and the date and time of the comment.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TMT-HIST-003

* **Test Case Title:** Verify that users can filter the history by type of change

* **Test Case Description:** This test case verifies the functionality of filtering task history based on the type of change made.

* **Test Suite:** Task History

* **Test Priority:** Medium

* **Preconditions:**

  * A user is logged into the system.

  * A task has multiple changes, including modifications to attributes, comments, and attachments.

* **Test Data:** Task with various change types: updates, comments, attachments

* **Test Steps:**

  1. Access the task details page and the "Task History" section.

  2. Use the filter option to select "Modifications" (e.g., change to priority or description).

  3. Verify that only history entries related to attribute modifications are displayed.

  4. Repeat the process with different filter options (e.g., "Comments," "Attachments") to confirm accurate

filtering.

* **Postconditions:**

    * Filters are applied within the history section.

* **Expected Result:** The user can successfully filter the task history based on specific change types, narrowing down the view to relevant information.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TMT-HIST-004

* **Test Case Title:** Verify that the system provides timestamps for all changes.

* **Test Case Description:** This test case confirms that every change recorded in the task history is precisely timestamped, allowing for accurate tracking of when changes occurred.

* **Test Suite:** Task History

* **Test Priority:** High

* **Preconditions:**

    * A user is logged in.

    * A task exists with multiple changes at different times.

* **Test Data:** Task with changes made at varying timestamps

* **Test Steps:**

    1. Open the task details page and view the "Task History" section.

    2. Examine each recorded change.

    3. Verify that a timestamp is associated with each change, representing the precise time it was made.

* **Postconditions:**
    * No changes to the task are made after reviewing the history.
* **Expected Result:** All entries in the task history display clear and accurate timestamps, indicating the exact time each change, comment, or action was performed.

* **Severity:** Critical
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TMT-HIST-005
* **Test Case Title:** Verify that task history is accessible from the task details page
* **Test Case Description:** This test case ensures that the task history section is readily accessible and prominent within the task details page, allowing users to easily access it.
* **Test Suite:** Task History
* **Test Priority:** Medium
* **Preconditions:**
    * A user is logged into the system.
    * A task is created.

* **Test Data:** Task with no changes, initially
* **Test Steps:**
    1. Navigate to the task details page.
    2. Locate the "Task History" section.
    3. Verify that the section is clearly labelled and easy to find.

4. Confirm that the history section is populated, even if the task has no changes.

* **Postconditions:**

    * The history section is accessible within the task details page.

* **Expected Result:** The "Task History" section is consistently positioned on the task details page (e.g., in a dedicated tab or area) and is prominently displayed, allowing users to quickly access the task history.

* **Severity:** Major

* **Type of Testing:** Usability

* **Test Case Approach:** Positive

Data - 38

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Notifications

Feature description -

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

Userstory title -

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

Acceptance criteria -

['Users receive notifications for task updates, comments, mentions, and other relevant activities.', 'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.', 'The system logs notification history for future reference.']

Generated Test Cases:

Here are five test cases designed to cover all acceptance criteria for the provided user story:

Test Case 1

* **Test Case ID:** TMS-NT01
* **Test Case Title:** Verify that users receive notifications for task updates
* **Test Case Description:** This test case verifies that users receive notifications when a task they are assigned to or watching is updated by another user.
* **Test Suite:** Task Notifications
* **Test Priority:** High
* **Preconditions:**
    * A user is logged into the Task Management Tool.
    * A task exists with assigned users and collaborators.

* **Test Data:** Sample task with updates (e.g., status change, due date change, comment added)
* **Test Steps:**
    * User A assigns a task to User B and User C.
    * User B updates the task status.
    * User C adds a comment to the task.
* **Postconditions:**
    * User A receives notifications for both the status update and the comment.
* **Expected Result:** Both User A receives email, push notification, or in-app alerts depending on their notification preferences.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TMS-NT02
* **Test Case Title:** Verify custom notification settings for task updates
* **Test Case Description:** This test verifies that users can customize their notification settings for task updates and receive notifications accordingly.
* **Test Suite:** Task Notifications
* **Test Priority:** Medium
* **Preconditions:**
    * A user is logged into the Task Management Tool.
* **Test Data:** Various notification settings (Email, Push, In-app, None)
* **Test Steps:**
    1. Navigate to user profile settings.
    2. Modify the notification settings for task updates.

   3. Create a new task and assign it to another user.
* **Postconditions:**
    * User confirms notification preferences.
* **Expected Result:** The user receives notifications for task updates only through the chosen notification channels.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TMS-NT03
* **Test Case Title:** Verify Email Notification Content for Task Updates
* **Test Case Description:** This test case ensures the email notification for task updates provides relevant information.
* **Test Suite:** Task Notifications
* **Test Priority:** High
* **Preconditions:**
    * A user is logged into the Task Management Tool.
    * Email notification is enabled.
* **Test Data:** Task with details (title, updated status, assignees, updated by)
* **Test Steps:**
    1. Another user updates a task assigned to the logged-in user.
    2. Observe the email received by the logged-in user.
* **Postconditions:**
    * User reviews email content.

* **Expected Result:** The email clearly states, "You have received an update on the task [Task Title]."; includes the updated status, the original assignee(s), and the user who made the update.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TMS-NT04

* **Test Case Title:** Verify Notification History

* **Test Case Description:** This test verifies the system correctly logs and displays notification history for a user.

* **Test Suite:** Task Notifications

* **Test Priority:** Medium

* **Preconditions:**

    * A user is logged into the Task Management Tool.

    * Multiple task updates, comments, or mentions have occurred.

* **Test Data:** Sample tasks with various activities (updates, comments, mentions).

* **Test Steps:**

    * User A performs several actions on tasks: updates status, adds comments, mentions another user.

    * User A access their notification history.

* **Postconditions:**

    * User reviews notification history.

* **Expected Result:** The notification history displays all past notifications for User A, including the type of notification (task update, comment, mention), the task title, the affected attributes, and the timestamp of each notification.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TMS-NT05

* **Test Case Title:** Verify handling of no new task updates

* **Test Case Description:** Ensures the system doesn't issue unnecessary notifications when there are no updates on tasks a user is watching.

* **Test Suite:** Task Notifications

* **Test Priority:** Medium

* **Preconditions:**

    * A user is logged into the Task Management Tool.

    * The user is watching a task.

* **Test Data:** No updates are made to tasks the user is watching.

* **Test Steps:**

    * Designated time interval passes (e.g., 30 minutes) with no updates to the watched task.

    * User checks for notifications.

* **Postconditions:**

    * User confirms notification absence.

* **Expected Result:** The user does not receive any notifications pertaining to the watched task because there were no updates during the specified time interval.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Negative

Remember, these test cases serve as a starting point.

Data - 39

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Archiving

Feature description -

Completed or inactive tasks can be archived or moved to a separate section, decluttering the task list and maintaining focus on active tasks. Task archiving helps users keep their task lists organized and prioritized, without losing access to important historical data.

Userstory title -

As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks.

Acceptance criteria -

['Users can archive completed or inactive tasks.', 'Archived tasks are moved to a separate section for easy access.', 'Users can search and view archived tasks.', 'The system provides an option to restore archived tasks to the active list.', 'Archived tasks retain all original details and history.']

Generated Test Cases:

Task Archiving Test Cases for Task Management Tool

Here are 5 test cases covering the acceptance criteria for Task Archiving in the Task Management Tool:

Test Case 1

Test Case ID: TMA-001

Test Case Title: Verify that users can archive completed tasks.

Test Case Description: This test case verifies the functionality of archiving completed tasks from the active task list. It ensures that the system correctly identifies completed tasks and moves them to the designated archive section.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in.
- A task with a "completed" status exists in the active task list.

Test Data: A completed task.

Test Steps:

1. Locate the completed task in the active task list.
2. Click the "archive" button associated with the completed task.
3. Verify that the task is removed from the active task list.
4. Navigate to the archived tasks section.

5. Confirm that the archived task is present in the archived tasks list.

Postconditions:

- Archived task can be found in the dedicated archive section.

Expected Result: The completed task is successfully archived and moved to the designated archive section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TMA-002

Test Case Title: Verify that archived tasks are accessible through search functionality.

Test Case Description: This test case verifies if users can search for and retrieve archived tasks using keywords or filters.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in.
- At least one task is archived in the system.

Test Data: Keywords or filters related to the archived task content.

Test Steps:

1. Navigate to the search bar within the task list or archive section.
2. Enter keywords or filters specific to the archived task.
3. Click the search button.
4. Verify that the searched archived task is displayed in the results.

Postconditions:

- Searched task appears within the search results.

Expected Result: The system successfully retrieves the archived task based on the provided search criteria.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TMA-003

Test Case Title: Verify that users can restore archived tasks.

Test Case Description: This test case confirms if the system allows restoring archived tasks back to the active task list.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in.
- At least one task should be archived in the system.

Test Data: An archived task.

Test Steps:

1. Navigate to the archived tasks list.
2. Locate the desired archived task.
3. Click the ?restore? button associated with the task.
4. Verify that the task is removed from the archived tasks list.
5. Check that the task reappears in the active task list.

Postconditions:

- Restored task is visible in the active task list.

Expected Result: The archived task is successfully restored back to the active task list with its original details and status.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: TMA-004

Test Case Title: Verify that archived tasks retain all original details.

Test Case Description: This test case ensures that archived tasks keep all their original information, including descriptions, attachments, deadlines, and assigned users.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in.
- A task with detailed information (description, attachment, deadline, assignee) is archived.

Test Data: Archived task with existing details.

Test Steps:

1. Locate the archived task in the archive section.
2. Open the archived task.

3. Verify that all original details (description, attachment, deadline, assigned user) are preserved.

Postconditions:

- All original details of the archived task are intact.

Expected Result: The archived task maintains all original information without any data loss or modification.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TMA-005

Test Case Title: Verify that tasks can be selectively archived.

Test Case Description: This test case confirms if users can choose specific tasks to archive instead of archiving all tasks at once.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in.
- Multiple tasks exist in the active task list.

Test Data: A few tasks from the active task list.

Test Steps:

1. Navigate to the active task list.
2. Select specific tasks for archiving using individual checkboxes.
3. Click the "archive" button.
4. Verify that only the selected tasks are moved to the archive section.

Postconditions:

- Only the selected tasks are archived and remain inactive.

Expected Result: The system accurately archives only the selected tasks, leaving the unselected tasks in the active list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 40

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Guest Access

Feature description -

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

Userstory title -

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

Acceptance criteria -

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

Generated Test Cases:

Here are 5 test cases based on your input data, structured according to the provided template:

Test Case 1

Test Case ID: TC_GuestInv1

Test Case Title: Verify that users can invite external collaborators as guests.

Test Case Description: This test case validates the functionality of inviting external collaborators with limited access as guests to view and interact with specific tasks or projects.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in with administrative privileges.
- A project or task exists in the system.

Test Data: Email address of an external collaborator.

Test Steps:

1. Navigate to the project or task page.
2. Locate the "Invite Guest" or similar option.
3. Enter the external collaborator's email address.
4. Select the desired access level (e.g., view-only, limited editing).
5. Send the invitation.

Postconditions:

- An invitation email is sent to the external collaborator.
- The guest user receives an invitation notification within the system.

Expected Result: The system successfully sends an invitation to the external collaborator, granting them guest access as selected.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_GuestView1

Test Case Title: Verify that guests can view tasks and projects.

Test Case Description: This test case ensures that guest users can access and view the content of designated tasks and projects.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- A guest user has been invited and accepted access to a project or task.

Test Data: Existing tasks and project details within the designated scope.

Test Steps:

1. Guest user logs in to the platform.

2. Navigate to the provided project or task list.

3. Access individual tasks and review their information (description, assigned user, deadlines, etc.).

Postconditions: No specific postconditions.

Expected Result: The guest user can successfully view all project/task details within the granted access level.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_AccessDistinction

Test Case Title: Verify clear distinction between guest and full access roles.

Test Case Description: This test case confirms that the system differentiates between guest and full access user roles, preventing unauthorized access to sensitive areas.

Test Suite: Guest Access

Test Priority: Medium

Preconditions:

- A guest user and a regular user are logged in.

Test Data: Navigation menus, system settings, restricted areas (if any).

Test Steps:

1. Guest user attempts to access specific menus/features within the platform.
2. Regular user attempts to access the same menus/features, and also attempts to modify guest user permissions.

Postconditions: Guests should not be able to access certain areas and edit/modify settings.

Expected Result: The system restricts guest users from accessing restricted areas and modifying settings. Regular users can access all areas and manage guest user permissions.

Severity: Major

Type of Testing: Security, Functional

Test Case Approach: Positive, Negative

Test Case 4

Test Case ID: TC_GuestManagement

Test Case Title: Verify user's ability to manage guest access.

Test Case Description: This test case ensures that users can add, remove, and modify user permissions for guest collaborators.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- A user with administrative privileges is logged in.
- At least one guest user is already added to a project.

Test Data:

- Email addresses of guest users.
- Different guest access levels.

Test Steps:

1. Navigate to the project settings.
2. In the "Guests" section, select a guest user.
3. Edit the guest user's access level.
4. Remove the guest user from the project.
5. Add a new guest user with a specific access level.

Postconditions:

- Changes to guest user permissions are reflected in the system.

Expected Result:

The system effectively allows users to manage guest access by editing permissions, removing users, and adding new ones.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TC_GuestRestrictedAccess

Test Case Title: Verify that guests cannot access sensitive areas.

Test Case Description: This test case ensures that guest users are prevented from accessing sensitive or restricted areas of the platform, such as user account settings or financial data.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- A guest user is logged in.
- Identify restricted areas in the platform (e.g., account settings, billing information).

Test Data: Links or navigation paths to sensitive areas.

Test Steps:

1. Guest user attempts to access identified restricted areas.

Postconditions: No specific postconditions.

Expected Result: The system blocks guest access to all restricted areas and displays appropriate error messages or denies access.

Severity: Critical

Type of Testing: Security

Test Case Approach: Negative

Data - 41

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Import/Export

Feature description -

Users can import tasks from external sources or export tasks to other task management tools or file formats. Import/export capabilities facilitate data migration and integration with other systems, ensuring seamless collaboration and workflow management across platforms.

Userstory title -

As a user, I want to import and export tasks so that I can migrate data between different task management tools.

Acceptance criteria -

['Users can import tasks from external sources in various formats (e.g., CSV, Excel).', 'Users can export tasks to other task management tools or file formats.', 'The system provides clear mapping of fields during import/export.', 'Users can preview imported data before finalizing the process.', 'The system logs import/export activities for future reference.']

Generated Test Cases:

Here are 5 detailed test cases for the provided input data, following the template:

Test Case 1

Test Case ID: TME57A

Test Case Title: Verify that users can import tasks from a CSV file

Test Case Description: This test case verifies the ability to import tasks from a CSV file, checking for successful mapping of fields and data integrity.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

- A valid CSV file containing task data is prepared.
- User is logged into the task management tool

Test Data: CSV file with task data (title, description, assignee, due date)

Test Steps:

1. Navigate to the "Import Tasks" section.
2. Select the CSV file containing task data.
3. Map the fields from the CSV file to corresponding fields in the task management tool.
4. Review the preview of imported tasks.
5. Click "Import" to finalize the process.

Postconditions:

- Tasks are imported successfully.
- Imported task data matches the CSV file data.

Expected Result: The system successfully imports tasks from the CSV file, accurately mapping fields and preserving data integrity.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TMI85B

Test Case Title: Verify that users can export tasks to a CSV file

Test Case Description: This test case verifies the ability to export tasks to a CSV file and ensures the exported data is accurate and comprehensive.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

- User is logged into the task management tool with existing tasks.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Export Tasks" section.
2. Select the "CSV" format for export.
3. Specify the range of tasks to export (e.g., all tasks, selected tasks).
4. Click "Export" to initiate the process.

Postconditions:

- A CSV file containing the selected task data is downloaded.

Expected Result: The system generates a CSV file containing the selected task data, including relevant fields (title, description, assignee, due date, etc.)

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TMF39C

Test Case Title: Verify that the system provides clear field mapping during import

Test Case Description: This test case evaluates the clarity and accuracy of field mapping guidance provided during the import process.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

- A CSV file with task data is prepared.
- User is logged into the task management tool.

Test Data: CSV file with task fields (title, description, assignee, due date).

Test Steps:

1. Navigate to the "Import Tasks" section.
2. Select the CSV file.
3. Observe the field mapping interface presented by the system.
4. Examine the mapping options for each CSV field.

Postconditions:

- User has successfully understood and mapped fields.

Expected Result: The system provides a clear and intuitive interface for mapping CSV fields to corresponding task fields, ensuring accurate data import.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: TMP91D

Test Case Title: Verify that users can preview imported data before finalization

Test Case Description: This test case ensures that users can review the imported data before committing to the import, allowing for verification and correction of potential errors.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

- A CSV file with task data is prepared.
- User is logged into the task management tool

Test Data: CSV file with task data (title, description, assignee, due date)

Test Steps:

1. Navigate to the "Import Tasks" section.
2. Select the CSV file.
3. Map the fields from the CSV file.
4. Click on the "Preview" button to review the imported data.
5. Check for any discrepancies or errors in the preview.

Postconditions:

- User has reviewed the previewed data.

Expected Result: The system allows users to preview the imported data before finalizing the import, enabling them to identify and address potential issues.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TML72E

Test Case Title: Verify that the system logs import/export activities

Test Case Description: This test case verifies that the task management tool records import and export activities for logging and auditing purposes.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

- User is logged into the task management tool
- at least one task exists in the system

Test Data: No test data needed

Test Steps:

1. Export tasks from the system to a CSV file.
2. Log out of the system.
3. Log back into the system
4. Navigate to the system logs or activity log section.
5. Search for the recorded export activity.

Postconditions:

- The export activity is recorded in the system logs.

Expected Result: The system successfully logs import and export activities, including details like date, time, format, uploaded/downloaded file name, and user involved in the operation.

Severity: Minor

Type of Testing: System

Test Case Approach: Positive

Data - 42

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Prioritization

Feature description -

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.

Userstory title -

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

Acceptance criteria -

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.', 'Priority changes are logged and tracked.']

Generated Test Cases:

Here are 5 test cases for the given input data, each meticulously crafted to cover one or more acceptance criteria:

Test Case 1

Test Case ID: TC-TASKPRIO-001

Test Case Title: Verify that users can set and adjust task priorities

Test Case Description: This test case confirms that users can assign priority levels (e.g., High, Medium, Low) to tasks and modify these priorities as needed within the task management tool.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- A new task is created

Test Data:

Task details with varying priority levels (High, Medium, Low)

Test Steps:

1. Navigate to the task list.
2. Select a newly created task.
3. Access the task priority setting (dropdown menu, radio buttons, etc.).
4. Change the priority from the default setting to High.
5. Change the priority back to Medium, then to Low.

Postconditions:

- Task priority is reflected in the task list view.

Expected Result: Users can sucessfully set and adjust task priorities without encountering errors.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TC-TASKPRIO-002

Test Case Title: Verify that tasks can be sorted and filtered based on priority levels

Test Case Description: This test case ensures that the system allows users to sort and filter tasks based on their assigned priority levels, providing a clear overview of tasks requiring immediate attention.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Multiple tasks with different priority levels exist

Test Data:

Tasks with varying priority levels (High, Medium, Low)

Test Steps:

1. Go to the task list view.
2. Sort the tasks by priority level (descending, ascending). Verify the task order reflects the priority levels.
3. Apply a filter to display only High-priority tasks.

Postconditions:

- Task list is sorted correctly by priority.

Expected Result: Users can effectively sort and filter tasks by priority level, providing a focused view of critical tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TC-TASKPRIO-003

Test Case Title: Verify the system provides visual indicators of task priority

Test Case Description: This test case checks if the system visually distinguishes tasks based on their assigned priority levels using different colors, icons or other visual cues.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Tasks with various priority levels are present

Test Data:

Tasks with different priority levels (High, Medium, Low)

Test Steps:

1. View the task list.
2. Observe the visual indicators assigned to tasks with different priorities.

Postconditions:

- All tasks display distinct visual indicators representing their priority levels.

Expected Result: The system consistently employs visual cues to clearly indicate the priority of each task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: TC-TASKPRIO-004

Test Case Title: Verify notification system for high-priority tasks

Test Case Description: This test case ensures notifications are triggered for high-priority tasks, informing users of urgent items requiring attention.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- A high-priority task with an approaching deadline exists.
- User has notification preferences enabled for task updates.

Test Data:

- A high-priority task

Test Steps:

1. Set up a notification system integration (email, in-app, etc.).
2. Create a high-priority task with a near-future deadline.
3. Observe notification delivery based on the task's prioritizing level.

Postconditions:

- A notification is received for the high-priority task.

Expected Result: The notification system promptly alerts the user about high-priority tasks, ensuring timely action.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TC-TASKPRIO-005

Test Case Title: Verify logging and tracking of priority changes

Test Case Description: This test case confirms that the system accurately logs and tracks any changes made to a task's priority level.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in

Test Data:

- A task

Test Steps:

1. Create a task and assign it an initial priority level (e.g., Medium).
2. Change the task's priority level to High.
3. Within the task details or audit log, review the history of priority changes.

Postconditions:

- A record of the priority change from Medium to High is documented.

Expected Result: The software maintains a clear history of all priority changes made to tasks, enabling users to track modifications and understand task prioritization evolution.

Severity: Moderate

Type of Testing: Functional

Test Case Approach: Positive

Let me know if you need further clarification on any of these test cases or require additional test cases for this project!

Data - 43

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

Userstory title -

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay updated on task progress.

Acceptance criteria -

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

Generated Test Cases:

Here are 5 test cases for the provided user story and acceptance criteria:

Test Case 1

Test Case ID: TCI123

Test Case Title: Verify that users can assign tasks to team members.

Test Case Description: This test case verifies the functionality of assigning tasks to team members within the Task Management Tool. It will check if a user can successfully select a team member and assign a task to them, ensuring the task is reflected in the assigned member's task list.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:
- User is logged in.
- A team has been created with multiple members.
- A new task has been created and is open for assignment.

Test Data: A list of team members.

Test Steps:

1. Navigate to the task details page.
2. Click on the "Assign Task" button or similar.
3. Select the desired team member from the dropdown list.
4. Confirm the task assignment.

Postconditions:

- The chosen team member now appears as the assignee for the task.
- The task is reflected in the team member's assigned task list.
- Any related notifications are sent to the assignee.

Expected Result: The system successfully assigns the task to the selected team member, updating both the task details page and the assignee's task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TCI456

Test Case Title: Verify that users can share updates and communicate through comments or messages on tasks.

Test Case Description: This test case verifies the functionality for sharing updates and communicating through comments or messages on a task. It will ensure users can create, view, and respond to these updates effectively, fostering clear communication on tasks.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- A task has been created.
- User is logged in.
- At least two users need to be involved (as a creator and a commenter/responder).

Test Data: Predefined task content, comment text.

Test Steps:

1. Navigate to the task details page.
2. Add a comment or update related to the task's progress.
3. Verify that the comment/update is visible to other team members who have access to the task.
4. Have another user respond to the comment.
5. Verify that the response is reflected in the task details.

Postconditions:

- All comments/updates for the task are recorded in the discussion thread.
- Notification system properly alerts users about new comments.

Expected Result: Users can successfully add comments/updates, and these are clearly displayed and tracked within the task. Responsive communication is fostered with appropriate notifications.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TCI789

Test Case Title: Verify that task updates and comments are visible to assigned team members.

Test Case Description: This test case checks if task updates and comments added by users are appropriately visible to the assigned team members. It ensures information transparency and keeps assigned users informed about task progress and discussions.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:
- A task has been assigned to a specific team member.

- User is logged in as the task creator or another user with viewing permissions.

Test Data: A task with assigned team member, predefined updates/comments.

Test Steps:

1. Navigate to the task details page.
2. As a different user or the task creator, add a comment or update to the task.
3. Access the same task as the assigned team member.
4. Verify that the added comment or update is visible on the task details page.

Postconditions: Post

Expected Result: Task updates and comments are clearly visible to the assigned team member, promoting effective collaboration and awareness of task progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: TC1011

Test Case Title: Verify that users receive notifications for task updates and comments.

Test Case Description: This test case verifies the notification system's functionality for delivering alerts about task updates and comments. It ensures that team members are promptly informed about relevant changes, promoting timely responsiveness and task management.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:
- A user is logged in and assigned to a task.
- A separate user with access to the same task will make an update or comment.
- The notification settings for the assigned user are enabled.

Test Data: Different types of task updates and comments.

Test Steps:
1. Create a task and assign it to a specific user.
2. Have another user add a comment or update to the task.
3. Verify that the assigned user receives a notification alert about the change.
4. Check if the notification details accurately reflect the update or comment made.

Postconditions:
- Notification preferences of all involved users were appropriately considered.

Expected Result: The notification system effectively delivers timely alerts to assigned users whenever someone adds an update or comment to a task they are assigned to, enhancing collaboration and responsiveness.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TC5678

Test Case Title: Verify that the system provides a clear view of task collaboration history.

Test Case Description: This test case verifies that the task details page provides a well-structured and understandable history of all collaborations related to the task. It ensures that users can easily track who made updates, comments, and task assignments, promoting transparency and accountability.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- A task has been created and has received multiple updates, comments, and assignments.
- User is logged in with permissions to view the task history.

Test Data: Multiple updates, comments, and assignments made by different users on the same task.

Test Steps:

1. Open the task details page.
2. Locate the section displaying the task collaboration history.
3. Verify that the history includes timestamps for each update, comment, and assignment.
4. Confirm that each action is clearly attributed to the user who performed it.
5. Check if the history is chronologically ordered and easily navigable.

Postconditions:

- Task history is updated with each new collaboration instance.

Expected Result: The system provides a clear and concise history of all collaborative actions undertaken on the task, enabling users to easily track progress, accountability, and past discussions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 44

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Tracking

Feature description -

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

Userstory title -

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

Acceptance criteria -

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.', 'System provides visual indicators for task progress.', '']

Generated Test Cases:

Here are 5 test cases for the provided input data, following the test case template you've outlined:

Test Case 1:

Test Case ID: taskTracking01

Test Case Title: Verify that users can view task statuses

Test Case Description: This test case verifies the ability of a user to view the current status of a task within the system. It assesses the display of status information accurately and consistently.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- A task has been created and assigned to a user.
- The user is logged in.

Test Data: Task with a defined status (e.g., To Do, In Progress, Completed)

Test Steps:

1. Navigate to the task list.
2. Select the created task.
3. Observe the displayed status of the task.

Postconditions:

- The task details page is open.

Expected Result: The task status is clearly displayed and reflects its current state (To Do, In Progress, Completed).

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: taskTracking02

Test Case Title: Verify that users can track task progress and deadlines

Test Case Description: This test case verifies the functionality of tracking task progress and corresponding deadlines within the system. It ensures users can monitor task advancement and approaching due dates.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- A task has been created with a designated deadline.

- The user is logged in and has access to the task.

Test Data: Task with a set deadline, progress percentage values.

Test Steps:

1. Go to the task details page.

2. Observe the progress bar and its percentage representation.

3. Note the displayed deadline date and time.

Postconditions:

- The task details page remains open.

Expected Result: The progress bar accurately reflects the current task progress, and the deadline is clearly presented.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: taskTracking03

Test Case Title: Verify that tasks are updated in real-time within the system

Test Case Description: This test case validates the real-time task updating functionality. It confirms that changes in task status or progress are immediately reflected across the system for all relevant users.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- Two users are logged in, one user creates a task.
- The task has a defined initial status (e.g., To Do).

Test Data: Task details, user accounts.

Test Steps:

1. User A creates a task and sets its initial status to "To Do."
2. User B, simultaneously, accesses the task list.
3. User A updates the task status to "In Progress."
4. User B observes the updated task status.

Postconditions:

- Both users remain logged in.

Expected Result: User B immediately sees the change in task status from "To Do" to "In Progress."

Severity: Major

Type of Testing: Performance, Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: taskTracking04

Test Case Title: Verify visual indicators for task progress are displayed

Test Case Description: This test case ensures that the system provides clear visual cues to indicate the progress of a task. This includes confirming the correct display of progress bars, color coding, or other visual representations.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- A task exists with a defined percentage of completion (e.g., 50% complete).

- The user is logged in and has access to the task.

Test Data: Task with a progress percentage.

Test Steps:

1. Open the task details page.

2. Inspect the visual indicator of progress (progress bar, color coding, etc.).

Postconditions:

- The task details page remains open.

Expected Result: The visual indicator accurately reflects the task progress of 50%.

Severity: Major

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case 5:

Test Case ID: taskTracking05

Test Case Title: Verify delayed task updates under unstable network conditions

Test Case Description: This test case simulates unreliable network connectivity to determine whether the system can handle delayed updates and display the correct status information once the connection is restored.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- A task exists, user is logged in

Test Data: Simulated unstable network connection

Test Steps:

1. Create a task and set its initial status

2. Disrupt the network connection (simulated)

3. Update the task status while offline

4. Restore the stable network connection

5. Observe the task status after the connection is restored

Postconditions:

- Network connection is re-established.

Expected Result: The task status reflects the latest update (from Step 3) even though it was made while offline.

Severity: Major

Type of Testing: Performance

Test Case Approach: Negative

Data - 45

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Notifications

Feature description -

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

Userstory title -

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

Acceptance criteria -

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.', 'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', '']

Generated Test Cases:

Here are 5 test cases for the Task Notifications feature of the Task Management Tool, following the provided template:

Test Case 1

Test Case ID: tmn_001

Test Case Title: Verify that users receive notifications for new tasks

Test Case Description: This test case ensures that users receive notifications when a new task is assigned to them or added to a project they are involved in.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- A user account is created.

- A project is created.

- At least two users are added to the project.

Test Data: Sample task details (title, description, assignee, due date etc.)

Test Steps:

1. Log in as user 1.

2. Create a new task and assign it to user 2.

3. Log out of user 1's account.

4. Log in as user 2.

5. Verify that user 2 receives a notification about the new task assignment.

Postconditions:

- Notification received by user 2.

Expected Result: User 2 receives a notification about the newly assigned task.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2

Test Case ID: tmn_002

Test Case Title: Verify that users receive notifications for task updates

Test Case Description: This test case ensures that users receive notifications when the status, description, or due date of a task they are assigned to or involved in is updated.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- A user account is created.

- A task is created with a user assigned to it.

Test Data: Modified task details (updated status, description, or due date)

Test Steps:

1. Log in as the user assigned to the task.

2. Update the task's status, description, or due date.

3. Log out of the user's account.

4. Log in as a different user involved in the same project.

5. Verify that the user receives a notification about the task update.

Postconditions:

- Notification received by the other user.

Expected Result: The other user receives a notification about the updated task details.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3

Test Case ID: tmn_003

Test Case Title: Verify that users receive notifications for comments on their tasks

Test Case Description: This test case verifies that users get notified when someone comments on a task they are assigned to or involved in.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- A user account is created.

- A task is created.

- At least two users are assigned to the task.

Test Data: Sample comment text

Test Steps:

1. Log in as user 1 (assigned to the task).

2. Log out of user 1's account.

3. Log in as user 2 (another user assigned to the task).

4. Add a comment to the task.

5. Log out of user 2's account.

6. Log back in as user 1.

7. Verify that user 1 receives a notification about the new comment.

Postconditions:

- Notification received by user 1.

Expected Result: User 1 receives a notification about the comment added by user 2.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4

Test Case ID: tmn_004

Test Case Title: Verify that reminders for upcoming deadlines are sent

Test Case Description: This test case ensures that users receive timely reminders before a task's deadline.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- A user account is created.

- A task is created with a future due date.

Test Data: Task details with a future due date (set at least a day in advance for testing)

Test Steps:

1. Log in as the user assigned to the task.

2. (Ensure notifications are enabled).

3. Log out of the user's account.

4. Wait until the reminder time for the upcoming deadline arrives.

5. Verify that the user receives a notification reminding them of the task's approaching deadline.

Postconditions:

- Reminder notification received.

Expected Result: The user receives a notification reminding them of the task's due date.

Severity: Critical

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5

Test Case ID: tmn_005

Test Case Title: Verify that users can customize notification settings

Test Case Description: This test case ensures that users can opt-in or opt-out of receiving different types of notifications.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- A user account is created.

Test Data:

List of notification types the user can customize (e.g., new tasks, task updates, comments, deadlines).

Test Steps:

1. Log in as a user.

2. Navigate to the user profile or settings area.

3. Locate the notification settings section.

4. Disable individual notification types (e.g., turn off notifications for task updates).

5. Perform actions that would typically trigger the disabled notifications (update a task, add a comment).

6. Verify that the user does **not** receive notifications for the disabled notification types.

Postconditions:

- Specific notification preferences applied.

Expected Result: Users can selectively disable notifications for different event types, controlling their notification frequency and relevance.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 46

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Categorization

Feature description -

Users can categorize tasks into different groups or projects for better organization and management. Task categorization aims to help users organize tasks by type, priority, or project.

Userstory title -

As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

Acceptance criteria -

['Users can create and assign tasks to different categories or projects.', 'Users can view tasks grouped by categories or projects.', 'Tasks can be moved between categories or projects.', 'Category or project information is accurately reflected in task views.', '']

Generated Test Cases:

Here are 5 test cases for the provided User Story and Acceptance Criteria:

Test Case 1

Test Case ID: tasks-cat-001

Test Case Title: Verify that users can create and assign tasks to different categories

Test Case Description: This test case aims to ensure that users can successfully create new categories or projects and assign tasks to these newly created categories.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

* User is logged in.

* A sample task exists.

Test Data:

* New category names (e.g., "Marketing," "Development," "Project X").

Test Steps:

1. Access the task creation form.
2. Select a newly created category.
3. Enter task details and save.

Postconditions:

* The task should be assigned to the selected category.

Expected Result: The task is successfully assigned to the chosen category, and this assignment is reflected in the task view and list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: tasks-cat-002

Test Case Title: Verify that users can view tasks grouped by categories

Test Case Description: This test case verifies that the system correctly displays tasks based on their assigned categories.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

* Multiple tasks exist, each assigned to a different category.

Test Data:

* Categories with assigned tasks (e.g., "Marketing" with 3 tasks, "Development" with 5 tasks)

Test Steps:

1. Navigate to the task list view.

2. Select a specific category from the list.

3. Confirm that only tasks assigned to that category are displayed.

Postconditions:

* The system displays tasks filtered by the selected category.

Expected Result: The task list accurately displays only tasks belonging to the chosen category, providing a filtered view for efficient task management.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: tasks-cat-003

Test Case Title: Verify that tasks can be moved between categories

Test Case Description: This test case confirms the ability to reassign tasks to different categories as needed, ensuring flexibility in task organization.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

* At least two distinct categories exist.

* A task exists.

Test Data:

* Two different category names (e.g., "Development", "Marketing").

Test Steps:

1. Locate the task to be moved.

2. Open the task details.

3. Select the "Edit" or "Update" option.

4. Choose the target category from the list.

5. Save the changes.

Postconditions:

* The task's assignment should be updated.

Expected Result: The task successfully changes categories, updating its categorization and display in both the task list and the selected category view.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: tasks-cat-004

Test Case Title: Verify that category or project information is accurately reflected in task views.

Test Case Description: This test case ensures that tasks display the correct category information consistently across various task views.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

* Tasks exist and are assigned to categories.

Test Data:

* A mix of tasks assigned to different categories.

Test Steps:

1. Access various task views, such as the "All Tasks" view, individual category views, and project views.

2. Verify that each task displays the assigned category name correctly.

Postconditions:

* Task views should correctly display category information

Expected Result: All task views accurately reflect the assigned categories for each task, maintaining data consistency and enabling users to quickly identify task affiliations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: tasks-cat-005

Test Case Title: Verify that users receive an error message if attempting to assign a task to a non-existent category

Test Case Description: This test case identifies the system's behavior if a user tries to assign a task to a category that doesn't exist.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

* User is logged in

Test Data:

* A task

* A category name that does not exist

Test Steps:

1. Locate the task to be assigned

2. Attempt to select the non-existent category during assignment.

3. Observe the displayed message

Postconditions:

* An error message is displayed

Expected Result: The system displays a clear error message indicating that the chosen category does not exist, preventing invalid assignments and guiding the user to create or select a valid category.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Negative

Data - 47

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Search

Feature description -

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

Userstory title -

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

Generated Test Cases:

Here are 5 test cases based on the provided information, each targeting a specific acceptance criterion:

Test Case 1:

Test Case ID: taskSearch_Name

Test Case Title: Verify that users can search for tasks by name.

Test Case Description: This test case aims to validate that users can accurately search for tasks within the application by entering a specific task name.

Test Suite: Task Search

Test Priority: High

Preconditions:

* A user account is logged in.

* At least 5 tasks are created with distinct names.

Test Data: A valid task name already existing in the system.

Test Steps:

1. Navigate to the task list view.

2. Enter the specific task name into the search bar.

3. Click on the search button.

4. Verify that the search results display the task with the matching name.

Postconditions:

* The search bar is cleared.

Expected Result: The search results accurately display the task matching the entered name.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: taskSearch_Description

Test Case Title: Verify that users can search for tasks by description.

Test Case Description: This test case aims to validate that users can accurately search for tasks within the application by entering a specific task description.

Test Suite: Task Search

Test Priority: High

Preconditions:

* A user account is logged in.

* At least 3 tasks are created with distinct descriptions.

Test Data: A valid task description already existing in the system.

Test Steps:

1. Navigate to the task list view.

2. Enter the specific task description into the search bar.

3. Click on the search button.

4. Verify that the search results display the task with the matching description.

Postconditions:

* The search bar is cleared.

Expected Result: The search results accurately display the task matching the entered description.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: taskSearch_Category

Test Case Title: Verify that users can filter search results by category.

Test Case Description: This test case aims to validate that users can filter search results by category accurately, narrowing down the displayed tasks.

Test Suite: Task Search

Test Priority: High

Preconditions:

* A user account is logged in.

* At least 5 tasks are created, categorized into 2 distinct categories.

Test Data: One of the defined categories.

Test Steps:

1. Navigate to the task list view.

2. In the search bar, focus on the category filter.

3. Select the desired category from the available options.

4. Verify that the search results display only tasks belonging to the selected category.

Postconditions:

* The search bar is cleared.

Expected Result: The search results are filtered correctly to display only tasks belonging to the selected category.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: taskSearch_Priority

Test Case Title: Verify that users can filter search results by priority.

Test Case Description: This test case aims to validate that users can filter search results by priority accurately, displaying task based on their importance level.

Test Suite: Task Search

Test Priority: High

Preconditions:

* A user account is logged in.

* At least 5 tasks are created with varying priority levels (Low, Medium, High).

Test Data: One of the defined priority levels (e.g., High).

Test Steps:

1. Navigate to the task list view.

2. In the search bar, focus on the priority filter.

3. Select the desired priority level from the available options.

4. Verify that the search results display only tasks with the selected priority level.

Postconditions:

* The search bar is cleared.

Expected Result: The search results are filtered correctly to display only tasks with the selected priority level.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: taskSearch_DueDate

Test Case Title: Verify that users can filter search results by due date.

Test Case Description: This test case aims to validate that users can filter search results by due date accurately, displaying tasks nearing their deadlines or within a specified date range.

Test Suite: Task Search

Test Priority: High

Preconditions:

* A user account is logged in.

* At least 5 tasks are created with varying due dates.

Test Data: A specific date range within which tasks must be due.

Test Steps:

1. Navigate to the task list view.

2. In the search bar, focus on the due date filter.

3. Select the desired date range from the available options.

4. Verify that the search results display only tasks with due dates within the selected range.

Postconditions:

* The search bar is cleared.

Expected Result: The search results are filtered correctly to display only tasks with due dates within the selected range.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 48

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks by adding comments, attaching files, and sharing tasks with other users. Task collaboration aims to enhance team communication and productivity.

Userstory title -

As a user, I want to collaborate on tasks by adding comments, attaching files, and sharing tasks with other users so that I can work more effectively with my team.

Acceptance criteria -

['Users can add comments to tasks.', 'Users can attach files to tasks.', 'Users can share tasks with other users.', 'Task collaboration features work seamlessly and enhance team communication.', "]

Generated Test Cases:

Here are 5 test cases for the provided User Story and Acceptance Criteria:

Test Case 1

Test Case ID: TMT-COL-001

Test Case Title: Verify that users can add comments to tasks.

Test Case Description: This test case verifies if users can successfully add comments to existing tasks within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* A task exists within the system with details like title, description, and assignee.

Test Data: Sample comment text (e.g., "Great work on this, almost finished!")

Test Steps:

1. Navigate to the task page.

2. Click on the "Add Comment" button.

3. Enter the sample comment text into the input field.

4. Click the "Post Comment" button.

Postconditions:

* The comment is displayed below the task details.

Expected Result: A new comment is successfully added to the task, visible to all collaborators.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TMT-COL-002

Test Case Title: Verify that users can attach files to tasks.

Test Case Description: This test case confirms the functionality of attaching files to tasks within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* A task exists within the system with details like title, description, and assignee.

* A local file (e.g., a document or image) is prepared for attachment.

Test Data: Local file (e.g., doc.txt, image.jpg)

Test Steps:

1. Navigate to the task page.

2. Click on the "Attach File" button.

3. Select the prepared local file from the system.

4. Click the "Upload" button.

Postconditions:

* The uploaded file name is displayed next to the task details.

Expected Result: The chosen file is successfully uploaded and linked to the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TMT-COL-003

Test Case Title: Verify that users can share tasks with other users.

Test Case Description: This test case examines the ability to share tasks with other users within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* User 1 is logged into the Task Management Tool.
* User 2 is logged into the Task Management Tool.
* A task exists with details like title, description, and assignee.

Test Data: User credentials for User 2

Test Steps:

1. User 1 navigates to the task page.
2. User 1 clicks on the "Share Task" button.
3. Enter User 2's username or email address in the designated field.
4. Choose the desired permission level (e.g., View Only, Edit, Full Access)
5. Click the "Share" button.

Postconditions:

* User 2 receives a notification about being added to the task.

Expected Result: User 2 is successfully added to the task with the selected permission level, and notified.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: TMT-COL-004

Test Case Title: Verify seamless task collaboration without errors.

Test Case Description: This test case assesses the overall functionality and user experience of task collaboration features.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* Two or more users are logged into the Task Management Tool.

* Multiple tasks exist in the system.

Test Data:

* Example task: "Finalize Project Proposal" with details like due date and assigned user.

* Sample comments, files (documents, images), and shared tasks for various tasks.

Test Steps:

1. Users collaborate on different tasks by adding comments, attaching files, and sharing tasks.

2. Monitor task updates and user notifications throughout the collaboration process.

3. Verify that comments, files, and shared tasks are displayed correctly and are accessible to intended collaborators.

Postconditions:
* Tasks remain updated with all changes and interactions.
* All users receive corresponding notifications.

Expected Result: Task collaboration functions seamlessly without errors, ensuring user interaction is smooth and efficient.
Severity: Major
Type of Testing: Usability and Functional
Test Case Approach: Positive

Test Case 5

Test Case ID: TMT-COL-005
Test Case Title: Verify handling of invalid file types for task attachment.
Test Case Description: This test case evaluates the system's response to attempts to upload invalid file types for task attachment.
Test Suite: Task Collaboration
Test Priority: Medium
Preconditions:
* User is logged into the Task Management Tool.
* A task exists within the system.

Test Data:

* Invalid file types (e.g., .exe, .php, .bat) that are generally not permitted.

Test Steps:

1. Navigate to the task page.

2. Attempt to select and upload an invalid file type from the system.

Postconditions:

* User receives a clear error message.

Expected Result: The system prevents the upload of invalid file types, displaying an appropriate error message to the user. The task remains intact without unintended file associations.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Negative

Data - 49

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Course Creation Tools

Feature description -

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to structure content effectively, ensuring an engaging and informative learning experience for students.

Userstory title -

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

Acceptance criteria -

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

Generated Test Cases:

Test Case Generation for Online Learning Platform - Course Creation Tools

Here are 5 detailed test cases for the provided user story and acceptance criteria:

Test Case 1

* Test Case ID: OLP-CC-001

* Test Case Title: Verify that instructors can create courses using videos, presentations, quizzes, and assignments.

* Test Case Description: This test case verifies the functionality of adding various content types, including videos, presentations, quizzes, and assignments, to a new course.

* Test Suite: Course Creation Tools

* Test Priority: High

* Preconditions:
    * Instructor is logged in to the platform.
    * Access to course creation section is granted.
* Test Data:
    * Pre-recorded video file
    * Presentation file (e.g., .pptx)
    * Quiz questions with multiple choice and true/false options
    * Assignment document
* Test Steps:
    1. Navigate to the course creation section.
    2. Click on "Create New Course" button.
    3. Upload the video file and set its description.
    4. Upload the presentation file and set its title.
    5. Create a new quiz with multiple choice and true/false questions.
    6. Create a new assignment with clear instructions.
* Postconditions:
    * Added multimedia content appears in the course preview.
* Expected Result: The system successfully adds the video, presentation, quiz, and assignment to the course, and they are displayed in the course preview.

Test Case 2

* Test Case ID: OLP-CC-002
* Test Case Title: Verify user-friendliness of course creation tools for instructors with minimal technical skills
* Test Case Description: This test case evaluates the ease of use for instructors with limited technical expertise when utilizing the course creation tools.
* Test Suite: Course Creation Tools

* Test Priority: Medium

* Preconditions:

    * Instructor user with minimal technical experience is logged in.

* Test Data:

    * Sample course outline with learning objectives, topics, and content types.

* Test Steps:

    1. Navigate to the course creation section.

    2. Click on "Create New Course" button.

    3. Follow the guided steps to input course information based on the sample outline.

    4. Attempt to add various content types (video, presentation, quiz, etc.) using the provided interface.

    5. Review and edit any added content.

* Postconditions:

    * Course structure reflects the provided outline.

* Expected Result: The instructor can easily navigate through the course creation interface, understand the functionalities, and efficiently add content without encountering technical difficulties.

Test Case 3

* Test Case ID: OLP-CC-003

* Test Case Title: Verify the functionality of course preview before publishing.

* Test Case Description: This test case ensures that instructors can preview their created courses before making them live for students.

* Test Suite: Course Creation Tools

* Test Priority: High

* Preconditions:

    * Instructor is logged in to the platform.

    * A partially completed course is saved.

* Test Data:
    * Added content to the course (videos, presentations, quizzes, assignments).
* Test Steps:
    1. Navigate to the course creation section.
    2. Select the partially completed course.
    3. Click the "Preview Course" button.
    4. Navigate through different sections and content within the preview.
* Postconditions:
    * Course preview displays the content in a user-friendly format.
* Expected Result: The preview accurately reflects the course structure, content, and navigation as intended by the instructor.

Test Case 4

* Test Case ID: OLP-CC-004
* Test Case Title: Verify the ability to save courses as drafts for later editing.
* Test Case Description: This test case ensures that instructors can save their course in a draft state for future modifications.
* Test Suite: Course Creation Tools
* Test Priority: Medium
* Preconditions:
    * Instructor is logged in to the platform.
* Test Data:
    * Partially completed course content with video, presentation, and assignment added.
* Test Steps:
    1. Add content to the course.
    2. Click the "Save as Draft" button instead of completing the publishing process.

* Postconditions:
    * The course appears in the "Drafts" section.
* Expected Result: The system successfully saves the course as a draft and stores it for future editing, preserving the instructor's progress.

Test Case 5

* Test Case ID: OLP-CC-005
* Test Case Title: Verify the functionality of editing published courses.
* Test Case Description: This test case evaluates the ability to make changes to a course that has already been published.
* Test Suite: Course Creation Tools
* Test Priority: High
* Preconditions:
    * Instructor has published a course.
* Test Data:
    * Minor changes to existing content, such as updating quiz questions or modifying presentation slides.
* Test Steps:
    1. Navigate to the published course details.
    2. Click the "Edit Course" button.
    3. Make desired changes to the existing content according to the test data.
* Postconditions:
    * Changes are reflected in the updated course.
* Expected Result: The system allows editing published courses by providing access to the course content, enabling modifications and saving them to the updated version.

These test cases aim to comprehensively cover the acceptance criteria for the "Course Creation Tools" feature and ensure the development of a user-friendly and effective platform for instructors to create engaging online learning experiences.

--

Data - 50

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Interactive Lessons

Feature description -

Lessons are thoughtfully designed to foster learner engagement through interactive components such as quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these interactive elements enhance comprehension and retention of course material.

Userstory title -

As a learner, I want interactive lessons so that I can engage with the content more effectively.

Acceptance criteria -

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided after each interactive component.', 'Interactive lessons can be accessed on various devices.']

Generated Test Cases:

Here are 5 test cases designed to cover the acceptance criteria for the "Interactive Lessons" feature in an online learning platform:

Test Case 1

* **Test Case ID:** OL-INT-001
* **Test Case Title:** Verify that lessons include quizzes, polls, simulations, and discussions
* **Test Case Description:** This test case aims to validate the presence and functionality of all four types of interactive elements within a sample lesson.
* **Test Suite:** Interactive Lessons
* **Test Priority:** High
* **Preconditions:**
    * User is logged in to the platform.
    * A sample lesson with interactive elements is available.
* **Test Data:** N/A
* **Test Steps:**
    1. Navigate to the chosen sample lesson.
    2. Review the lesson content for the presence of quizzes, polls, simulations, and discussions.
    3. Attempt to engage with each interactive element by answering quiz questions, participating in polls, interacting with simulations, and contributing to the discussion forum.
* **Postconditions:**
    * User successfully engages with each interactive element.
* **Expected Result:** All four types of interactive elements (quizzes, polls, simulations, and discussions) are present and functional within the lesson.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** OL-INT-002
* **Test Case Title:** Verify seamless integration of interactive elements within lesson content
* **Test Case Description:** This test verifies that the interactive elements are incorporated smoothly into the lesson flow and do not disrupt the learning experience.
* **Test Suite:** Interactive Lessons
* **Test Priority:** High
* **Preconditions:**
    * User is logged in.
    * A lesson with interactive elements is accessed.
* **Test Data:** N/A
* **Test Steps:**
    1. Begin the chosen lesson and observe the natural integration of interactive elements.
    2. Manually navigate through both text content and interactive components to assess flow and navigation.
    3. Analyze the visual cues and design elements to determine if interactive elements blend seamlessly with the lesson structure.
* **Postconditions:**
    * User navigates through the lesson without encountering disruptions or confusing transitions.
* **Expected Result:** Interactive elements are seamlessly integrated into the lesson content, enhancing user engagement without disrupting the learning process.
* **Severity:** Medium
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** OL-INT-003
* **Test Case Title:** Verify learner progress tracking within interactive lesson activities
* **Test Case Description:** This test confirms that the platform accurately tracks learner progress through interactive activities within a lesson.
* **Test Suite:** Interactive Lessons
* **Test Priority:** High
* **Preconditions:**
    * User is logged in.
    * A lesson with interactive elements is accessed.
* **Test Data:** N/A
* **Test Steps:**
    1. Complete various interactive activities within the lesson (e.g., quiz questions, poll responses, simulation interactions).
    2. Review the learner dashboard or progress tracker to confirm completion status for each activity.
    * Note any scores or feedback received for quiz or simulation activities.
* **Postconditions:**
    * Progress information is correctly displayed on the dashboard.
* **Expected Result:** The platform accurately tracks the learner's progress through all interactive activities, displaying completion status and relevant feedback.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** OL-INT-004
* **Test Case Title:** Verify feedback provision after each interactive component
* **Test Case Description:** This test verifies that learners receive timely and informative feedback upon completion of each interactive component.
* **Test Suite:** Interactive Lessons
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged in.
    * A lesson with interactive elements is accessed.
* **Test Data:** N/A
* **Test Steps:**
    1. Attempt a quiz question within the lesson.
    2. Review the feedback provided after submitting the answer (correct/incorrect, explanation).
    3. Attempt a poll or simulation activity.
    4. Observe and analyze any feedback provided for the outcome of these activities.
* **Postconditions:**
    * User receives feedback for each interactive component completed.
* **Expected Result:** Clear and constructive feedback is provided for every interactive component, informing the learner about their performance and aiding in comprehension.
* **Severity:** Minor
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** OL-INT-005
* **Test Case Title:** Verify interactive lesson accessibility on various devices

* **Test Case Description:** This test ensures that interactive lessons function correctly and provide an optimal user experience across different devices (desktop, tablet, mobile).
* **Test Suite:** Interactive Lessons
* **Test Priority:** Medium
* **Preconditions:**
    * A sample lesson with interactive elements is available.
    * User accounts are created.
* **Test Data:** N/A
* **Test Steps:**
    1. Access the sample lesson from a desktop computer, ensuring all interactive elements function smoothly.
    2. Access the same lesson from a tablet device, verifying the responsiveness and accessibility of interactive components.
    3. Access the lesson from a mobile phone, confirming a user-friendly mobile interface and proper functionality of interactive elements.
* **Postconditions:**
    * Interactive lesson is accessible on all tested devices.
* **Expected Result:** The interactive lesson maintains its functionality and user-friendliness across various devices without compromising the learning experience.
* **Severity:** Major
* **Type of Testing:** Compatibility Testing, Functional Testing
* **Test Case Approach:** Positive

Data - 51

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Personalized Learning Paths

Feature description -

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

Userstory title -

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

Acceptance criteria -

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

Generated Test Cases:

Here are 5 test cases for the Personalized Learning Paths feature of the Online Learning Platform:

Test Case 1

* **Test Case ID:** LP_001

* **Test Case Title:** Verify Learners Can Select Courses Based on Interests

* **Test Case Description:** This test case verifies that learners can browse through available courses and select courses aligned with their interests.

* **Test Suite:** Personalized Learning Paths

* **Test Priority:** High

* **Preconditions:**

    * Learner account is created and logged in.

    * A variety of courses across different subjects are available on the platform.

* **Test Data:** List of courses categorized by subject areas (e.g., programming, history, art)

* **Test Steps:**

    1. Navigate to the "Courses" section of the platform.

    2. Browse through the different course categories.

    3. Select a course from a category that aligns with the learner's stated interests.

    4. Verify that the selected course is added to the learner's personalized path.

* **Postconditions:**

    * Course is added to the learner's "My Path" section.

* **Expected Result:** The learner is able to successfully select courses based on their designated interests and these selections are reflected in their personalized learning path.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** LP_002

* **Test Case Title:** Verify Dynamic Adjustment of Learning Paths

* **Test Case Description:** This test case ensures that the personalized learning path adapts based on the learner's progress and completion of modules.

* **Test Suite:** Personalized Learning Paths

* **Test Priority:** High

* **Preconditions:**

    * Learner account is created and logged in.

    * A course with multiple modules has been added to the learner's path.

* **Test Data:** Course with varied modules, some completed, some incomplete

* **Test Steps:**

    1. Access the learner's personalized path.

    2. Complete the first two modules of the chosen course.

    3. Refresh the path and observe the order and recommendations.

* **Postconditions:**

    * Module completion updates are reflected in the path.

* **Expected Result:** The system automatically adjusts the learning path, potentially moving on to related modules based on the learner's progress, and provides relevant recommendations for further learning.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive, Behavioral

Test Case 3

* **Test Case ID:** LP_003

* **Test Case Title:** Verify System Provides Course Recommendations

* **Test Case Description:** This test case checks if the system offers personalized recommendations for new courses or modules based on the learner's chosen path and interests.
* **Test Suite:** Personalized Learning Paths
* **Test Priority:** High
* **Preconditions:**
    * Learner account is created and logged in.
    * The learner has progressed through a portion of their chosen learning path.
* **Test Data:** Data on learner's progress, interests recorded.
* **Test Steps:**
    1. Access the learner's personalized path.
    2. Observe any "Recommended Courses" or "Suggested Modules" sections.
    3. Click on one of the recommendations and review its description.
* **Postconditions:**
    * Learner has viewed a recommendation.
* **Expected Result:** The system provides relevant course or module recommendations tailored to the learner's current path and expressed interests.

* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive,

Test Case 4

* **Test Case ID:** LP_004
* **Test Case Title:** Verify Progress Tracking is Personalized
* **Test Case Description:** This test case ensures that the progress tracking accurately reflects the learner's individual journey through their personalized path.

* **Test Suite:** Personalized Learning Paths
* **Test Priority:** Medium
* **Preconditions:**
    * Learner account is created and logged in.
    * A course with multiple modules is selected.
* **Test Data:** Data on module completion status within the chosen course
* **Test Steps:**
    1. Navigate to the learner's personalized path.
    2. Check the progress tracking for each module within the chosen course.
    3. Complete modules and observe updates in the progress tracking.
* **Postconditions:**
    * Module completion dates are recorded.
* **Expected Result:** The progress tracker accurately shows the learner's completion status for each module within their personalized path, allowing them to monitor their advancement.

* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive,

Test Case 5

* **Test Case ID:** LP_005
* **Test Case Title:** Verify Learner Can Adjust Learning Path
* **Test Case Description:** This test case verifies if learners can adjust their learning path by adding, removing, or rearranging courses or modules.
* **Test Suite:** Personalized Learning Paths

* **Test Priority:** Medium

* **Preconditions:**

    * Learner account is created and logged in.

    * A personalized learning path with multiple courses is established.

* **Test Data:** Data on the current learner's personalized path

* **Test Steps:**

    1. Navigate to the learner's personalized path.

    2. Attempt to add a new course to the path.

    3. Attempt to remove an existing course from the path.

    4. Attempt to reorder the courses within the path.

* **Postconditions:**

    * The learner's path has been modified.

* **Expected Result:** Learners can make changes to their personalized learning path dynamically, adding, removing, or rearranging courses and modules according to their evolving preferences and goals.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Data - 52

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Progress Tracking

Feature description -

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

Userstory title -

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Acceptance criteria -

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

Generated Test Cases:

Here are 5 test cases for the Online Learning Platform's Progress Tracking feature based on the provided acceptance criteria:

Test Case 1

* **Test Case ID:** OLTP-PT-001

* **Test Case Title:** Verify that progress tracking shows completed activities and grades.

* **Test Case Description:** This test case verifies the functionality of the progress tracker to display the status of completed activities and associated grades for a learner.
* **Test Suite:** Progress Tracking
* **Test Priority:** High
* **Preconditions:**
    * Learner is registered and enrolled in at least one course.
    * Course contains multiple activities and assignments with associated grades.
* **Test Data:** Course content with completed activities and assignments.
* **Test Steps:**
    1. Log in as a learner.
    2. Navigate to the chosen course.
    3. Access the progress tracking section.
    4. Verify that completed activities are clearly marked.
    5. Ensure that corresponding grades for each completed activity or assignment are displayed accurately.
* **Postconditions:**
    * Learner returns to course overview.
* **Expected Result:** The progress tracker displays a list of all completed activities and their associated grades accurately.
* **Severity:** Critical
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** OLTP-PT-002
* **Test Case Title:** Verify learners can view detailed reports on their progress.
* **Test Case Description:** This test case ensures learners can access detailed progress reports that provide

comprehensive insights into their learning journey within a course.

* **Test Suite:** Progress Tracking

* **Test Priority:** High

* **Preconditions:**

    * Learner is registered and enrolled in a course.

* **Test Data:** Course content with some completed activities and assignments.

* **Test Steps:**

    1. Log in as a learner.

    2. Navigate to the chosen course.

    3. Access the progress tracking section.

    4. Select the option to view a detailed progress report.

    5. Analyze the report for information on completed modules, activities, assignments, and overall progress percentage.

* **Postconditions:**

    * Learner returns to the course overview.

* **Expected Result:** The detailed report accurately reflects the learner?s completion status, grades, and overall progress in the course.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** OLTP-PT-003

* **Test Case Title:** Verify achievements are highlighted in the progress tracker.

* **Test Case Description:** This test case confirms that the system effectively highlights achievements, signifying milestones reached by learners, within the progress tracker.

* **Test Suite:** Progress Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * Learner is enrolled in a course with achievements defined. (e.g., completing a module, passing a quiz)

    * Learner accomplishes a course activity that triggers an achievement.

* **Test Data:** Course content with achievements incorporated.

* **Test Steps:**

    1. Log in as a learner.

    2. Navigate to the chosen course.

    3. Access the progress tracking section.

    4. Verify that the achieved milestone or achievement is visually highlighted in the progress tracker.

* **Postconditions:**

    * Learner remains within the progress tracking section.

* **Expected Result:** The achieved milestone or achievement is visually distinct (e.g., using color, icon) to indicate its completion.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** OLTP-PT-004

* **Test Case Title:** Verify learners receive notifications about their progress.

* **Test Case Description:** This test case tests the functionality of notifications that inform learners about their progress, such as completing a module or achieving a milestone.

* **Test Suite:** Progress Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * Learner account settings allow for progress notifications.

    * Learner completes a course activity that triggers a notification.

* **Test Data:** Course content with a definable progress trigger for a notification.

* **Test Steps:**

    1. Log in as a learner.

    2. Complete a specific course activity (e.g., finish a module) that triggers a notification.

    3. Check the notification interface for a notification related to the completed activity.

    4. Verify the notification content accurately reflects the progress made.

* **Postconditions:**

    * Learner returns to the course.

* **Expected Result:** The learner receives a timely notification regarding their progress, detailing the specific achievement.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** OLTP-PT-005

* **Test Case Title:** Verify progress tracking can be exported as reports.

* **Test Case Description:** This test verifies the functionality of exporting progress tracking data as reports for learners and instructors.

* **Test Suite:** Progress Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * Learner is enrolled in a course.

    * Learner has completed some activities and assignments.

* **Test Data:** Course content with completed activities and assignments.

* **Test Steps:**

    1. Log in as a learner.

    2. Access the progress tracking section.

    3. Locate and select the option to export the progress data.

    4. Choose a desired format for the report (e.g., PDF, CSV).

    5. Examine the exported report for accuracy in reflecting the learner's progress information.

* **Postconditions:**

    * Learner has a downloaded progress report.

* **Expected Result:** The system successfully exports a comprehensive report containing detailed progress information in the chosen format.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Data - 53

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Assessment and Grading

Feature description -

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

Userstory title -

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

Acceptance criteria -

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

Generated Test Cases:

Here are 5 test cases detailing the features for the Online Learning Platform's Assessment and Grading system:

Test Case 1

Test Case ID: LRN_001

Test Case Title: Verify that instructors can create quizzes

Test Case Description: This test case aims to confirm that instructors can successfully create quizzes with multiple-choice questions within the online learning platform.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Instructor is logged in.

* A course is available for the instructor to manage.

Test Data: Course content, quiz title, multiple-choice questions with options.

Test Steps:

1. Navigate to the course dashboard.

2. Click on the "Create Assessment" button.

3. Select "Quiz" from the assessment type options.

4. Enter a title for the quiz.

5. Add multiple-choice questions with four answer options each.

6. Save the quiz.

Postconditions:

* A new quiz is added to the course.

* Quiz contains multiple-choice questions as defined.

Expected Result: The system allows instructors to successfully create and save quizzes containing multiple-choice questions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: LRN_002

Test Case Title: Verify that instructors can create assignments

Test Case Description: This test case verifies the functionality for instructors to create assignments with text-based submission for learners.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- Instructor is logged in.
- A course is available for the instructor to manage.

Test Data: Assignment title, detailed description, submission instructions.

Test Steps:

1. Navigate to the course dashboard.
2. Click on the "Create Assessment" button.
3. Select "Assignment" from the assessment type options.
4. Enter a title for the assignment.
5. Provide a detailed description and submission instructions.
6. Save the assignment.

Postconditions:

- A new assignment is added to the course.
- Learners can submit text-based responses to the assignment.

Expected Result: The system allows instructors to create and save assignments with text-based submission options for learners.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: LRN_003

Test Case Title: Verify automatic grading for multiple-choice questions

Test Case Description: This test case confirms the system automatically grades multiple-choice questions in a quiz.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* A quiz containing multiple-choice questions has been created (see Test Case 1)

* Learners complete the quiz and submit their answers.

Test Data: Quiz with objective questions and pre-defined correct answers.

Test Steps:

1. A learner accesses the created quiz and attempts all questions.

2. The learner submits the quiz.

Postconditions:

* Quiz results are displayed.

Expected Result: The system accurately scores multiple-choice questions based on the learner's selected answers, providing an automatic grade.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: LRN_004

Test Case Title: Verify the provision of feedback for correct and incorrect answers

Test Case Description: This test case checks if the system provides feedback to learners after grading, distinguishing between correct and incorrect answers.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Learners have submitted a quiz or assignment containing subjective questions.

Test Data: Quiz with both objective and subjective questions, instructor's feedback for subjective questions.

Test Steps:

1. Learners attempt the quiz or assignment.

2. Learners submit the quiz or assignment.

3. Learners view their graded results.

Postconditions:

* Feedback on subjective questions is provided by the instructor.

Expected Result: The system displays feedback to learners for both correct and incorrect answers, particularly highlighting areas for improvement on subjective questions where instructor feedback is available.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: LRN_005

Test Case Title: Verify the ability to manually grade subjective questions

Test Case Description: This test case ensures instructors can manually grade subjective questions, such as essays or short-answer questions.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* A quiz or assignment containing subjective questions has been created.

* Learners have submitted the quiz or assignment.

Test Data: A submitted subjective question from a learner, grading rubric.

Test Steps:

1. The instructor accesses the submitted assessments.

2. The instructor reviews the learner's submitted response.

3. The instructor applies the defined grading rubric.

4. The instructor saves the graded response.

Postconditions:

* The learner's assessment is marked as graded.

Expected Result: The system allows instructors to efficiently manually grade subjective questions based on defined criteria and provide personalized feedback to learners.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 54

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Forums

Feature description -

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge sharing, enriching the learning experience through diverse perspectives and active participation.

Userstory title -

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

Acceptance criteria -

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all devices.']

Generated Test Cases:

Here are 5 test cases for the given input data, following the provided template:

Test Case 1

Test Case ID: disF89_pos

Test Case Title: Verify that learners can post questions and responses in forums

Test Case Description: This test case verifies the ability of learners to create new forum posts and reply to existing discussions.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- A Learner account is logged in.
- A forum exists within a course.

Test Data: Sample question, sample response

Test Steps:

1. Navigate to the specific course forum.
2. Click on the "New Post" or "Create Thread" button.
3. Enter the sample question in the designated text field.
4. Click on the "Post" or "Submit" button to create the new post.
5. Locate the created post and reply to it with the sample response.
6. Click "Post" or "Submit" to save the response.

Postconditions:

- The new post and the response are successfully displayed in the forum.

Expected Result: The learner can successfully post a question and respond to it within a forum.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: disTh_neg

Test Case Title: Verify that learners cannot post in closed forums

Test Case Description: This test case ensures learners cannot post in forums that have been explicitly closed by instructors.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- A Learner account is logged in.
- A forum exists within a course that has been marked as closed.

Test Data: Sample question

Test Steps:

1. Navigate to the closed forum in the course.
2. Attempt to create a new post using the sample question.

Postconditions:

- No new post is created

Expected Result: An error message is displayed indicating that the forum is closed, preventing learners from posting new content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

Test Case 3

Test Case ID: disThrd

Test Case Title: Verify that forums support threaded discussions

Test Case Description: This test case ensures that replies to forum posts are displayed as nested threads, creating a clear hierarchy of conversation.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- A Learner account is logged in.
- A forum exists within a course.

Test Data: Sample question, sample response

Test Steps:

1. Navigate to the forum.
2. Create a new post using the sample question.
3. Reply to the created post using the sample response.

Postconditions:

- The response is displayed as a nested thread below the original post.

Expected Result: The forum displays threaded discussions, with replies organized logically beneath their corresponding posts.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: disMod

Test Case Title: Verify that instructors can moderate discussions

Test Case Description: This test case ensures instructors can manage forum posts, including editing, deleting, and marking them as resolved.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- An Instructor account is logged in.
- A forum exists within a course.

Test Data: Incorrect forum post, duplicate forum post

Test Steps:

1. Navigate to the forum.

2. Identify an inappropriate post (e.g., off-topic, spam)
3. Select the option to edit, delete, or mark as resolved for the identified post.

Postconditions:

- The post is edited, deleted, or marked as resolved accordingly.

Expected Result: Instructors can effectively moderate forum discussions and maintain a healthy learning environment.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: disNtf

Test Case Title: Verify that learners receive notifications for new posts

Test Case Description: This test case ensures learners receive timely notifications when new posts are made in forums they are following or actively participating in.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- A Learner account is logged in.
- A forum exists within a course that the learner is enrolled in.

Test Data: New forum post

Test Steps:

1. Log into the learner account.

2. Navigate to the forum and participate in an active discussion.

3. Exit the forum and log back in after a short interval.

4. A new post in the same forum should be created.

4. Observe the notification system for new updates from the forum.

Postconditions:

- The learner receives a notification about the new post.

Expected Result: When a new post is made in a forum a learner is following or actively involved in , they receive a timely notification.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 55

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Live Classes and Webinars

Feature description -

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

Userstory title -

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

Acceptance criteria -

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

Generated Test Cases:

Here are 5 test cases for the Online Learning Platform's "Live Classes and Webinars" feature:

Test Case 1

Test Case ID: OLPLiveClass001

Test Case Title: Verify that live classes and webinars are conducted through video conferencing tools.

Test Case Description: This test case ensures the platform utilizes a reliable video conferencing tool for live sessions, enabling seamless interaction between instructors and learners.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Instructor is logged in.
- A live class/webinar is scheduled.

- Learner is logged in and invited to the session.

Test Data: Test session content, potential pre-recorded materials

Test Steps:

1. The instructor opens the scheduled live class/webinar.
2. The instructor navigates the video conferencing tool and confirms all features (audio, video, screen sharing) are functioning correctly.
3. The learner attends the session and verifies their audio and video are working.
4. During the session, the instructor shares a screen or resource, and the learner confirms they can see it.

Postconditions: Live session ends.

Expected Result: Live session proceeds smoothly through the video conferencing tool. Audio and video work correctly for both instructor and learner.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: OLPLiveClass002

Test Case Title: Verify that learners can ask questions in real-time.

Test Case Description: This test case confirms learners can effectively engage with instructors by asking questions during live sessions.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Instructor is logged in.
- A live class/webinar is in progress.
- Learner is logged in and attending the session.

Test Data: Example questions related to session content

Test Steps:

1. The learner types a question in the designated chat or Q&A section.
2. The learner submits the question.
3. The instructor acknowledges the question and provides a response.

Postconditions: Question is answered.

Expected Result: Learner can successfully ask questions during the live session, and the instructor responds appropriately.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: OLPLiveClass003

Test Case Title: Verify that sessions are recorded and available for later viewing.

Test Case Description: This test case verifies the recording functionality of live sessions, ensuring learners have access to content after the live event.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Instructor is logged in and has recording enabled.

- A live class/webinar is scheduled and conducted.

Test Data: -

Test Steps:

1. The instructor ends the live class/webinar.

2. The instructor locates the recording of the session.

3. The learner accesses the recording from their dashboard or course page.

Postconditions: Recording is accessible.

Expected Result: A recording of the live session is available for learners to view after the live event.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive + Verification

Test Case 4

Test Case ID: OLPLiveClass004

Test Case Title: Verify that instructors can share screen and resources during sessions.

Test Case Description: This test case ensures instructors can effectively present content and materials during live sessions by sharing their screen and resources.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Instructor is logged in and conducting a live class/webinar.
- Laptop or desktop with necessary resources (presentation slides, document, etc.) is connected.

Test Data: Presentation slides, instructional document

Test Steps:

1. The instructor opens the screen sharing feature during the live session.
2. The instructor selects the resource they wish to share (slides, document, etc.).
3. Learners confirm they can see the shared content.

Postconditions: Shared content is displayed. Session continues.

Expected Result: The instructor successfully shares their screen and pre-prepared resources with the learners during the live session.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: OLPLiveClass005

Test Case Title: Verify that notifications are sent before sessions start.

Test Case Description: This test case confirms the system effectively sends notifications to learners prompting them about upcoming live sessions.

Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

- Learner is logged in.
- Learner is enrolled in a course with upcoming live sessions.

Test Data: Active email address associated with learner account.

Test Steps:

1. A live class/webinar is scheduled within the upcoming days.
2. The system sends a notification email to the learner about the upcoming session.
3. The learner checks their email and confirms they received the notification.

Postconditions: Notification delivered.

Expected Result: The learner receives a notification email with details about the upcoming live session well in advance of the scheduled start time.

Severity: Moderate

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 56

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Mobile Accessibility

Feature description -

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

Userstory title -

As a learner, I want mobile access to the platform so that I can learn on the go.

Acceptance criteria -

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

Generated Test Cases:

Here are 5 test cases covering the acceptance criteria for mobile accessibility in the Online Learning Platform:

Test Case ID: MobileAcc1

Test Case Title: Verify that the platform is accessible on smartphones and tablets

Test Case Description: This test case aims to verify that the learning platform's website functions correctly and is usable on both smartphones and tablets, ensuring a seamless browsing and learning experience across diverse mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Access to a smartphone and a tablet with stable internet connections.
- Different screen sizes and operating systems (iOS and Android) should be tested.

Test Data: N/A

Test Steps:

1. Access the learning platform using a web browser on the smartphone.
2. Navigate through various sections of the platform (e.g., home page, courses, profile).
3. Access and interact with features like course listings, videos, quizzes, and forums.
4. Repeat steps 1-3 using a tablet.

Postconditions:

- The mobile versions of each page should load without errors.
- All buttons, menus, and interactive elements should be responsive and functional.

Expected Result: The platform is fully functional and accessible on both smartphones and tablets, providing a consistent user experience across different mobile devices.

Severity: Major

Type of Testing: System Testing, Compatibility Testing

Test Case Approach: Positive

Test Case ID: MobileAcc2

Test Case Title: Verify that the mobile UI is optimized for touch navigation

Test Case Description: This test case evaluates the platform's user interface on mobile devices, specifically how it accommodates touch-based interactions for seamless navigation and usability.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Access to a smartphone with stable internet connection

Test Data: N/A

Test Steps:

1. Access the platform on the smartphone.
2. Attempt to navigate through menus, buttons, and hyperlinks using finger taps.
3. Test swipe gestures for navigation between screens or lists.
4. Select interactive elements like checkboxes, radio buttons, and dropdown menus using touch input.

Postconditions: N/A

Expected Result: All elements respond accurately to touch interactions, displaying a natural and intuitive mobile user experience. Touch gestures should be smooth and responsive, enhancing user navigation and interaction.

Severity: Major

Type of Testing: Usability Testing, System Testing

Test Case Approach: Positive

Test Case ID: MobileAcc3

Test Case Title: Verify the accessibility of course materials on mobile devices

Test Case Description: This test case assesses whether course content, including text, images, and videos, is accessible and readable on mobile screens.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Complete access to a course with varied content types (text, images, videos)
- Stable internet connection on a smartphone.

Test Data: A course with different content formats

Test Steps:

1. Access a course on the smartphone.
2. Read through text-based course materials, verifying readability and font size.
3. View images and videos within the course, ensuring they load properly and are viewable without distortion.

Postconditions: N/A

Expected Result: All course materials display clearly and are easily readable on the mobile device screen. Images and videos should load without interruptions and be optimized for mobile screen sizes.

Severity: Major

Type of Testing: System Testing, Accessibility Testing

Test Case Approach: Positive

Test Case ID: MobileAcc4

Test Case Title: Verify that notifications are sent to mobile devices

Test Case Description: This test case confirms that the platform successfully sends push notifications to users' mobile devices, keeping them informed about important updates, course activities, and alerts.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- User has logged in and enabled push notifications in their mobile device settings.

Test Data: N/A

Test Steps:

1. Log in to the platform on the mobile device.
2. Trigger a notification event (e.g., new course announcement, assignment reminder).

Postconditions:

- The user should receive a notification on their mobile device.

Expected Result: The platform delivers notifications promptly to the user's mobile device, ensuring they are kept informed about relevant events and activities within the learning platform.

Severity: Major

Type of Testing: System Testing, Functionality Testing

Test Case Approach: Positive

Test Case ID: MobileAcc5

Test Case Title: Verify offline access to downloaded materials

Test Case Description: This test case examines the functionality of downloading and accessing course materials offline, ensuring learners can continue their education even without an internet connection.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- Stable internet connection to download materials
- Space available on the mobile device for downloads

Test Data: A course with downloadable materials (e.g., PDF documents, audio files)

Test Steps:

1. While connected to the internet, download relevant course materials on the mobile device.
2. Disconnect from the internet.
3. Attempt to access the downloaded materials offline.

Postconditions:

- Downloaded materials should be accessible without an internet connection.

Expected Result: The downloaded course materials are accessible offline, allowing learners to continue their studies without relying on an active internet connection.

Severity: Major

Type of Testing: System Testing, Functionality Testing

Test Case Approach: Positive

Data - 57

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Resource Library

Feature description -

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

Userstory title -

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

Acceptance criteria -

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

Generated Test Cases:

Test Cases for Online Learning Platform - Resource Library Feature

Here are 5 detailed test cases for the defined User Story and Acceptance Criteria:

Test Case 1

* **Test Case ID:** resLib01
* **Test Case Title:** Verify that the resource library includes e-books, articles, and reference guides
* **Test Case Description:** This test case aims to confirm the presence of the required resource types within the library, ensuring the platform meets the specified content diversity.
* **Test Suite:** Resource Library
* **Test Priority:** High
* **Preconditions:**
    - User is logged in as a learner
    - Access to the resource library section is granted
* **Test Data:** None
* **Test Steps:**
    1. Navigate to the Resource Library section of the platform.
    2. Verify that different categories of resources, such as e-books, articles, and reference guides, are prominently displayed.
    3. Browse through at least three resources within each category.
    4. Ensure that the resources align with relevant topics covered in the platform's courses.
* **Postconditions:** Resources are categorized and displayed appropriately.
* **Expected Result:** The resource library contains a variety of e-books, articles, and reference guides organized into distinct categories. Resources should be relevant to the platform's course curriculum.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Behaviour:** Positive

Test Case 2

* **Test Case ID:** resLib02

* **Test Case Title:** Verify that learners can search for specific materials in the resource library
* **Test Case Description:** This test case evaluates the search functionality of the resource library, ensuring learners can effectively find the desired materials.
* **Test Suite:** Resource Library
* **Test Priority:** High
* **Preconditions:**
    - User is logged in as a learner
    - Access to the resource library section is granted
* **Test Data:** Different keywords related to specific topics offered on the platform.
* **Test Steps:**
    1. Navigate to the Resource Library section.
    2. Enter a specific keyword related to a course topic in the search bar.
    3. Click on the search button.
    4. Verify that search results display relevant resources pertaining to the keyword.
    5. Repeat steps 2-4 with different keywords encompassing various topics.
* **Postconditions:** Search results are displayed in a clear and organized manner.
* **Expected Result:** Search functionality accurately retrieves relevant resources based on entered keywords, allowing learners to efficiently find the materials they need.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Behaviour:** Positive

Test Case 3

* **Test Case ID:** resLib03
* **Test Case Title:** Verify that resources are categorized and tagged for easy navigation
* **Test Case Description:** This test case checks the categorization and tagging system of the resource

library, ensuring easy browsing and locating specific materials.

* **Test Suite:** Resource Library

* **Test Priority:** Medium

* **Preconditions:**

    - User is logged in as a learner

    - Access to the resource library section is granted

* **Test Data:** None

* **Test Steps:**

    1. Navigate to the Resource Library section.

    2. Observe the categorization structure (e.g., by subject, type, level).

    3. Click on different categories and browse through the resources within each.

    4. Verify that resources are tagged with relevant keywords.

5. Search for a resource using a specific tag and confirm that the search results display the intended resource.

* **Postconditions:** Resource categorization and tagging are accurate and user-friendly.

* **Expected Result:** Resources are effectively categorized and tagged, allowing learners to easily navigate and find materials based on subject, type, level, or keywords.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Behaviour:** Positive

Test Case 4

* **Test Case ID:** resLib04

* **Test Case Title:** Verify that learners can download resources for offline access

* **Test Case Description:** This test case confirms the functionality for downloading resources, enabling learners to access them even without an internet connection.

* **Test Suite:** Resource Library

* **Test Priority:** High

* **Preconditions:**

    - User is logged in as a learner

    - Device has sufficient storage space

    - Internet access is available

* **Test Data:** A resource (e.g., an e-book or article) available for download.

* **Test Steps:**

    1. Navigate to the chosen resource in the Resource Library.

    2. Verify a download button or option is present.

    3. Click the download button.

    4. Confirm that the download process begins successfully.

    5. Locate the downloaded resource on the device's storage.

* **Postconditions:** Downloaded resource is accessible offline.

* **Expected Result:** The chosen resource downloads completely to the device and is accessible offline as intended.

    * **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Behaviour:** Positive

Test Case 5

* **Test Case ID:** resLib05

* **Test Case Title:** Verify that instructors can recommend resources to learners

* **Test Case Description:** This test case evaluates the instructor's ability to recommend resources to learners, supporting effective learning experiences.

* **Test Suite:** Resource Library

* **Test Priority:** Medium

* **Preconditions:**

    - Instructor account is logged in

    - At least one course is assigned to the instructor

    * Access to the Resource Library section is granted

* **Test Data:** A relevant resource (e.g., e-book, article) from the library.

* **Test Steps:**

    1. Log into the platform as an instructor.

    2. Navigate to the course where recommendations are to be made.

    3. Locate the section for resource recommendations.

    4. Search for and select the desired resource to recommend.

    5. Add a brief description or note explaining the relevance of the resource.

    6. Submit the recommendation.

    7. Verify that the recommended resource appears in the learners' resource library.

* **Postconditions:** Resource recommendation is visible to logged-in learners.

* **Expected Result:** The instructor successfully recommends a specific resource to learners enrolled in the course.

Data - 58

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Peer Review

Feature description -

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

Userstory title -

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

Acceptance criteria -

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers remain anonymous to ensure unbiased feedback.']

Generated Test Cases:

Here are 5 test cases for the provided user story and acceptance criteria, using the provided template:

Test Case 1

Test Case ID: lr4k7p

Test Case Title: Verify that learners can submit assignments for peer review

Test Case Description: This test case verifies the functionality for learners to submit their work for peer review within the stipulated deadlines and criteria.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- A learner is logged into the platform.

- A course with a peer review assignment is selected.
- The assignment deadline has not passed.

Test Data: A sample assignment file (e.g., word document, PDF, project code)

Test Steps:

1. Navigate to the selected course and locate the assigned peer review task.
2. Upload the prepared assignment file adhering to the file format and size limitations.
3. Add any necessary comments or instructions for reviewers.
4. Submit the assignment for peer review.

Postconditions:

- The assignment appears in the designated peer review queue.
- The learner receives a confirmation message about successful submission.
- The assignment status is updated to "Pending Review".

Expected Result: The learner's assignment is successfully submitted for peer review and the system acknowledges the action.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: gF2jw9

Test Case Title: Verify that peers can provide feedback on submitted assignments

Test Case Description: This test case ensures that peers can effectively provide constructive feedback on submitted assignments in a structured manner.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- A learner has submitted an assignment for peer review.

- Another logged-in learner is assigned as a reviewer for the submitted assignment.

Test Data: A sample feedback comment (e.g., strengths, areas for improvement, suggestions)

Test Steps:

1. Navigate to the peer review assignment dashboard and select the assigned task.

2. Download the submitted assignment file.

3. Review the assignment content thoroughly.

4. Use the provided feedback form to add constructive comments addressing the assignment's strengths, weaknesses, and potential improvements.

5. Submit the feedback.

Postconditions:

- The feedback is saved and appears alongside the submitted assignment.

- The reviewer is notified that their feedback has been submitted.

- The submitted assignment status is updated (e.g., "Reviewed").

Expected Result: Peers can successfully providestructured and actionable feedback on submitted assignments.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: y8p3hn

Test Case Title: Verify that a structured review process guides the feedback

Test Case Description: This test case ensures that the platform provides a clear and organized structure for peer review, guiding reviewers and ensuring comprehensive feedback.

Test Suite: Peer Review

Test Priority: Medium

Preconditions:

- A learner has submitted an assignment for peer review.
- A reviewer is assigned to the submission.

Test Data: Assignment feedback rubric or guidelines

Test Steps:

1. Access the peer review assignment.
2. Examine the provided feedback guidelines or rubric.
3. Note the specific criteria for evaluation.
4. Use the designated feedback form to provide comments addressing each criterion.

Postconditions:

- Reviewee can easily see how their work is being assessed.
- Feedback aligns with the established criteria.
- The platform guides reviewers to provide comprehensive and focused feedback.

Expected Result: The peer review process is structured and guided by clear criteria, ensuring quality and consistency in feedback.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: p04w9t

Test Case Title: Verify that learners can view feedback on their work

Test Case Description: This test case confirms that learners can access and review the feedback provided by their peers on their submitted assignments.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- A learner has submitted an assignment for peer review.
- At least one reviewer has provided feedback on the assignment.

Test Data: No test data needed

Test Steps:

1. Log in as the learner who submitted the assignment.
2. Access the peer review assignment dashboard.
3. Locate the review assignment.
4. View the feedback provided by peers.

Postconditions:

- Learner can see all the comments and ratings on their assignment.
- The feedback is presented in a clear and readable format.

Expected Result: Learners can successfully view all provided feedback on their work.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: f2z7qk

Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback

Test Case Description: Ensures that reviewers' identities are concealed to prevent biased or personal feedback, promoting fairness and objectivity.

Test Suite: Peer Review

Test Priority: Medium

Preconditions:

- A learner has submitted an assignment for peer review.

- Multiple reviewers are assigned to the submission.

Test Data: No Specific Test Data Needed

Test Steps:

1. Log in as the learner who submitted the assignment.
2. Access the peer review assignment dashboard to view the feedback.
3. Observe the feedback comments; verify that reviewer identities are not disclosed.

Postconditions:

- No reviewer names or identifying information are visible alongside the feedback.
- The platform maintains the anonymized nature of the review process.

Expected Result: Reviewers' identities remain hidden, fostering unbiased and objective feedback.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 59

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Certificates and Badges

Feature description -

Upon successful completion of courses or specific milestones, learners receive certificates and badges to recognize their achievements and accomplishments. These credentials provide tangible recognition of learners' efforts, skills, and expertise, enhancing their credentials and demonstrating their commitment to continuous learning and professional development.

Userstory title -

As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

Acceptance criteria -

['Learners earn certificates upon course completion.', 'Badges are awarded for specific achievements and milestones.', 'Certificates and badges are accessible through learner profiles.', 'Learners can share their certificates and badges on social media.', 'Instructors can issue custom badges for special achievements.']

Generated Test Cases:

Online Learning Platform - Certificates & Badges Test Cases

Here are 5 test cases designed to cover the acceptance criteria for the "Certificates and Badges" feature:

Test Case 1:

* **Test Case ID:** clb7r5

* **Test Case Title:** Verify that learners earn certificates upon course completion.

* **Test Case Description:** This test case verifies that a learner successfully receives a certificate after finishing all modules and assessments of a designated course on the platform.

* **Test Suite:** Certificates and Badges

* **Test Priority:** High

* **Preconditions:**

  - A learner account is created.

  - A course with multiple modules and assessments is available.

* **Test Data:** Course syllabus with modules and assessments, learner progress data.

* **Test Steps:**

  1. Navigate to the available course and enroll as a learner.

  2. Complete all modules and assessments within the course.

  3. Verify the "Course Completed" notification.

  4. Access the learner's profile and confirm the presence of the earned certificate.

* **Postconditions:**

  - Certificate is issued and displayed in the learner's profile.

* **Expected Result:** The learner receives a digital certificate indicating successful completion of the course, accessible within their learner profile.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** dfg28k

* **Test Case Title:** Verify that badges are awarded for specific achievements and milestones.

* **Test Case Description:** This test case verifies that learners receive badges upon completing specific actions or achieving defined milestones within a course.

* **Test Suite:** Certificates and Badges

* **Test Priority:** Medium

* **Preconditions:**

  - A learner account is created.

  - A course with predefined badge criteria (e.g., completing a module, scoring above a certain mark on a quiz) is available.

* **Test Data:** Course structure with badge criteria, quiz scores, module completion data.

* **Test Steps:**

    1. Enroll in the course with predefined badge criteria.

    2. Complete the specified module(s) or achieve the required quiz score.

    3. Verify the badge is awarded and appears on the learner's profile.

* **Postconditions:**

    - Badge is visible in the learner's profile and associated with the achieved milestone.

* **Expected Result:** The learner receives the designated badge after completing the specified action or reaching the pre-defined milestone.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** hjks90

* **Test Case Title:** Verify learners can share their certificates and badges on social media.

* **Test Case Description:** This test case ensures learners can seamlessly share their earned certificates and badges with their social media networks directly from the platform.

* **Test Suite:** Certificates and Badges

* **Test Priority:** Medium

* **Preconditions:**

    - Learner account with successfully achieved achievements, including certifications and badges.

    - Social media integration (Facebook, Twitter, LinkedIn, etc.) is configured and working.

* **Test Data:** Pre-configured social media accounts for testing (preferably dummy).

* **Test Steps:**

   1. Access the learner's profile and locate the certification/badge details.

   2. Click the share button associated with the certification/badge.

   3. Select the desired social media platform for sharing.

   4. Verify a share post with the certificate/badge appears on the chosen platform.

* **Postconditions:**

   - The share post is successfully published on the selected social media platform.

* **Expected Result:** The learner can share their earned certificate/badge to their social media platforms with minimal effort.

* **Severity:** Major

* **Type of Testing:** Functional, Usability

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** asdfgh8

* **Test Case Title:** Verify instructors can issue custom badges for special achievements.

* **Test Case Description:** This test case confirms that instructors have the ability to create and assign unique badges for learners who demonstrate exceptional performance or achieve specific goals beyond the standard course criteria.

* **Test Suite:** Certificates and Badges

* **Test Priority:** Medium

* **Preconditions:**

   - An instructor account is created.

   - A course is available with predefined badges.

* **Test Data:** Criteria for a new custom badge, learner data.

* **Test Steps:**

   1. Navigate to the course section managed by an instructor.

2. Access the badges section and select the option to create a custom badge.

3. Define the criteria and design for the new custom badge.

4. Assign the custom badge to a learner who meets the defined criteria.

5. Verify the learner's profile reflects the awarded custom badge.

* **Postconditions:**

- The custom badge is successfully created and added to the badge system.

* **Expected Result:** The instructor can create and issue custom badges based on specific performance or achievement criteria, and these badges are displayed accurately on learner profiles.

* **Severity:** Major

* **Type of Testing:** Functional, User Rights

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** zxcvbnm1

* **Test Case Title:** Verify accessibility of certificates and badges through learner profiles.

* **Test Case Description:** This test case ensures learners can easily access and view their earned certificates and badges within their dedicated profiles.

* **Test Suite:** Certificates and Badges

* **Test Priority:** Medium

* **Preconditions:**

- Learner accounts with at least one earned certificate/badge.

* **Test Data:** Learner profiles.

* **Test Steps:**

1. Access a learner's profile page.

2. Navigate to the "Certificates" or "Badges" section within the profile.

3. Verify displayed certificates/badges are correctly associated with the respective learner.

* **Postconditions:**

    - Certificates and badges are presented in a user-friendly and organized manner.

* **Expected Result:** Certificates and badges are prominently displayed within the learner's profile, offering easy access and visibility of their achievements.

* **Severity:** Major

* **Type of Testing:** Usability, Functional

* **Test Case Approach:** Positive

Data - 60

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Gamification Elements

Feature description -

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

Userstory title -

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

Acceptance criteria -

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

Generated Test Cases:

Online Learning Platform - Gamification Elements Test Cases

Here are 5 test cases designed to cover the acceptance criteria for gamification elements within the Online Learning Platform:

Test Case 1:

Test Case ID: GLP-GC-001

Test Case Title: Verify that learners earn points for completing activities.

Test Case Description: This test case verifies that learners receive points for engaging in various learning activities as defined by the platform.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- A learner account is created & logged in.

- A course with learnable content is enrolled.

Test Data:

* List of activities within the course (e.g., quizzes, lessons, assignments)
* Expected point value for each activity

Test Steps:

1. Navigate to the chosen course.
2. Complete each listed activity within the course.
3. View learner's profile or gamification dashboard.

Postconditions:

- Points are correctly accumulated in the learner's profile.

Expected Result: Points are correctly awarded to the learner for completing each activity as specified, reflected on their profile or gamification dashboard.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: GLP-GC-002

Test Case Title: Verify that levels are achieved based on accumulated points.

Test Case Description: This test case confirms that learners progress through levels based on the points they earn.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- A learner account is created & logged in.
- A course is enrolled.
- Points have been earned through completing activities.

Test Data:

* Point thresholds for each level (e.g., Level 1: 0-100 points, Level 2: 101-200 points, etc.)

Test Steps:

1. Track the learner's progress through activities, ensuring they accrue sufficient points to reach a specific level threshold.
2. Observe the learner's profile or gamification dashboard for level updates.

Postconditions:

- Successfully reach the desired level.

Expected Result: The learner's level progresses according to the pre-defined point thresholds, reflected on the learner's profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: GLP-GC-003
Test Case Title: Verify that leaderboards display top-performing learners.
Test Case Description: This test ensures that leaderboards accurately present learners ranked based on their gamification scores.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Multiple learner accounts are created & logged in.
- Learners complete activities within courses, earning points.

Test Data:

* A predefined set of learners with different point accumulations.

Test Steps:

1. Access the leaderboard feature.
2. Verify the order of learners on the leaderboard aligns with their point totals.

Postconditions:

- Access leaderboards across various timeframes (daily, weekly, overall).

Expected Result:

The leaderboard accurately reflects the top-performing learners based on their accumulated points, with sorting consistent across different timeframes.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: GLP-GC-004
Test Case Title: Verify that badges are awarded for reaching milestones.

Test Case Description: This test confirms that learners receive badges upon achieving specific milestones within the platform.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner account is created & logged in.
- Course enrolment and participation in specific activities.

Test Data:

* List of milestones and corresponding badges.

Test Steps:

1. Identify milestones within chosen courses.
2. Successfully complete the required activities to achieve each milestone.
3. Check the learner's profile or badge section.

Postconditions:

- Badges are awarded and prominently displayed in the learner's profile.

Expected Result: Learners earning badges for completing pre-defined milestones, with badges visually presented in the learner's profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: GLP-GC-005

Test Case Title: Verify learners can view their gamification progress.

Test Case Description: This test confirms the accessibility and functionality of the gamification dashboard for learners to monitor their achievements.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

    - Learner account is created & logged in.

Test Data:

* Learner's points, levels, badges, and activity history.

Test Steps:

1. Access the gamification dashboard.
2. Review displayed information on points, levels, earned badges, and activity progress.
3. Navigate through various sections within the dashboard.

Postconditions:

- Data is accurately presented and user-friendly.

Expected Result: The learner can seamlessly access and understand the gamification dashboard, visualizing their progress and achievements clearly.

Data - 61

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Multilingual Support

Feature description -

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

Userstory title -

As a learner, I want multilingual support so that I can access the platform in my preferred language.

Acceptance criteria -

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated help and support resources.', 'Instructors can provide content in multiple languages.']

Generated Test Cases:

Here are 5 test cases for the Online Learning Platform's Multilingual Support feature, following the provided template:

Test Case 1

Test Case ID: OLP-MS-001

Test Case Title: Verify that users can select their preferred language from the platform interface

Test Case Description: This test case verifies the functionality of the language selection feature and ensures users can switch between supported languages in the platform's interface.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- The platform is accessible.

- Multiple languages are configured within the system.

Test Data: List of supported languages

Test Steps:

1. Access the platform as a learner

2. Locate the "Language Settings" or similar option

3. Select a supported language from the list

4. Verify that all interface elements (menus, buttons, text) are displayed in the chosen language.

Postconditions:

- The user's preferred language is updated in the system.

Expected Result: The platform interface seamlessly adapts to the chosen language, with all elements translated accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: OLP-MS-002

Test Case Title: Verify that course materials are available in various languages.

Test Case Description: This test case confirms that course content is translated into multiple languages as specified.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- A course with multiple language versions is available in the system.

- The learner has selected a supported language.

Test Data: List of supported languages for a specific course, sample course content

Test Steps:

1. Access a course known to have multiple language versions.

2. Select the desired language version of the course.

3. Verify that all course content (lectures, readings, exercises, descriptions) are available and accurately translated into the chosen language.

Postconditions:

- Course material is loaded successfully in the selected language.

Expected Result: The chosen language version of the course content is displayed without errors or omissions in translation.

Test Case 3

Test Case ID: OLP-MS-003

Test Case Title: Verify translation accuracy of help and support resources.

Test Case Description: This test case ensures help articles, FAQs, and support documentation are translated correctly to meet the needs of learners in various languages.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- The platform has a dedicated help or support section.

- Multiple languages are configured for help resources.

Test Data: List of translated help articles or FAQs

Test Steps:

1. Access the help or support section within the platform.

2. Select the desired language.

3. Choose a help article or FAQ relevant to the platform's functionality.

4. Verify that the content is translated accurately and comprehensively.

Postconditions:

- Access to translated help content is provided.

Expected Result: Help resources are correctly translated into the chosen language, conveying information clearly and accurately.

Test Case 4

Test Case ID: OLP-MS-004

Test Case Title: Verify that instructors can provide course content in multiple languages

Test Case Description: This test case confirms instructors have the ability to upload and present course materials in different languages.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- User is logged in as an instructor

- Learning management system (LMS) allows multiple language support for content uploads.

Test Data: Sample course content in two different languages

Test Steps:

1. Access the course creation or editing interface as an instructor.

2. Upload course materials in the first chosen language.

3. Upload supplementary materials or translations in the second chosen language.

4. Verify that both language versions are accessible and distinguishable within the course structure.

Postconditions:

- Course material in two languages is saved and displayed.

Expected Result: Instructors can effectively manage and deliver course content in multiple languages, catering to diverse learners.

Test Case 5

Test Case ID: OLP-MS-005

Test Case Title: Verify seamless language switching during platform usage.

Test Case Description: This test case tests how smoothly a learner can switch language preferences while navigating the platform, interacting with features, and accessing course materials.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- The platform is accessible.
- The learner has an active account.
- Multiple languages are configured for the system.

Test Data: Courses with content in multiple languages.

Test Steps:

1. Access the platform as a learner, select a language.
2. Navigate to different sections of the platform (dashboard, course catalog, personal profile).
3. Switch the language preference to a different supported language.
4. Verify that all platform elements adapt to the new language, including course content, menus, and user interface.

Postconditions:

- Language preference is updated effectively.

Expected Result: Language switching is seamless and consistent across all platform sections, ensuring a smooth user experience.

Data - 62

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Accessibility Features

Feature description -

The platform prioritizes accessibility by incorporating features such as screen reader compatibility, adjustable font sizes, and color contrast options. These accessibility features ensure that learners with disabilities can access and navigate the platform with ease, promoting inclusivity and equitable access to educational opportunities.

Userstory title -

As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

Acceptance criteria -

['The platform is compatible with screen readers.', 'Learners can adjust font sizes for better readability.', 'Color contrast options are available for improved visibility.', 'Accessibility features are easy to activate and customize.', 'Support resources are available to assist learners with disabilities.']

Generated Test Cases:

Here are 5 detailed test cases for the provided input data, following the outlined template:

Test Case ID: ACCS-01

Test Case Title: Verify that the platform is compatible with screen readers

Test Case Description: This test case verifies the accessibility of the online learning platform's content and features for users relying on screen readers. It ensures that key elements such as headings, links, buttons, and text descriptions are correctly identified and announced by a screen reader.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

* User is logged in to the platform.
* A screen reader (e.g., NVDA, JAWS) is installed and configured.

Test Data: No test data needed

Test Steps:

1. Navigate to the platform homepage using the screen reader.
2. Access various course listings, lectures, and interactive elements.
3. Explore menu options and user profile settings.
4. Attempt to complete a quiz or forum participation.
5. Verify that all screen reader announcements are clear, accurate, and provide sufficient context.

Postconditions:

* User navigates the platform successfully using the screen reader.

Expected Result: All platform components are announced correctly by the screen reader, enabling navigation and interaction for users with visual impairments.

Severity: Critical
Type of Testing: Accessibility Testing
Test Case Approach: Positive

Test Case ID: ACCS-02
Test Case Title: Verify that learners can adjust font sizes for better readability

Test Case Description: This test case validates the functionality of font size adjustments for learners who require larger or smaller text sizes. It checks the range of available font sizes and ensures they can be easily applied.

Test Suite: Accessibility Features

Test Priority: Medium
Preconditions:

* User is logged in to the platform.

Test Data: No test data needed
Test Steps:

1. Access a course module with considerable text content.
2. Locate and activate the font size adjustment option.

3. Choose progressively larger font sizes within the available range.
4. Verify that the increase in font size is noticeable and legible.
5. Repeat steps 3 and 4 with progressively smaller font sizes.

Postconditions:

* User successfully adjusts font size.

Expected Result: The platform provides a range of font sizes, enabling users to customize text clarity to their preferences.

Severity: Major

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive,

Test Case ID: ACCS-03

Test Case Title: Verify that color contrast options are available for improved visibility

Test Case Description: This test case examines the availability and effectiveness of color contrast options for learners with visual sensitivities. It ensures sufficient contrast between text and background colors for clear readability.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

* User is logged in to the platform.

Test Data: No test data needed

Test Steps:

1. Access a course module with various text blocks and backgrounds.

2. Locate and activate the color contrast option.

3. Select different contrast presets (e.g., high contrast, standard contrast, custom).

4. Evaluate the contrast between text and background colors in each preset.

5. Observe the change in readability with different contrast settings.

Postconditions:

* User successfully applies color contrast options.

Expected Result: The platform offers color contrast options that improve readability for learners with visual impairments or sensitivities.

Severity: Minor

Type of Testing: Functional Testing, Accessibility Testing

Test Case Approach: Positive

Test Case ID: ACCS-04

Test Case Title: Verify that accessibility features are easy to activate and customize

Test Case Description: This test case assesses the user-friendliness of activating and customizing accessibility features. It ensures that options are clearly labeled, easily accessible, and intuitive to use.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

* User is logged in to the platform.

Test Data: No test data needed

Test Steps:

1. Navigate to the platform?s accessibility settings or profile.
2. Locate and activate screen reader compatibility.
3. Adjust font size options using the provided controls.
4. Select a desired color contrast preset.
5. Verify that changes are applied immediately and effectively.

Postconditions:

* User has successfully activated and customized accessibility settings.

Expected Result: Accessibility features are readily accessible and customizable through a straightforward and user-friendly interface.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: ACCS-05

Test Case Title: Verify the availability and accessibility of support resources

Test Case Description: This test case confirms the existence and accessibility of support resources for learners experiencing accessibility-related issues.

Test Suite: Accessibility Features

Test Priority: Low

Preconditions:

* User is logged in to the platform

Test Data: No test data needed

Test Steps:

1. Access the platform's "Help" or "Support" section.
2. Look for dedicated resources for accessibility.
3. Review content for clarity, accuracy, and relevance to common accessibility challenges.
4. Attempt to contact support staff through available channels (e.g., email, chat).

Postconditions:

* User has reviewed or contacted support regarding accessibility.

Expected Result: The platform provides accessible support resources or a clear pathway for learners to seek assistance with accessibility-related matters.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 63

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Moderation

Feature description -

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

Userstory title -

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

Acceptance criteria -

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

Generated Test Cases:

Here are 5 test cases designed to cover the acceptance criteria for the "Discussion Moderation" feature on the Online Learning Platform.

Test Case 1

Test Case ID: OLPL-MC-001

Test Case Title: Verify that instructors can edit posts in discussions.

Test Case Description: This test case verifies the instructor's ability to edit existing posts within a discussion forum on the Online Learning Platform.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

* Instructor is logged in.
* A discussion forum exists with at least one instructor-created post.

Test Data: An example existing post in the discussion forum.

Test Steps:

1. Navigate to the discussion forum.
2. Locate the instructor's existing post.
3. Click the edit icon associated with the post.
4. Modify the content of the post.
5. Click the "Update" or "Save Changes" button.

Postconditions:

* The edited post appears in the forum with the updated content.

Expected Result: The instructor can successfully edit the existing post's content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: OLPL-MC-002

Test Case Title: Verify that instructors can delete posts from discussions.

Test Case Description: This test case ensures that instructors have the functionality to remove inappropriate or off-topic posts from discussion forums.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

* Instructor is logged in.
* A discussion forum exists with at least one instructor-created post.

Test Data: An example existing post in the discussion forum.

Test Steps:

1. Navigate to the discussion forum.
2. Locate the instructor's existing post.
3. Click the delete icon associated with the post.
4. Confirm the deletion action in the pop-up window.

Postconditions:

* The deleted post is removed from the forum.

Expected Result: The instructor can successfully delete the existing post from the discussion forum.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: OLPL-MC-003

Test Case Title: Verify that instructors can highlight important posts in discussions.

Test Case Description: This case examines the instructor's ability to highlight posts for emphasis, making them stand out to other participants.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

* Instructor is logged in.
* A discussion forum exists with relevant posts.

Test Data:

An existing post in the discussion forum.

Test Steps:

1. Go to the discussion forum.

2. Locate the post to be highlighted.

3. Click the "Highlight" or "Pin" button associated with the post.

Postconditions:

* The highlighted post displays visually, indicating its importance.

Expected Result: The instructor can successfully highlight the chosen post, making it noticeable to other discussion participants.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: OLPL-MC-004

Test Case Title: Verify that instructors can mute disruptive users in discussions.

Test Case Description: This test case confirms that instructors can silence users who engage in inappropriate or disruptive behavior, promoting a more civil discourse.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

* Instructor is logged in.
* A discussion forum exists with a user displaying disruptive behavior.

Test Data: A user account known to cause disruptive behavior in discussions.

Test Steps:

1. Navigate to the discussion forum.
2. Identify the disruptive user posting messages.
3. Click the "Mute" or "Silence" option associated with the user.
4. Confirm the muting action.

Postconditions:

* The muted user's posts are no longer visible to other participants in the forum.

Expected Result: The instructor successfully mutes the disruptive user, preventing further uncivil contributions to the discussion.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: OLPL-MC-005

Test Case Title: Verify that notifications are sent for moderated actions.

Test Case Description: This test case ensures that users are notified when their posts are edited, deleted, or when they are muted in a discussion forum.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

* Instructor is logged in.
* A user account is logged in while a post is being modified, deleted, or for which a mutes action is taken.

Test Data:

* An existing post in a discussion forum.

*   A user account with email notifications enabled.

* (additional mockups for email notifications might be useful)

Test Steps:

1. As the instructor, perform one of the following actions on a user's post without email notifications:

    * Edit the post

    * Delete the post

    * Mute the user

2. Immediately check the user's email inbox

Postconditions:

* The instructor's action on the post was completed within the discussion forum.

Expected Result: The user receives an email notification informing them of the action taken by the instructor.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 64

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Analytics

Feature description -

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement, performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning experience to meet learners' needs effectively.

Userstory title -

As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

Acceptance criteria -

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.', 'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.', 'Instructors can export analytics reports for further analysis.']

Generated Test Cases:

Test Cases for Learning Analytics Feature

Test Case 1

* **Test Case ID:** LA-001
* **Test Case Title:** Verify that learning analytics dashboards display engagement metrics
* **Test Case Description:** This test case verifies the display of key engagement metrics like views, time spent on content, completion rate, and forum activity on the learning analytics dashboard for instructors.
* **Test Suite:** Learning Analytics
* **Test Priority:** High
* **Preconditions:**
    * Instructor account is logged in
    * A course with some content and forum activity is available
* **Test Data:** N/A
* **Test Steps:**
    1. Navigate to the learning analytics dashboard.
    2. Verify the display of key engagement metrics like views, time spent on content, completion rate, and forum activity in a clear and understandable format.
    3. Analyze the data for trends and patterns in learner engagement.
* **Postconditions:** Instructor remains on the analytics dashboard.
* **Expected Result:** Learning analytics dashboard displays relevant engagement metrics with clear visualizations for instructor review.
* **Severity:** Major
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** LA-002
* **Test Case Title:** Verify that performance trends are visualized over time
* **Test Case Description:** This test case checks if the learning analytics dashboard effectively visualizes performance trends of learners on assessments, quizzes, and overall course progress over a defined period.
* **Test Suite:** Learning Analytics
* **Test Priority:** High
* **Preconditions:**
    * Instructor account is logged in
    * A course with quizzes and assessments is available
    * Some learner activity data exists within the specified timeframe.
* **Test Data:** Course with quizzes and assessments, historical learner data from multiple assessment attempts.
* **Test Steps:**
    1. Navigate to the learning analytics dashboard and select a specific timeframe for analysis.
    2. Verify the visualization of performance trends on assessments and quizzes through charts (e.g., line graph, bar chart)
    3. Examine the visualization of overall course progress trends over time.
* **Postconditions:** Instructor navigates between different timeframe options on the dashboard.
* **Expected Result:** The learning analytics dashboard presents clear and meaningful visualizations of learner performance trends over selected timeframes.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** LA-003
* **Test Case Title:** Verify that instructors can access detailed reports on learner progress
* **Test Case Description:** This test case validates an instructor's ability to access detailed reports on individual learner progress within a course, including quiz scores, completion status, and forum participation.

* **Test Suite:** Learning Analytics
* **Test Priority:** High
* **Preconditions:**
    * Instructor account is logged in
    * A course with multiple learners and assessment activities is available.
* **Test Data:** Course with multiple learners and assessment activities.
* **Test Steps:**
    1. Navigate to the learning analytics dashboard and select a particular course.
    2. Access the detailed learner progress report for a chosen learner.
    3. Verify that the report includes information on quiz scores, assignment completion status, forum activity, and overall course performance.
* **Postconditions:** Instructor navigates back to the main analytics dashboard.
* **Expected Result:** The detailed learner progress report provides comprehensive insights into the academic performance and engagement of individual learners within the course.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** LA-004
* **Test Case Title:** Verify that analytics data is updated in real-time
* **Test Case Description:** This test case ensures that learning analytics data is reflected in the dashboard promptly as learners interact with the platform.

* **Test Suite:** Learning Analytics
* **Test Priority:** Medium
* **Preconditions:**
    * Instructor account is logged in
    * A course with forum discussions and quizzes is available
    * Learners are actively engaging with the platform and course content.
* **Test Data:** N/A
* **Test Steps:**
    1. Observe the learner engagement metrics displayed on the dashboard.
    2. Trigger learner activity (e.g., post a forum comment, attempt a quiz)
    3. Monitor the dashboard for the updated metrics reflecting the recent learner activity within a reasonable timeframe (e.g., 5-10 seconds).
* **Postconditions:** Instructor remains on the analytics dashboard.
* **Expected Result:** Learning analytics data is updated in real-time as learner interactions occur, providing up-to-date insights.

* **Severity:** Critical
* **Type of Testing:** Functional Testing/Performance Testing
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** LA-005

* **Test Case Title:** Verify that instructors can export analytics reports for further analysis

* **Test Case Description:** This test case checks if instructors can export data from the learning analytics dashboard in a usable format (e.g., CSV, Excel) for more detailed offline analysis or integration with other systems.

* **Test Suite:** Learning Analytics

* **Test Priority:** Medium

* **Preconditions:**

  * Instructor account is logged in

  * A course with sufficient learner activity data is available

* **Test Data:** Course with data points to be exported.

* **Test Steps:**

  1. Access the learning analytics dashboard and select the desired data to be exported (e.g., learner progress, engagement metrics).

  2. Initiate the export process through a designated option on the dashboard.

  3. Verify that the exported data is received in the selected format (e.g., CSV, Excel).

* **Postconditions:** Instructor receives the exported data file.

* **Expected Result:** The system successfully exports the chosen analytics data in a usable format that can be easily imported and analyzed in external tools.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 65

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Learning Features

Feature description -

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

Userstory title -

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Acceptance criteria -

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

Generated Test Cases:

Here are 5 test cases for the provided user story and acceptance criteria:

Test Case 1

Test Case ID: OLP-SC-001

Test Case Title: Verify that learners can create groups for projects

Test Case Description: This test case verifies the functionality of creating groups for collaboration on projects within the online learning platform.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- User account is created and logged in

Test Data:

- Project title
- Desctiption for the project
- Group name

Test Steps:

1. Navigate to the "Projects" section of the platform.
2. Click on the "Create Group" button.
3. Enter a unique group name.
4. Provide a brief description for the group.
5. Click on the "Create Group" button.

Postconditions:

- A new group is created and visible to the user

Expected Result: The user successfully creates a new group for project collaboration.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: OLP-SC-002

Test Case Title: Verify that learners can join existing groups

Test Case Description: This test case verifies the functionality of joining existing groups for project collaboration.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- User account is created and logged in
- An existing group has been created

Test Data:

- Existing group name

Test Steps:

1. Navigate to the "Projects" section of the platform.
2. Locate the existing group by name.
3. Click on the "Join Group" button for the selected group.

Postconditions:

- The user is added to the group

Expected Result: The user successfully joins the existing group and can participate in discussions and activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: OLP-SC-003

Test Case Title: Verify that learners can share resources within groups

Test Case Description: This test case verifies the functionality of sharing learning resources (e.g., documents, videos, links) with other members within a group.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- User account is created and logged in
- The user is a member of an existing group

Test Data:

- A learning resource file (e.g., PDF document)

Test Steps:

1. Navigate to the group's discussion forum or designated resource sharing area.
2. Select the "Upload Resource" or similar option.
3. Choose the learning resource file from their computer.
4. Add a brief description or title for the shared resource.
5. Click on the "Share" or "Upload" button.

Postconditions:

- The resource is visible to all group members

Expected Result: The user successfully shares the learning resource within the group, and other members can access it.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: OLP-SC-004

Test Case Title: Verify that instructors can assign group projects

Test Case Description: This test case verifies the ability of instructors to assign group projects to students within the online learning platform, ensuring proper project management and collaborative learning.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Instructor account is logged in
- A course is open for enrollment
- Student accounts are created

Test Data:

- Project topic
- Project description
- Deadline date
- Group size requirement

Test Steps:

1. Navigate to the course section where project assignments are managed.
2. Click on the "Assign Group Project" button.
3. Enter a project title and provide a detailed description.
4. Set a deadline for project submission.
5. Specify the desired group size.
6. Automatically assign students to groups or allow students to select their groups.

Postconditions:

- The project is visible to students
- Students are automatically assigned to groups OR are given the option to choose their own group

Expected Result: The instructor successfully assigns a group project with clear instructions and expectations, enabling collaborative learning among students.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: OLP-SC-005

Test Case Title: Verify that learners can share achievements within social groups

Test Case Description: This test case verifies the functionality of learners sharing their accomplishments and milestones within social groups on the platform.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- User account is created and logged in

- The user is a member of an existing group

Test Data:

- Achievement text
- Optional: Image or file related to the achievement

Test Steps:

1. Navigate to the selected social group's discussion forum.
2. Share a message announcing an achievement.
3. Include relevant details about the achievement (e.g., completed a course, earned a special recognition).
4. Optionally attach an image or file to visually represent the achievement.

Postconditions:

- The achievement is visible to all group members

Expected Result: Learners can share their achievements within social groups, fostering a sense of community and celebrating accomplishments.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 66

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Authoring Tools

Feature description -

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

Userstory title -

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

Acceptance criteria -

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

Generated Test Cases:

Here are five test cases for the provided User Story and Acceptance Criteria:

Test Case 1

Test Case ID: content_authoring_001

Test Case Title: Verify that instructors can create multimedia presentations.

Test Case Description: This test case verifies that instructors can successfully create presentations incorporating various multimedia elements like images, videos, and audio within the learning platform's content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

* Instructor account is logged in.

* Access is granted to create new course content.

Test Data: Sample images, short video clips, and audio files.

Test Steps:

1. Navigate to the "Create New Content" section within the course dashboard.
2. Select the "Multimedia Presentation" template.
3. Upload the sample images, video clips, and audio files into the designated areas.
4. Arrange and sequence the multimedia elements within the presentation using the provided tools.
5. Preview the presentation to ensure all elements are displayed correctly and in the intended order.

Postconditions:

* A new multimedia presentation is created and saved in the course content.

Expected Result: The system successfully uploads and integrates the multimedia elements, allowing the instructor to create a visually engaging presentation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: interactive_modules_001

Test Case Title: Verify that interactive modules can be developed and integrated.

Test Case Description: This test case confirms the functionality of the platform's tools for creating interactive exercises, quizzes, and simulations that enhance learner engagement and knowledge retention.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor account is logged in.
- Access is granted to create new course content.

Test Data: Sample quiz questions, scenarios for simulations.

Test Steps:

1. Navigate to the "Create New Content" section within the course dashboard.
2. Select the "Interactive Module" template.
3. Input multiple-choice quiz questions with correct answers.
4. Design a simulation scenario with interactive elements and branching pathways.
5. Preview the interactive module to ensure functionality and flow are as intended.

Postconditions:

- A new interactive module is created and saved in the course content.

Expected Result: The system correctly renders the quiz questions, simulations, and interactive elements, providing learners with an engaging and interactive learning experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: rich_media_support_001

Test Case Title: Verify support for rich media resources like videos and animations.

Test Case Description: This test confirms the ability to embed and display videos and animations within course content using the platform's authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

* Instructor account is logged in.

* Access is granted to create new course content.

Test Data: Short video files, animated graphics files.

Test Steps:

1. Navigate to the "Create New Content" section within the course dashboard.

2. Select a suitable content type (e.g., text, presentation) where rich media could be incorporated.

3. Upload the sample video and animation files.

4. Insert the video and animation elements into the content at appropriate points.

5. Preview the content to ensure videos and animations display correctly and are playable.

Postconditions:

* Rich media resources are integrated and displayed within the course content.

Expected Result: The system successfully embeds and plays the uploaded video and animation files within the course content, enhancing the learning experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: user_friendliness_001

Test Case Title: Verify the user-friendliness and intuitiveness of content authoring tools.

Test Case Description: This test assesses the ease of use and clarity of navigation within the authoring tools, aiming to ensure instructors can quickly create quality content without significant technical difficulties.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

* Instructor account is logged in.

* Access is granted to create new course content.

Test Data: Usability feedback from instructors unfamiliar with the platform.

Test Steps:

1. Provide instructors with access to the platform's authoring tools.

2. Have them attempt to create a sample lesson plan or presentation using the tools.

3. Observe their workflow, note any challenges or confusing elements, and collect feedback.

Postconditions:

* Feedback is gathered on the user experience and tool functionality.

Expected Result: The authoring tools are found to be user-friendly, intuitive, and easy to learn, empowering instructors to create content efficiently.

Severity: Major

Type of Testing: Usability

Test Case Approach: Exploratory

Test Case 5

Test Case ID: preview_and_edit_001

Test Case Title: Verify instructors can preview and edit content before publishing.

Test Case Description: This test ensures that instructors can review their created content, make any necessary adjustments, and preview the final output before making it available to learners.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

* Instructor account is logged in.

* A new course content item (e.g., presentation, interactive module) has been created.

Test Data: Content with minor errors or areas requiring adjustments.

Test Steps:

1. Within the content authoring environment, utilize the "Preview" function to view the content as learners would see it.

2. Make specific edits to the content, correcting errors or refining elements.

3. Repeat the preview process to ensure changes are reflected correctly.

Postconditions:

* Content is edited and previewed multiple times.

Expected Result: The preview feature accurately displays the content as it will appear to learners, allowing instructors to make necessary edits and refinements before publishing.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

--

Data - 67

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Paths and Curriculum

Feature description -

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence and continuity in learners' educational journeys.

Userstory title -

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

Acceptance criteria -

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

Generated Test Cases:

Test Cases for Online Learning Platform - Learning Paths and Curriculum

Test Case ID: LPCT001

Test Case Title: Verify that learning paths include sequential courses and modules.

Test Case Description: This test case verifies the platform's functionality in structuring learning paths with sequential courses and modules, ensuring a logical progression of learning content.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

* An administrator user is logged in.
* A new learning path is created.

Test Data: Course and module details

Test Steps:

1. Navigate to the learning path creation interface.
2. Add two courses to the learning path, ensuring they belong to the same subject or topic.
3. Within each course, add several modules, arranging them in a logical order.
4. Save the learning path.
5. Access the learning path as a learner.
6. Navigate through the courses and modules, confirming they are presented in the intended sequential order.

Postconditions: Learning path is saved

Expected Result: Learners can access courses and modules in a pre-determined and sequential order, reflecting the intended learning path structure.

Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive

Test Case ID: LPCT002

Test Case Title: Verify that curricula are aligned with specific learning objectives.

Test Case Description: This test case examines if the platform allows instructors to define learning objectives for curricula and ensures that the course content aligns with these objectives.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

* An instructor user is logged in.
* A new curriculum is created.

Test Data: Curriculum learning objectives, course content

Test Steps:

1. Navigate to the curriculum creation interface.
2. Define clear and specific learning objectives for the curriculum.
3. Add related courses to the curriculum, ensuring each course directly contributes to achieving the defined learning objectives.
4. Review the curriculum content, verifying that all elements support and contribute to the learning objectives.

Postconditions: Curriculum is saved.

Expected Result: The platform allows instructors to clearly define learning objectives and ensures that the selected courses and modules align with those objectives, fostering a structured and goal-oriented learning experience.

Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive

Test Case ID: LPCT003

Test Case Title: Verify that learners can track their progress within a learning path.

Test Case Description: This test case assesses the platform's ability to accurately display learner progress within a learning path, allowing for self-assessment and motivation.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

* A learner user is logged in.
* The learner enrolls in a learning path.

Test Data: Learning path structure, learner activity

Test Steps:

1. Access the learning path as the learner.
2. Complete a few courses and modules within the learning path.
3. Check the progress bar or tracker to confirm that the completed modules and courses are reflected in the progress indication.

Postconditions: Learner progress is tracked

Expected Result: The platform accurately displays the learner's completion status for each course and module within the learning path, providing a clear visual representation of their progress. **Severity:** Major

Type of Testing: Functional
Test Case Behaviour: Positive

Test Case ID: LPCT004

Test Case Title: Verify that instructors can customize learning paths for different needs.

Test Case Description: This test case assesses the platform's flexibility in allowing instructors to tailor learning paths to address specific learner needs or learning styles.

Test Suite: Learning Paths and Curriculum

Test Priority: Medium

Preconditions:

* An instructor user is logged in.

* An existing learning path is accessible.

Test Data: Learner groups, alternative modules

Test Steps:

1. Go to the learning path configuration interface.
2. · Create two distinct learner groups based on their needs or prior knowledge (e.g., beginner vs. advanced learners).
3. Modify the modules within the learning path, assigning different sets of modules to each learner group based on their specific requirements.
4. Save the changes.

Postconditions: Learning path is updated

Expected Result: The platform allows instructors to create personalized learning paths for different learner groups, ensuring content is tailored to their individual needs and backgrounds.

Severity: Moderate

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: LPCT005

Test Case Title: Verify that notifications are sent for upcoming courses or milestones.

Test Case Description: This test case ensures that the platform effectively sends timely notifications to learners regarding upcoming courses or significant milestones within their chosen learning path.

Test Suite: Learning Paths and Curriculum

Test Priority: Medium

Preconditions:

* A learner user is logged in.
* A learner has enrolled in a learning path with upcoming courses or milestones.

Test Data: Notification settings

Test Steps:

1. Configure the learner's notification preferences to receive email or in-app notifications for upcoming courses or milestones.
2. Set a future date for an upcoming course or milestone within the learning path.
3. Observe if the learner receives a notification at the specified time.

Postconditions: Learners are notified.

Expected Result: The system successfully delivers notifications to learners about upcoming courses or milestones within their chosen learning paths, ensuring they stay informed and on track. **Severity:** Medium
Type of Testing: Functional

Test Case Behaviour: Positive

Data - 68

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Attendance Tracking

Feature description -

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

Userstory title -

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Acceptance criteria -

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

Generated Test Cases:

Here are 5 test cases for the Online Learning Platform's Attendance Tracking feature, following the provided template:

Test Case 1

Test Case ID: OLPL-ATT-001

Test Case Title: Verify that attendance is automatically recorded during live sessions.

Test Case Description: This test case aims to confirm that the system accurately registers learner attendance upon joining and remaining active in a live session.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Instructor creates a live session.
- Learners enroll in the session.
- A stable internet connection is available for both instructor and learners.

Test Data: 5-10 learners

Test Steps:

1. Instructor initiates a live session.
2. Learners join the session.
3. Learners actively participate in the session (e.g., asking questions, typing in chat).
4. After a predefined duration, the instructor ends the session.

Postconditions:

- Attendance data is uploaded to the system.

Expected Result: The system will accurately record attendance for all learners who joined and remained active throughout the live session. Attendance records should reflect their presence and engagement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: OLPL-ATT-002

Test Case Title: Verify that instructors can view attendance reports for each session

Test Case Description: This test case ensures instructors can access comprehensive attendance reports for individual sessions.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- An instructor has conducted a live session.
- Attendance data has been successfully recorded.

Test Data: Previous live session attendance data.

Test Steps:

1. Instructor logs into the platform.
2. Navigate to the "Attendance" or "Reports" section.
3. Select the specific live session from the list.
4. Review the attendance report.

Postconditions:

- Attendance report is displayed.

Expected Result: The attendance report should clearly list all learners who attended the selected session, indicating their presence and any notes about their engagement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: OLPL-ATT-003

Test Case Title: Verify that learners can see their attendance records

Test Case Description: This test case confirms that learners have access to their attendance records for previously attended sessions.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- A learner has enrolled in and attended at least one live session.

Test Data: Learner account and attendance data.

Test Steps:

1. Learner logs into the platform.
2. Navigate to the "My Profile" or "Activity" section.
3. Locate the attendance records section.
4. Review their attendance history.

Postconditions: Playback feature for missed sessions.

Expected Result: Learners should be able to view their attendance status for each session they have participated in.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: OLPL-ATT-004

Test Case Title: Verify that attendance data is integrated with participation metrics.

Test Case Description: This test case tests the integration of attendance records with other engagement metrics within the platform.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- A live session has been conducted.
- Learners have submitted quiz answers and/or participated in discussion forums.

Test Data: Live session data including attendance and engagement activities.

Test Steps:

1. Instructor logs in and navigates to the attendance report for a specific session.
2. Identify the integration points where attendance intersects with participation data (e.g., forum post count, quiz score).
3. Correlate attendance records with learner participation levels.

Postconditions:

- Participation metrics associated with attendance data.

Expected Result: Attendance information should be seamlessly linked with relevant participation metrics, providing a comprehensive view of learner engagement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: OLPL-ATT-005

Test Case Title: Verify that notifications are sent for missed sessions.

Test Case Description: This test case ensures that learners receive timely notifications when they miss a session.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- An instructor has scheduled a live session.
- A learner is enrolled in the session.

Test Data: Learner email address and notification settings.

Test Steps:

1. Instructor initiates the live session.
2. Learner does not attend the session.
3. After a designated delay (e.g., 1 hour), verify the learner's inbox for attendance notifications.

Postconditions: Notification system sends email alerts.

Expected Result: The learner should receive a notification email alerting them to their missed session. The notification should include session details and a prompt to catch up on missed content.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive/Negative

Data - 69

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Licensing and Rights Management

Feature description -

The platform implements robust content licensing and rights management features to protect intellectual property and ensure compliance with copyright laws. By managing content licenses, usage rights, and permissions, the platform safeguards the interests of content creators and providers, enabling secure and lawful distribution of educational materials.

Userstory title -

As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

Acceptance criteria -

['Content licenses are assigned to each resource.', 'Usage rights and permissions are clearly defined.', 'Content creators can set restrictions on usage and distribution.', 'Instructors and learners are informed of content licensing terms.', 'Notifications are sent for license expiration or renewal.']

Generated Test Cases:

Here are 5 test cases for the Online Learning Platform, focusing on Content Licensing and Rights Management:

Test Case ID: olp-lc-001

Test Case Title: Verify that content licenses are assigned to each resource.

Test Case Description: This test case aims to ensure that every uploaded resource (video lectures, documents, etc.) is correctly associated with a specific content license.

Test Suite: Content Licensing & Rights Management

Test Priority: High

Preconditions:

* A content creator account is created and logged in.

* A sample learning resource (e.g., a video lecture) is uploaded.

* Different content license options are available within the platform's settings.

Test Data: Sample learning resource, various content license options.

Test Steps:

1. Log in as a content creator.

2. Upload a sample learning resource.

3. Navigate to the resource settings.

4. Select a content license from the available options and save the changes.

5. Verify that the selected license is displayed against the uploaded resource.

Postconditions:

* The resource has been successfully assigned a content license.

Expected Result: The chosen content license is correctly assigned and visible in the resource's details.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-lc-002

Test Case Title: Verify that usage rights and permissions are clearly defined.

Test Case Description: This test case ensures that the platform clearly communicates the usage rights and permissions associated with each content license.

Test Suite: Content Licensing & Rights Management

Test Priority: High

Preconditions:

* A content creator account is created and logged in.
* A learning resource with a specific content license is uploaded.

Test Data: Learning resource with a defined content license.

Test Steps:

1. Log in as a learner.
2. Access the learning resource with the specific content license.
3. Review the details section of the resource for information on usage rights and permissions.
4. Verify that the terms are clearly explained and understandable.

Postconditions:

* The learner has reviewed the usage rights and permissions.

Expected Result: The content license details clearly specify the allowed usage, distribution, modification, and other relevant permissions for the resource.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-lc-003

Test Case Title: Verify content creators can set restrictions on usage and distribution.

Test Case Description: This test case verifies the platform's functionality allowing creators to customize usage rights for their content.

Test Suite: Content Licensing & Rights Management

Test Priority: Medium

Preconditions:

* A content creator account is created and logged in.
* Different restriction options are available within the content licensing settings.

Test Data: Sample learning resource, various restriction options (e.g., no commercial use, limited downloads, etc.).

Test Steps:

1. Log in as a content creator.
2. Upload a sample learning resource.
3. Navigate to the resource settings.
4. Select the desired restriction options for usage and distribution.
5. Save the changes and verify that the restrictions are applied to the resource.

Postconditions:

* The content creator has imposed restrictions on the resource's usage and distribution.

Expected Result: The chosen restrictions are successfully applied to the learning resource, limiting its usage and distribution according to the creator's specifications.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-lc-004

Test Case Title: Verify that instructors and learners are informed of content licensing terms.

Test Case Description: This test case ensures that instructors and learners receive clear notifications and information about the content licenses associated with the learning materials.

Test Suite: Content Licensing & Rights Management

Test Priority: Medium

Preconditions:

* An instructor account is created and logged in.

* A learner account is created and logged in.

* A learning course includes resources with different content licenses.

Test Data: Learning course with varying content licenses.

Test Steps:

1. Log in as an instructor and add the course to a curriculum.

2. Log in as a learner and enroll in the course.

3. Access the course content and review the information screens associated with each resource.

4. Verify that clear and concise notifications about the content license conditions are displayed to both instructors and learners.

Postconditions:

* Instructors and learners are aware of the content license information.

Expected Result: Both instructors and learners are clearly informed about the content license terms for each resource within the course, including usage rights, permissions, and restrictions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-lc-005

Test Case Title: Verify notifications are sent for license expiration or renewal.

Test Case Description: This test case checks the platform's functionality to notify content creators about approaching or expired content licenses.

Test Suite: Content Licensing & Rights Management

Test Priority: Medium

Preconditions:

* A content creator account is created and logged in.

* Content license settings allow configuring expiration dates.

* A testing email account is created and associated with the creator account.

Test Data: Sample learning resource with a set expiration date for its content license.

Test Steps:

1. Log in as a content creator.

2. Upload a sample learning resource with a predefined expiration date for the license.

3. Verify that the platform schedules notifications for approximately a week before the expiration date.

4. Wait until a notification is sent to the testing email account.

5. Log back into the platform as the content creator and confirm the approaching license expiration.

Postconditions:

* Content creator has received notification about the license expiration.

Expected Result: The platform successfully sends timely notifications to the content creator about the expiration date of their content licenses.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Data - 70

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Versioning

Feature description -

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

Userstory title -

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

Acceptance criteria -

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.', 'Notifications are sent for significant content updates.']

Generated Test Cases:

Here are 5 test cases designed to cover the acceptance criteria for the "Content Versioning" feature in the Online Learning Platform:

Test Case 1

* **Test Case ID:** OLP-CTV-001
* **Test Case Title:** Verify that instructors can create and save multiple versions of course content.
* **Test Case Description:** This test case aims to confirm that instructors have the functionality to create and save distinct versions of their course materials within the platform.
* **Test Suite:** Content Versioning
* **Test Priority:** High
* **Preconditions:**
    * Instructor account is created and logged in.
    * A new course is created by the instructor.
    * Content (e.g., lecture notes, slides) is added to the course.
* **Test Data:** Updated content for each new version
* **Test Steps:**
    1. Access the course created.
    2. Edit the existing content in the course.
    3. Through the version control feature, save a new version of the content.
    4. Repeat steps 2 & 3 for at least two additional versions.
* **Postconditions:**
    * Multiple versions of the content are saved within the course.
    * Version history is visible in the course content management dashboard.
* **Expected Result:** The system successfully creates and saves multiple distinct versions of the course content, with clear version history tracking.
* **Severity:** Major

* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** OLP-CTV-002
* **Test Case Title:** Verify that instructors can view and compare different versions of content.
* **Test Case Description:** This test aims to confirm that instructors can easily compare two different versions of a particular course content element side-by-side.
* **Test Suite:** Content Versioning
* **Test Priority:** High
* **Preconditions:**
    * Scenario setup from Test Case 1 is in place.
* **Test Data:** Versions of the content created in Test Case 1
* **Test Steps:**
    1. Navigate to the specific content within the course.
    2. Access the version history feature.
    3. Select two distinct versions of the content to compare.
* **Postconditions:**
    * A version comparison view is displayed.
* **Expected Result:** The system presents a clear and concise side-by-side comparison of the selected versions, highlighting the differences made in each revision.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** OLP-CTV-003
* **Test Case Title:** Verify previous versions of content can be restored.
* **Test Case Description:** This test case verifies that instructors can revert to a specific previous version of a course content if needed.
* **Test Suite:** Content Versioning
* **Test Priority:** High
* **Preconditions:**
  * Scenario setup from Test Case 1 is in place.
  * The current version of the content is made intentionally flawed (e.g., typo, incorrect information).
* **Test Data:** The flawed current version of the content and the desired previous version.
* **Test Steps:**
  1. Access the course content.
  2. Using the version history, select the previous, intact version (before the flaw was introduced).
  3. Initiate the content restoration process to replace the current version.
* **Postconditions:**
  * The content is updated with the selected previous version.
* **Expected Result:** The restored version successfully replaces the current version, correcting the previously introduced flaw.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** OLP-CTV-004
* **Test Case Title:** Verify that notifications are sent for significant content updates. (This is an advanced

requirement)

* **Test Case Description:** This test case utilizes notification functionality to check if the platform informs instructors about substantial changes made to course content.

* **Test Suite:** Content Versioning

* **Test Priority:** Medium

* **Preconditions:**

    * Instructor account with email notification preferences enabled.

    * An existing course with some content owned by the instructor.

* **Test Data:** Modified content

* **Test Steps:**

    1. Make a significant modification to one of the course content elements (e.g., adding a completely new section, deleting substantial portions).

    2. Save the changes.

* **Postconditions:**

    * Confirmed email notification received by the instructor

* **Expected Result:** The instructor receives a notification email alerting them to the significant content update that occurred.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** OLP-CTV-005

* **Test Case Title:** Verify system handling of deleted content versions. (This tests a negative scenario)

* **Test Case Description:** This test case evaluates the system's behavior when a previously modified

content version is deleted.

* **Test Suite:** Content Versioning

* **Test Priority:** Medium

* **Preconditions:**

    * Scenario setup from Test Case 1 is in place.

* **Test Data:** A version created in Test Case 1.

* **Test Steps:**

    1. Access the version history for the course content.

    2. Attempt to permanently delete one of the saved versions.

* **Postconditions:**

    * Deleted version is no longer visible in the version history.

* **Expected Result:** The system successfully removes the selected version from the history. No data loss occurs; the system retains other versions. Furthermore, it should prevent accidental deletion of critical versions (e.g., the most recent version) to maintain course integrity.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Negative

Data - 71

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

API Integration

Feature description -

The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

Userstory title -

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

Acceptance criteria -

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

Generated Test Cases:

Here are 5 test cases for the provided input data, following the template you've outlined:

Test Case 1

* **Test Case ID:** API_INT_001

* **Test Case Title:** Verify that APIs for key platform functionalities are available

* **Test Case Description:** This test case aims to verify the existence and accessibility of APIs for core platform functionalities, such as user authentication, course enrollment, and quiz submission.

* **Test Suite:** API Integration

* **Test Priority:** High

* **Preconditions:**

    * API documentation is accessible

    * A testing environment with API access is set up

* **Test Data:** List of expected key API endpoints (e.g., `/api/v1/users`, `/api/v1/courses`, `/api/v1/quizzes`)

* **Test Steps:**

    1. Navigate to the platform's API documentation.

    2. Identify the listed API endpoints for key functionalities.

    3. Attempt to access each API endpoint using a suitable API testing tool (e.g., Postman).

    4. Verify that each endpoint returns a valid response code (e.g., 200 OK) indicating success.

* **Postconditions:** API endpoints are documented and accessible.

* **Expected Result:** All identified key API endpoints are accessible, returning appropriate responses.

* **Severity:** Critical

* **Type of Testing:** API Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** API_DOC_001

* **Test Case Title:** Verify that comprehensive API documentation is provided

* **Test Case Description:** This test case evaluates the comprehensiveness and clarity of the API documentation, ensuring developers have the necessary information to integrate with the platform effectively.

* **Test Suite:** API Integration

* **Test Priority:** Medium

* **Preconditions:**

    * API documentation is accessible

* **Test Data:** List of expected documentation elements (e.g., API endpoints, authentication methods, data

formats, error codes, usage examples)

* **Test Steps:**

  1. Access the platform's API documentation.

  2. Review the documentation for each accepted API endpoint.

  3. Verify that each documentation section includes clear descriptions, examples, and specifications for:

     * Endpoint URL

     * Supported HTTP methods

     * Request parameters

     * Response structure

     * Authentication requirements

* **Postconditions:** API documentation is readily available and understandable.

* **Expected Result:** API documentation is comprehensive, well-structured, and provides sufficient information for developers to understand and utilize the APIs.

* **Severity:** Major

* **Type of Testing:** Documentation Review

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** API_AUTH_001

* **Test Case Title:** Verify that third-party applications can authenticate successfully

* **Test Case Description:** This test case ensures that developers can securely authenticate third-party applications to access the platform's APIs.

* **Test Suite:** API Integration

* **Test Priority:** High

* **Preconditions:**

  * API documentation outlining authentication methods is provided

    * A test application or tool capable of making API calls and handling authentication is available
* **Test Data:** Valid API credentials (API key or token)
* **Test Steps:**
    1. Using the test application or tool, initiate a request to a protected API endpoint.
    2. Include the valid API credentials in the request header or body.
    3. Verify that the API endpoint responds with an appropriate success code (e.g., 200 OK) only when using valid credentials.
* **Postconditions:** Authentication mechanism is effectively implemented.
* **Expected Result:** The API endpoint only grants access when presented with valid API credentials. Unauthorized requests should be rejected.
* **Severity:** High
* **Type of Testing:** Security Testing, API Testing
* **Test Case Approach:** Positive/Negative

Test Case 4

* **Test Case ID:** API_RATE_LIMITING_001
* **Test Case Title:** Verify the presence of rate limiting and monitoring features
* **Test Case Description:** This test case checks if the APIs have implemented rate limiting to prevent abuse and ensure system stability. It also tests if there are monitoring mechanisms in place to track API usage.
* **Test Suite:** API Integration
* **Test Priority:** Medium
* **Preconditions:**
    * Knowledge of the platform's API rate limit policy
    * A test application capable of making multiple API calls within a short time frame
* **Test Data:** API endpoints subjected to rate limiting.
* **Test Steps:**

1. Use the test application to send a series of API requests to the rate-limited endpoint within a short time frame.
2. Monitor the server's responses and observe any rate limit violations or throttling mechanisms.
3. Within the platform's dashboard (if applicable), verify the availability of API usage monitoring reports and logs.

* **Postconditions:** Rate limiting effectiveness is confirmed.
* **Expected Result:**
    * Upon exceeding the rate limit, the API returns a specific error code or status indicating rate limiting.
    * The platform's dashboard shows logs or reports detailing API usage, including request count, successful/failed requests, and timestamps.
* **Severity:** Major
* **Type of Testing:** API Testing, Load Testing
* **Test Case Approach:** Positive/Negative

Test Case 5:

* **Test Case ID:** API_SEC_001
* **Test Case Title:** Verify secure access protocols are implemented
* **Test Case Description:** This test case assesses the security measures in place to protect API communication and prevent unauthorized access to sensitive data.
* **Test Suite:** API Integration
* **Test Priority:** High
* **Preconditions:**
    * API documentation specifies the authentication methods used (e.g., OAuth, API keys).
* **Test Data:**
* **Test Steps:**
    1. Attempt to access protected API endpoints without valid credentials.

2. Monitor the responses for appropriate security-related error codes.

3. (If applicable) Test different authentication methods to ensure they are implemented correctly and securely.

* **Postconditions:** Secure access is enforced.

* **Expected Result:** API endpoints protected with authentication mechanisms should only grant access to authenticated requests.

  * Authorization failures should result in specific error codes, and no sensitive data should be exposed without proper authorization.

* **Severity:** Critical

* **Type of Testing:** Security Testing, API Testing

* **Test Case Approach:** Positive/Negative

Data - 72

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

Here are 5 test cases for the provided input data, following the specified template:

Test Case 1

Test Case ID: aln9D3

Test Case Title: Verify that data visualization tools are available for analyzing learner performance.

Test Case Description: This test case verifies if the platform provides visual tools for administrators to analyze learner performance metrics, such as completion rates, quiz scores, and progress through modules.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in.
- At least one course is available with learner activity.

Test Data: Learner performance data from multiple courses

Test Steps:

1. Navigate to the "Analytics" or "Reports" section of the platform.

2. Select "Learner Performance" or a similar data visualization option.

3. Explore available charts, graphs, and dashboards.

4. Filter the data by course, learner, or other relevant parameters.

5. Examine the visualized data for clarity, accuracy, and insightful representation.

Postconditions:

- Administrator views various learner performance metrics in visual format.

Expected Result: The platform displays learner performance data through interactive visualizations, allowing administrators to easily understand trends and identify areas for improvement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: zF8k2m

Test Case Title: Verify that reports can be generated for learner performance metrics.

Test Case Description: This test case checks if the platform allows administrators to generate reports containing specific learner performance metrics for analysis and reporting purposes.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in.

- A course with sufficient learner activity exists.

Test Data: Time period for report generation (e.g., last month)

Test Steps:

1. Access the "Reports" section of the platform.

2. Select "Learner Performance" or a similar report template.

3. Choose the desired time period for the report data.

4. Customize the report by including specific metrics (e.g., quiz scores, module completion rates)

5. Generate the report in a suitable format (e.g., PDF, CSV, Excel).

Postconditions:

- Report containing learner performance data is generated.

Expected Result: The platform generates a comprehensive report with learner performance data, allowing administrators to analyze trends and identify areas for improvement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: v9xG75

Test Case Title: Verify real-time analytics provide insights into learner engagement and outcomes

Test Case Description: This test case verifies that the platform provides real-time dashboards or updates on learner engagement and outcomes, allowing administrators to monitor platform activity instantly.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator is logged in.

- Several learners are actively interacting with the platform (e.g., viewing lectures, participating in forums).

Test Data: No specific data required, real-time platform activity is utilized.

Test Steps:

1. Access the "Dashboard" or "Real-time Analytics" section of the platform.

2. Observe learner engagement metrics, such as the number of active users, course views, forum posts, and

quiz attempts.

3. Monitor these metrics for updates and changes over a short period.

4. Note any trends or insights gained from the real-time data.

Postconditions:

- Administrator analyzes real-time data on learner engagement.

Expected Result: The platform displays real-time analytics, providing immediate insights into learner activity and allowing administrators to respond to changing user behaviors promptly.

Severity: Major

Type of Testing: Functional, Performance

Test Case Approach: Positive

Test Case 4

Test Case ID: u7bL1w

Test Case Title: Verify that data can be exported for further analysis

Test Case Description: This test case confirms if the platform allows administrators to export data for offline analysis or integration with external tools.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator is logged in.
- User data (e.g., performance, activity) is available within the platform.

Test Data: No specific data required; the export functionality is tested.

Test Steps:

1. Navigate to the "Data Export" or "Download Data" section within the analytics area.

2. Select the desired data type to export (e.g., learner performance, course completion rates).

3. Choose the export format (e.g., CSV, Excel, JSON).

4. Specify the date range or filter criteria for the exported data.

5. Initiate the data export process.

Postconditions:

- Exported data file is downloaded.

Expected Result: The platform successfully exports the selected data in the chosen format, enabling administrators to analyze it using external tools or for offline processing.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: a5kRj9

Test Case Title: Verify notification system for significant data trends or anomalies.

Test Case Description: This test case checks if the platform sends notifications to administrators when significant data trends are identified or unusual anomalies occur.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator email address is configured in the platform settings.
- Simulated or real data events that meet customizable notification thresholds are generated.

Test Data: No specific data required; notification triggers are tested.

Test Steps:

1. Simulate or observe a data event that should trigger a notification (e.g., sudden drop in quiz scores, spike in video views, unusual login patterns).

2. Check the administrator's email inbox for a notification.

3. Verify the notification content includes relevant details about the event, such as impacted courses, learners, or metrics.

Postconditions:

- Administrator receives a notification.

Expected Result: The platform sends timely and informative notifications to administrators when significant data trends or anomalies occur, enabling them to investigate and address potential issues effectively.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 73

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Roles and Permissions

Feature description -

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

Userstory title -

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

Acceptance criteria -

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

Generated Test Cases:

Here are 5 test cases for the provided input data, following the template you've outlined:

Test Case 1

Test Case ID: rolMgmt-001

Test Case Title: Verify that administrators can create new user roles.

Test Case Description: This test case verifies the ability of administrators to define and create new user roles within the platform, ensuring the system supports the creation of diverse user types.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- An administrator account is logged in.

Test Data: Descriptive role names (e.g., "Instructor", "Student", " Moderator")

Test Steps:

1. Navigate to the user management section.

2. Locate the "Create Role" button or option.

3. Enter a descriptive role name.

4. Optionally, assign initial permissions to the role.

5. Save the newly created role.

Postconditions:

- A new role with the specified name is displayed in the list of roles.

Expected Result: The system successfully creates the new user role with the provided name.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: permCust-002

Test Case Title: Verify that permissions are customizable for each role.

Test Case Description: This test case ensures that administrators have granular control over assigning permissions to each user role, allowing for fine-grained access management.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- An administrator account is logged in.

- A pre-defined user role exists (e.g., "Student").

Test Data: Various permissions (e.g., "Create Courses", "View Grades", "Download Resources")

Test Steps:

1. Access the role management section.

2. Select the existing user role.

3. Navigate to the permissions section for the role.

4. Enable or disable specific permissions for the role.

5. Save the changes to the role's permissions.

Postconditions:

- Changes to permissions are reflected in the role's settings.

Expected Result: The system successfully modifies the permissions assigned to the selected user role.

Test Case 3

Test Case ID: accessRest-003

Test Case Title: Verify that users are restricted to functionalities based on their roles.

Test Case Description: This test case confirms that users are only granted access to features and functions permitted by their assigned user roles, upholding the principles of role-based access control.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- Multiple user accounts with different roles (e.g., administrator, student, instructor) are created.

Test Data: Feature functionalities (e.g., course creation, grade viewing, forum posting)

Test Steps:

1. Log in as a user with a specific role (e.g., student).

2. Attempt to access various platform functionalities.

3. Observe whether access is granted or restricted based on the role.

Postconditions:

- Access behavior based on user roles is observed.

Expected Result: Users are permitted to access only the functionalities designated for their assigned roles.

Unauthorized access is denied.

Test Case 4

Test Case ID: roleLogs-004

Test Case Title: Verify that changes to roles and permissions are logged.

Test Case Description: This test case ensures that the system maintains a record of all modifications made to user roles and permissions, facilitating auditing and accountability.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- An administrator account is logged in.

Test Data: Role name, permission changes

Test Steps:

1. Create a new user role.
2. Modify the permissions of an existing role.
3. Delete a user role.
4. Check the system logs for recorded changes related to the roles and permissions.

Postconditions:

- Audit logs are accessible.

Expected Result: The system logs all modifications made to user roles and permissions, providing a comprehensive audit trail.

Test Case 5

Test Case ID: userNotifs-005

Test Case Title: Verify that users are notified of their assigned roles and permissions.

Test Case Description: This test case ensures that users receive clear and timely notifications when they are assigned roles and permissions, enhancing awareness and transparency.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- A new user account is created.

Test Data: User account information

Test Steps:

1. Assign a user role to the new user account.

2. Verify that the user receives a notification about their assigned role and permissions.

3. (Optional) Monitor the notification channels used (e.g., email, in-app messages)

Postconditions:

- Notification is received by the user.

Expected Result: The system successfully notifies the new user about their assigned role and permissions through the appropriate notification channels.

Data - 74

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Feedback and Rating System

Feature description -

The platform includes a user feedback and rating system, allowing learners to provide feedback on courses, instructors, and learning materials. This feature enables continuous improvement by gathering user insights and ratings, helping instructors refine their content and approach.

Userstory title -

As a learner, I want to provide feedback and ratings so that I can share my experience and help improve the platform.

Acceptance criteria -

['Learners can rate courses and instructors on a scale.', 'Feedback forms are available for detailed comments.', 'Instructors can view and respond to feedback.', 'Aggregated ratings are displayed publicly.', 'Anonymous feedback options are available.']

Generated Test Cases:

Test Cases for Online Learning Platform - User Feedback and Rating System

Here are 5 test cases for the User Feedback and Rating System feature of the Online Learning Platform:

Test Case 1:

Test Case ID: OLP-FBR-001

Test Case Title: Verify that learners can rate courses and instructors on a scale.

Test Case Description: This test case aims to ensure learners can successfully provide numerical ratings for courses and instructors using the provided rating scale.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

* Learner is logged into the platform
* A course with an instructor is available

Test Data: Course ID, Instructor ID

Test Steps:

1. Navigate to the course page.
2. Locate the rating section for the course and select a rating from the scale (e.g., 1-5 stars).
3. Verify the selected rating is displayed on the page.
4. Navigate to the instructor's profile page.
5. Locate the rating section for the instructor and select a rating from the scale (e.g., 1-5 stars).
6. Verify the selected rating is displayed on the page.

Postconditions:

* Rating is saved for both the course and the instructor.

Expected Result: Learners can successfully select and submit ratings within the given scale for both courses and instructors.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLP-FBR-002

Test Case Title: Verify that feedback forms are available for detailed comments.

Test Case Description: This test case ensures learners can access feedback forms and submit detailed comments regarding courses, instructors, and learning materials.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

* Learner is logged into the platform
* A course is available

Test Data: Feedback content

Test Steps:

1. Navigate to the course page
2. Locate the feedback form section
3. Enter detailed feedback in the provided fields
4. Submit the feedback form

Postconditions:

* Feedback is submitted successfully.

Expected Result: Learners can access and utilize feedback forms to provide detailed written comments regarding courses, instructors, and learning materials.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case 3:

Test Case ID: OLP-FBR-003
Test Case Title: Verify that instructors can view and respond to feedback.
Test Case Description: This test case verifies instructors' ability to access and respond to learner feedback submitted through the platform.

Test Suite: User Feedback and Rating System
Test Priority: High
Preconditions:

* Instructor is logged into the platform
* Feedback has been submitted by a learner for a course the instructor teaches.

Test Data: Course ID
Test Steps:

1. Navigate to the course dashboard.

2. Locate the feedback section.

3. View the submitted feedback from the learner.

4. Compose a response to the feedback.

Postconditions:

* Instructor has viewed and responded to the learner feedback

Expected Result: Instructors can successfully access, read, and respond to learner feedback provided through the platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLP-FBR-004

Test Case Title: Verify that aggregated ratings are displayed publicly.

Test Case Description: This test case ensures that the platform displays aggregated ratings for courses and instructors publicly, allowing learners to see the overall sentiment and quality of offerings.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

* Multiple learners have rated a course and/or an instructor.

Test Data: Course ID, Instructor ID

Test Steps:

1. Navigate to the course page.
2. Locate the aggregated rating section for the course.
3. Verify the average rating is displayed along with the total number of ratings.
4. Navigate to the instructor's profile page.
5. Locate the aggregated rating section for the instructor.
6. Verify the average rating is displayed along with the total number of ratings.

Postconditions:

* Aggregated ratings are clearly displayed for courses and instructors.

Expected Result: The platform accurately displays the aggregated ratings for courses and instructors, allowing learners to assess the overall quality and popularity of offerings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP-FBR-005

Test Case Title: Verify that anonymous feedback options are available.

Test Case Description: This test case verifies learners can choose to provide feedback anonymously, ensuring their privacy and potentially encouraging more honest and candid feedback.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

* Learner is logged into the platform
* A course is available

Test Data: Feedback content

Test Steps:

1. Navigate to the course page.
2. Locate the feedback form section.
3. Select the "Anonymous Feedback" option.
4. Enter feedback content and submit the form.

Postconditions:

* Feedback is submitted successfully and marked as anonymous.

Expected Result: Learners have the option to submit feedback anonymously, providing a level of privacy and encouraging open and honest feedback.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 75

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Media Integration

Feature description -

The platform supports social media integration, allowing learners and instructors to share their achievements, courses, and updates on social media platforms. This feature promotes engagement and community building, enhancing the visibility of the platform's offerings.

Userstory title -

As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

Acceptance criteria -

['Learners can connect their social media accounts to the platform.', 'Achievements and course completions can be shared on social media.', 'Privacy settings control the visibility of shared content.', 'Instructors can promote courses and updates through social media.', 'Analytics track the reach and engagement of shared content.']

Generated Test Cases:

Here are 5 test cases for the provided input data, following the structure and guidelines:

Test Case 1

Test Case ID: LMS-SOC-1

Test Case Title: Verify that learners can connect their social media accounts to the platform.

Test Case Description: This test case verifies the functionality of connecting learners' social media accounts (Facebook, Twitter, LinkedIn) to their profiles on the Online Learning Platform.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- User is a registered learner on the platform.

Test Data: Valid social media login credentials (Facebook, Twitter, LinkedIn)

Test Steps:

1. Log in as a learner.
2. Navigate to the "Profile" settings.
3. Click on the "Connect Social Media" option.
4. Select the desired social media platform (e.g., Facebook).
5. Enter the required login credentials for the chosen platform.
6. Authorize the platform to access the user's social media account.

Postconditions:

- User's social media account is successfully linked to their platform profile.

Expected Result: The system successfully connects the learner's social media account to their profile, displaying a confirmation message.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: LMS-SOC-2

Test Case Title: Verify that achievements and course completions can be shared on social media.

Test Case Description: This test case ensures that learners can share their achievements, such as course completions, quizzes scores, and forum participation, on their connected social media accounts.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- User is a registered learner with connected social media accounts.
- User has completed a course or achieved a specific milestone.

Test Data: Course completion data, quiz scores, forum postings

Test Steps:

1. Log in as learner.
2. Access the "My Achievements" section.
3. Select the desired achievement to share (e.g., course completion certificate).
4. Choose the platform on which to share (e.g., Facebook, Twitter).
5. Modify the message (optional) and click "Share."

Postconditions:

- The achievement is shared on the selected social media platform.

Expected Result: The achievement is successfully shared on the selected social media platform, including relevant details and a link to the Online Learning Platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: LMS-SOC-3

Test Case Title: Verify that privacy settings control the visibility of shared content.

Test Case Description: This test case checks if learners can adjust privacy settings to control the visibility of shared content on their social media accounts.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- User is a registered learner with connected social media accounts.

Test Data: Privacy setting options (Public, Friends Only, Private)

Test Steps:

1. Log in as learner.
2. Access "Profile" settings.
3. Select the "Privacy Settings" section.
4. Choose different privacy settings for shared content (Public, Friends Only, Private).
5. Attempt to share an achievement with each privacy setting.

Postconditions:

- Shared content's visibility aligns with the chosen privacy setting.

Expected Result: Sharing content with different privacy settings results in the content being published according to the selected option.

Severity: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: LMS-SOC-4

Test Case Title: Verify that instructors can promote courses and updates through social media.

Test Case Description: This test case validates the ability of instructors to promote their courses and share updates with their students and followers on social media platforms.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- User is an instructor with connected social media accounts.
- An active course or update is available.

Test Data: Course details, promotional message, update content

Test Steps:

1. Log in as instructor.
2. Navigate to the "My Courses" section.
3. Select a course or access the "Course Updates" section.
4. Choose the "Share on Social Media" option.

5. Select the desired social media platform and platform-specific message format (pre-written or custom).

6. Add relevant course details or update information.

7. Click "Post" or "Share."

Postconditions:

- The course promotion or update is published on the chosen social media platform.

Expected Result: Instructors can successfully share course promotions and updates on connected social media platforms, including necessary details and engaging content.

Severity: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: LMS-SOC-5

Test Case Title: Verify that analytics track the reach and engagement of shared content.

Test Case Description: This test case confirms that the platform monitors and reports key metrics about shared content performance on social media.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- User has connected their social media accounts to the platform.

- User has shared at least one course-related achievement or update.

Test Data:

Test Steps:

1. Log in as instructor or learner.

2. Access the platform's "Analytics" or "Insights" section.

3. Navigate to the social media analytics dashboard.

4. Verify that the dashboard displays metrics such as total shares, reach, likes, comments, and clicks for the shared content.

Postconditions:

- Data about shared content is available in the analytics dashboard.

Expected Result: The analytics dashboard provides clear and comprehensive information about the reach, engagement, and performance of shared content on social media platforms.

Severity: Major

Type of Testing: Non-Functional Testing

Test Case Approach: Positive/Regression

Data - 76

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Activity Tracking

Feature description -

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity levels and progress towards fitness goals.

Userstory title -

As a user, I want to track my activities so that I can monitor my daily physical activity levels.

Acceptance criteria -

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']

Generated Test Cases:

Fitness Tracker App Test Cases - Activity Tracking Feature

Here are 5 test cases for the Activity Tracking feature of the Fitness Tracker App, each covering one or more acceptance criteria:

Test Case 1

* **Test Case ID:** FT-ACT-001
* **Test Case Title:** Verify that the app records data on distance, duration, pace, and calories burned for running activities.
* **Test Case Description:** This test case validates the system's ability to accurately capture and store data points (distance, duration, pace, calories burned) for running activities logged by the user.
* **Test Suite:** Activity Tracking

* **Test Priority:** High

* **Preconditions:**

    * User is logged in to the app.

    * User has granted access to location services.

* **Test Data:** Pre-defined running activity data (distance, duration, pace).

* **Test Steps:**

    1. Start a running activity in the app.

    2. Run for a predetermined duration.

    3. Stop the running activity in the app.

    4. Navigate to the activity log.

    5. Verify the recorded distance, duration, pace, and calories burned match the pre-defined data.

* **Postconditions:**

    * Running activity is logged in the user's activity history.

* **Expected Result:** The app accurately records all specified data points for the running activity.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** FT-ACT-002

* **Test Case Title:** Verify that users can view activity summaries for different activity types.

* **Test Case Description:** This test case ensures the app displays comprehensive summaries of user activity for various activity types (running, walking, cycling, swimming).

* **Test Suite:** Activity Tracking

* **Test Priority:** Medium

* **Preconditions:**

   * User is logged in to the app.

   * User has logged multiple activities of different types (e.g., running, walking).

* **Test Data:** Logged activities from different categories.

* **Test Steps:**

   1. Navigate to the activity summary section.

   2. Select a specific activity type (e.g., running).

   3. Verify the summary displays relevant data points (total distance, duration, average pace, calories burned) for that activity type.

* **Postconditions:**

   * The activity summary window is displayed.

* **Expected Result:** The app accurately generates summaries with relevant data points for each selected activity type.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** FT-ACT-003

* **Test Case Title:** Verify that activity data is stored for historical analysis.

* **Test Case Description:** This test case ensures the app properly maintains a record of past activities, allowing users to track their progress over time.

* **Test Suite:** Activity Tracking

* **Test Priority:** Medium

* **Preconditions:**

   * User is logged in to the app.

   * User has logged several activities over a period.

* **Test Data:** Past activity data from different dates.

* **Test Steps:**

    1. Navigate to the historical activity data section.

    2. Select a specific date range.

    3. Verify the app displays the activities logged within that date range.

* **Postconditions:**

    * Historical activity data is displayed.

* **Expected Result:** The app accurately retrieves and displays past activity data for the selected date range.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** FT-ACT-004

* **Test Case Title:** Verify the ability to pause and resume activity tracking.

* **Test Case Description:** This test case ensures the app appropriately handles the pausing and resuming of ongoing activity sessions.

* **Test Suite:** Activity Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in to the app.

    * User has started an ongoing activity.

* **Test Data:** Simulated activity duration for pausing and resuming.

* **Test Steps:**

    1. Start a running activity.

    2. Pause the activity at a specific point in time.

   3. Resume the activity after a short interval.
   4. Stop the activity and review recorded data.
* **Postconditions:**
   * Activity is paused and resumed successfully.
* **Expected Result:** The app accurately captures the elapsed time, distance, pace, and calories burned despite the pause and resume actions.
* **Severity:** Critical
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** FT-ACT-005
* **Test Case Title:** Verify that users receive notifications for achieving fitness milestones.
* **Test Case Description:** This test case validates the functionality of milestone-based notifications, ensuring users are alerted to significant progress in their fitness journey.
* **Test Suite:** Activity Tracking
* **Test Priority:** High
* **Preconditions:**
   * User is logged in to the app.
   * User has set a specific fitness milestone (e.g., steps per day).
   * Notification settings are enabled for relevant milestones.
* **Test Data:** Predefined fitness milestones.
* **Test Steps:**
   1. Log activities to reach predefined milestones.
   2. Close the app and wait for the expected notification trigger time.

    3. Check for the notification alerting the user about their fitness milestone achievement.

* **Postconditions:**

    * Notification is displayed

* **Expected Result:** The user receives a timely and relevant notification upon reaching predetermined milestones.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Data - 77

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

GPS Tracking

Feature description -

Integrated GPS tracking allows users to accurately track outdoor activities and routes, providing detailed maps, elevation profiles, and location-based metrics. Users can visualize their workout routes, analyze performance data, and explore new routes for added variety and challenge.

Userstory title -

As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

Acceptance criteria -

['GPS tracking accurately records route data.', 'Users can view detailed maps and elevation profiles.', 'Performance data includes location-based metrics.', 'Routes can be saved and revisited.', 'Notifications are provided for deviations from planned routes.']

Generated Test Cases:

Here are 5 test cases for the GPS Tracking feature of the Fitness Tracker App, based on the provided acceptance criteria.

Test Case 1:

* Test Case ID: gps-route-accuracy
* Test Case Title: Verify that GPS tracking accurately records route data for outdoor activities
* Test Case Description: This test case aims to confirm that the GPS tracking feature accurately captures the user's route during an outdoor activity. This will be accomplished by comparing the recorded route data with a known, accurate map of the area.

* Test Suite: GPS Tracking
* Test Priority: High
* Preconditions:
    - User is logged in to the Fitness Tracker app
    - User has granted location permissions to the app.

  - A mobile device with a GPS receiver is used.

* Test Data: A pre-defined outdoor route map

* Test Steps:

  1. Start a new outdoor activity session (e.g., running, walking, cycling) within the app while outside.

  2. Follow the pre-defined route map closely.

  3. Complete the outdoor activity session.

  4. Review the recorded route data within the app.

  5. Compare the recorded route in the app with the known accurate map of the predefined route.

* Postconditions:

  - The recorded route data is saved in the app.

  - Data including distance, time, and pace are displayed.

* Expected Result: The recorded route data accurately reflects the path traveled along the pre-defined route on the map, with minimal deviations.

* Severity: Critical

* Type of Testing: System Testing, Functional Testing

* Test Case Approach: Positive

Test Case 2:

* Test Case ID: gps-elevation-profile

* Test Case Title: Verify that detailed elevation profiles are displayed for outdoor activities.

* Test Case Description: This test case checks if the app accurately generates and presents elevation profiles alongside recorded routes for outdoor activities.

* Test Suite: GPS Tracking

* Test Priority: High

* Preconditions:

   - User is logged in to fitness tracker app.

   - Location permissions granted.

   - Device with GPS and access to elevation data (may require internet connection).

* Test Data: A pre-defined route with varying elevations.

* Test Steps:

   1. Start an outdoor activity (hiking, running, cycling).

   2. Follow the pre-defined route.

   3. Complete the activity and review the route details.

   4. Examine the elevation profile displayed alongside the route map.

* Postconditions:

   - Elevation profile data is shown for the recorded route.

* Expected Result: The elevation profile accurately depicts the ascents and descents along the pre-defined route.

* Severity: Major

* Type of Testing: System Testing, Functional Testing

* Test Case Approach: Positive

Test Case 3:

* Test Case ID: gps-location-metrics

* Test Case Title: Verify that Performance data includes location-based metrics.

* Test Case Description: This test case focuses on confirming that the app calculates and displays location-based performance metrics during outdoor activities.

* Test Suite: GPS Tracking

* Test Priority: High

* Preconditions:

    - User is logged in.

    - Location permissions enabled for the app.

* Test Data: Various outdoor activity sessions with different locations and distances.

* Test Steps:

    1. Start a new outdoor activity.

    2. Follow a designated route for the activity.

    3. View the detailed performance metrics provided after completing the activity.

    4. Pay attention to location-specific metrics like average speed in different areas, distance traveled, elevation gain/loss.

* Postconditions:

    - Performance data for the activity is saved.

* Expected Result: The app accurately presents various location-based metrics like average speed, distance, elevation, and changes in these metrics based on location along the recorded route.

* Severity: Major

* Type of Testing: System Testing, Functional Testing

* Test Case Approach: Positive

Test Case 4:

* Test Case ID: gps-route-save-revisit

* Test Case Title: Verify that GPS-tracked routes can be saved and revisited.

* Test Case Description: This test case confirms that users are able to save a GPS-tracked route and successfully retrieve it for future use (e.g., to retrace a path for another session).

* Test Suite: GPS Tracking

* Test Priority: Medium

* Preconditions:
    - User is logged in.
    - Location permissions are granted.
* Test Data:
    A new GPS-tracked route during an outdoor activity.
* Test Steps:
    1. Start an outdoor activity, recording a GPS route.
    2. Upon completing the activity, save the recorded route within the app.
    3. Later, from the app's history or map view, find and select the saved route.
* Postconditions:
    - Saved route is accessible within the app.
* Expected Result: The user can successfully view and "revisit" the saved GPS route, allowing them to retrace the path.
* Severity: Minor
* Type of Testing: System Testing, Functional Testing
* Test Case Approach: Positive

Test Case 5:

* Test Case ID: gps-route-deviation
* Test Case Title: Verify notifications are provided for deviations from planned routes.
* Test Case Description: This test case evaluates the functionality of notifications that alert users when their actual path deviates from a planned route.

* Test Suite: GPS Tracking
* Test Priority: Medium

* Preconditions:

  - User is logged in.
  - Location permissions granted.

* Test Data: A pre-planned route within the app.

* Test Steps:

  1. Create a new planned route using the app's route-planning features.
  2. Start an outdoor activity while following the planned route.
  3. Intentionally deviate from the planned route during the activity.

* Postconditions:

  - The app records the deviation.

* Expected Result: The app detects the route deviation and provides an appropriate notification to the user, informing them of the change from the planned path.

* Severity: Moderate

* Type of Testing: System Testing, Functional Testing

* Test Case Approach: Positive

--

Data - 78

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Heart Rate Monitoring

Feature description -

The app monitors heart rate in real-time using built-in sensors or compatible wearable devices, providing insights into exercise intensity, recovery, and cardiovascular health. Heart rate data helps users optimize their workouts, maintain target heart rate zones, and track improvements in fitness over time.

Userstory title -

As a user, I want to monitor my heart rate so that I can optimize my workouts and track my cardiovascular health.

Acceptance criteria -

['Heart rate data is recorded in real-time.', 'Users can set and monitor target heart rate zones.', 'Historical heart rate data is stored and viewable.', 'Alerts are provided for abnormal heart rate readings.', 'Data is compatible with various wearable devices.']

Generated Test Cases:

Okay, here are five test cases for the Fitness Tracker App's Heart Rate Monitoring feature, designed to cover all the acceptance criteria.

Fitness Tracker App - Heart Rate Monitoring Test Cases

Test Case 1: Verify that heart rate data is recorded in real-time

Test Case ID: HRT-01

Test Case Title: Verify real-time heart rate data recording

Test Case Description: This test case verifies the app accurately captures and displays heart rate data in real-time during physical activity.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

  * User is logged in to the app.

  * GPS is enabled (or simulated) for distance tracking.

  * Bluetooth is enabled and a compatible wearable device is connected.

Test Data: User's heart rate fluctuations during various exercise intensities.

Test Steps:

  1. Start the workout tracking feature.

  2. Perform exercises that increase heart rate (e.g., running, cycling, jumping jacks).

  3. Observe the real-time heart rate display in the app.

  4. Note the heart rate fluctuations and ensure they correspond reasonably to the exercise intensity.

Postconditions:

  * Workout data is recorded.

Expected Result: The app accurately and consistently records the user's heart rate data in real-time as they engage in physical activity.

Severity: Critical

Type of Testing: Functional, Performance

Test Case Approach: Positive, Real-time

Test Case 2: Verify setting and monitoring of target heart rate zones

Test Case ID: HRT-02

Test Case Title: Verify setting and monitoring of target heart rate zones

Test Case Description: This test case checks the functionality of setting custom target heart rate zones and the app's ability to display whether the user is within or outside those zones during exercise.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

   * User is logged in to the app.

   * A heart rate source is connected (e.g., wearable device).

Test Data: User's age, weight, and fitness level to calculate target heart rate zones.

Test Steps:

   1. Access the heart rate settings within the app.

   2. Input user's age, weight, and desired target heart rate zone percentages (e.g., 60-80% of maximum heart rate).

   3. Start a workout and monitor the heart rate display.

   4. Perform exercises to reach different heart rate levels.

   5. Observe the visual and/or auditory alerts indicating whether the user is within or outside their target heart rate zone.

Postconditions:

   * User-defined target heart rate zones are saved.

Expected Result: The app accurately calculates and displays custom target heart rate zones, providing real-time feedback on whether the user is within or outside those zones during activity.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify historical heart rate data retrieval and visualization

Test Case ID: HRT-03

Test Case Title: Verify historical heart rate data retrieval and visualization

Test Case Description: This test case checks if historical heart rate data is stored correctly and if the app allows users to view and analyze this data effectively.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

* User has completed at least one workout with heart rate data recorded.

Test Data: Historical heart rate data from previous workouts.

Test Steps:

1. Access the heart rate history section within the app.
2. Filter and sort the data by date, time, workout type, or duration.
3. Examine the graphical visualization of the heart rate data (e.g., line graph, chart).
4. Observe individual data points and overall trends in heart rate patterns.

Postconditions:

* Historical data is displayed accurately.

Expected Result: The app retrieves and visualizes historical heart rate data clearly, allowing users to analyze their heart rate trends over time and track their progress.

Severity: Major

Type of Testing: Functional, Data Verification

Test Case Approach: Positive

Test Case 4: Verify abnormal heart rate alerts

Test Case ID: HRT-04

Test Case Title: Verify abnormal heart rate alerts

Test Case Description: This test case ensures that the app correctly identifies and alerts the user about potentially abnormal heart rate readings.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

* User is logged in and setting has activated for abnormal heart rate alerts.
* Wearable device or heart rate sensor is connected.
*

Test Data: Pre-defined scenarios with heart rate data exceeding the desired alert thresholds.

Test Steps:

1. Manually simulate high and low heart rate readings (above and below pre-set thresholds) during a workout.
2. Observe if the app triggers appropriate alerts (visual, auditory, or haptic) for the simulated abnormal readings.
3. Observe whether the app allows for customization of alert thresholds.

Postconditions:

Expected Result: The app accurately detects abnormal heart rate values and delivers timely alerts to the user, allowing them to take appropriate measures.

Severity: Major

Type of Testing: Functional, Security

Test Case Approach: Positive, Negative

Test Case 5: Verify data compatibility with wearable devices

Test Case ID: HRT-05

Test Case Title: Verify heart rate data compatibility with various devices

Test Case Description: This test case checks the app's ability to seamlessly integrate and retrieve heart rate data from different types of compatible wearable devices. *

*

Test Suite:* Heart Rate Monitoring

Test Priority: Medium

Preconditions:

    * List of compatible wearable devices, e.g., Apple Watch, Fitbit, Polar, Garmin etc.

    * Bluetooth is enabled on the device and the app.

Test Data: Heart rate data from diverse wearable devices.

Test Steps:

    1. Pair the app with each compatible wearable device.

    2. Commence a workout with heart rate recording enabled on both the wearable and the app.

    3. Cross-reference the heart rate data displayed by the app with the data shown on the wearable device.

Postconditions:

    * Connection with wearable devices remains stable.

Expected Result The app accurately retrieves and displays heart rate data from diverse wearable devices, ensuring seamless integration and user convenience.

Severity: Moderate

Type of Testing: Integration, Compatibility

Test Case Approach: Positive

Data - 79

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Calorie Tracking

Feature description -

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

Userstory title -

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

Acceptance criteria -

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

Generated Test Cases:

Here are 5 detailed test cases for the Fitness Tracker app's Calorie Tracking feature, each designed to cover a specific acceptance criterion:

Test Case 1

* **Test Case ID:** CT-001
* **Test Case Title:** Verify that users can log daily food intake and calorie consumption.
* **Test Case Description:** This test case verifies the functionality of logging food items and their corresponding calorie information into the app.
* **Test Suite:** Calorie Tracking
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the app.
    * The app is connected to a working internet connection.
* **Test Data:** List of common food items with varying calorie counts (e.g., apple, banana, chicken breast, pizza slice).
* **Test Steps:**
    1. Navigate to the "Food Log" section of the app.
    2. Select the "Add Food" option.
    3. Search for a food item (e.g., "apple").
    4. Select the desired food item from the search results.

    5. Input the quantity of the food item consumed (e.g., 1 apple).
    6. Tap on "Log Food Entry."
* **Postconditions:**
    * The logged food item appears in the food log with its corresponding calorie count.
    * The total calories consumed for the day are updated accordingly.
* **Expected Result:** The system successfully logs the food item and its calorie information, updating the total calories consumed for the day.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** CT-002
* **Test Case Title:** Verify the app calculates and displays total calories consumed and burned.
* **Test Case Description:** This test case verifies the accuracy of the app's calorie calculation and display for both consumed and burned calories.
* **Test Suite:** Calorie Tracking
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the app.
    * The app has access to user's activity data (if applicable).
* **Test Data:** Pre-logged food items and estimated calorie expenditure from physical activity.
* **Test Steps:**
    1. Navigate to the "Calorie Summary" section of the app.
    2. Review the displayed "Total Calories Consumed" value against logged food entries.
    3. Review the displayed "Total Calories Burned" value (if data is available) against activity data.

* **Postconditions:**

    * The calorie summary screen is displayed.

* **Expected Result:** The app accurately calculates and displays the total calories consumed and burned based on the logged data. If the app integrates with wearable devices, the calorie burn should be consistent with data from those devices.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** CT-003

* **Test Case Title:** Verify that users can set daily calorie goals and track progress.

* **Test Case Description:** This test case verifies the functionality of setting a calorie goal and tracking progress against that goal.

* **Test Suite:** Calorie Tracking

* **Test Priority:** High

* **Preconditions:**

    * User is logged into the app.

* **Test Data:**

* **Test Steps:**

    1. Navigate to the "Settings" section of the app.

    2. Select the "Calorie Goals" option.

    3. Set a desired daily calorie goal (e.g., 2000 calories).

    4. Save the calorie goal.

    5. Log food items and physical activity as per a test plan.

    6. Monitor the progress bar or tracking chart against the set calorie goal.

* **Postconditions:**

    * The calorie goal is saved in the user's profile.

* **Expected Result:** The app allows the user to set a calorie goal and accurately tracks their progress towards achieving it throughout the day, providing visual feedback.

* **Severity:** Critical

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** CT-004

* **Test Case Title:** Verify food items and calorie information are searchable in the database.

* **Test Case Description:** This test case verifies the functionality of the food search engine within the calorie tracking feature.

* **Test Suite:** Calorie Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the app.

    * The app has access to a comprehensive food database.

* **Test Data:**

* **Test Steps:**

    1. Navigate to the "Food Log" section of the app.

    2. Input different search terms (e.g., "chicken," "salad," "banana").

    3. Verify the relevance of the returned search results.

    4. Select a food item from the search results and verify its calorie information.

* **Postconditions:**

    * The updated list of foods is displayed.

* **Expected Result:** The app's search engine accurately identifies and displays relevant food items based on user input, providing the associated calorie information.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** CT-005
* **Test Case Title:** Verify users receive alerts for exceeding calorie goals.
* **Test Case Description:** This test case ensures the system notifies users when they consume calories exceeding their set daily goal.
* **Test Suite:** Calorie Tracking
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the app.
    * A calorie goal is set.
* **Test Data:**
* **Test Steps:**
    1. Log food items that exceed the set calorie goal.
    2. Observe if the app displays an alert notification prompting users to be mindful of their calorie intake.
* **Postconditions:**
    * An alert notification is displayed.
* **Expected Result:** The app generates an alert notification when the user logs calories that exceed their defined daily goal, helping them stay within their planned calorie range.
* **Severity:** Minor
* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Data - 80

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

Here are five test cases for the Sleep Tracking feature of the Fitness Tracker App, following the provided template and addressing all acceptance criteria:

Test Case 1

Test Case ID: FTR-S1

Test Case Title: Verify that sleep data is recorded including duration and quality

Test Case Description: This test case aims to verify the app's ability to accurately record sleep duration and quality based on sensor data.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has enabled sleep tracking in the app settings.
- User is wearing a compatible sleep-tracking device (e.g., smartwatch) or utilizes the phone's built-in sensors.
- User sleeps during the designated testing period.

Test Data: No test data needed

Test Steps:

1. Initiate sleep tracking before bed.
2. Notify the app upon waking up.
3. Review the sleep summary displayed in the app.
4. Compare the recorded sleep duration with the actual time spent asleep.
5. Observe the sleep quality score and its correlation with individual sleep stages.

Postconditions:

- Sleep data is logged.

Expected Result:

The app accurately records the total sleep duration and provides a quality score based on analyzing sensor data, indicating the user's sleep quality during the night. Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2

Test Case ID: FTR-S2

Test Case Title: Verify that users can view detailed sleep stages and patterns

Test Case Description: This test case will validate the app's ability to display detailed sleep stage information and identify recurring patterns.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has sufficient sleep data recorded (at least 3 nights).
- User has enabled detailed sleep stage tracking in the app settings.

Test Data: No test data needed

Test Steps:

1. Navigate to the sleep tracking section of the app.
2. Select a specific sleep session from the recorded data.
3. Examine the breakdown of sleep stages (e.g., Light, Deep, REM).

4. Observe visual representations of sleep stages throughout the night.

5. Check for patterns in sleep stage distribution over several nights.

Postconditions:

- Sleep stages and trends are displayed

Expected Result:

The app presents detailed information on sleep stages, including their duration and timing during the night, allowing users to analyze their sleep patterns.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3

Test Case ID: FTR-S3

Test Case Title: Verify that the app provides insights into sleep disturbances

Test Case Description: This test case will ensure the app correctly identifies disturbances during sleep and offers relevant insights.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has recorded sleep data including disturbances (e.g., wakeups, restless periods).

Test Data: No test data needed

Test Steps:

1. Review the sleep summary for a sleep session containing disturbances.
2. Identify the reported instances of sleep interruptions and their durations.
3. Analyze any insights provided by the app regarding the potential causes of disturbances.
4. Check if the app offers suggestions for mitigating sleep disruption.

Postconditions:

- Insights into sleep disturbances are presented

Expected Result:

The app accurately identifies sleep disturbances and provides relevant information about their frequency, duration, and potential contributing factors.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4

Test Case ID: FTR-S4

Test Case Title: Verify that users receive recommendations for improving sleep quality

Test Case Description: This test case will evaluate the app's ability to offer personalized suggestions for enhancing user sleep.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has gathered sufficient sleep data for the app to analyze trends and patterns.

Test Data: No test data needed

Test Steps:

1. Access the sleep tracking section and review the personal insights provided.
2. Examine the recommendations offered by the app for improving sleep quality.
3. Assess the relevance and actionable nature of the suggested strategies.

Postconditions:

- Sleep quality improvement recommendations are available.

Expected Result:

The app delivers personalized recommendations based on the user's sleep data, offering specific and actionable advice to enhance sleep quality.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5

Test Case ID: FTR-S5

Test Case Title: Verify that sleep data can be viewed in daily, weekly, and monthly summaries

Test Case Description: This test case aims to confirm the app's functionality for presenting sleep data in different timeframes.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- User has recorded sleep data across several days.

Test Data: No test data needed

Test Steps:

1. Navigate to the sleep tracking main page.
2. Select the "Daily" view to see sleep data from the previous day.
3. Switch to "Weekly" view to analyze sleep patterns over the past week.
4. Change to "Monthly" view to examine sleep trends for the current month.

Postconditions:

- Sleep data is displayed in selected timeframes.

Expected Result:

The app effectively displays sleep data in daily, weekly, and monthly summaries, allowing users to track trends and progress over different time periods.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 81

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Logging

Feature description -

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

Userstory title -

As a user, I want to log my workouts so that I can track my progress and improve my performance.

Acceptance criteria -

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

Generated Test Cases:

Test Cases for Workout Logging Feature

Test Case ID: WLOG-001

Test Case Title: Verify that users can log various workout types with details.

Test Case Description: This test case verifies that the application allows users to log workouts with different types, durations, intensities, and notes.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User is logged in to the fitness tracker app.

- User is on the workout logging screen.

Test Data: Different workout types (e.g., Running, Swimming, Cycling, Weightlifting, Yoga)

Test Steps:

1. Select a workout type from the list of available options.

2. Input the duration of the workout (e.g., 30 minutes).

3. Select the intensity level of the workout (e.g., Moderate).

4. Add notes or comments about the workout (e.g., "Focused on core strength").

5. Save the workout log entry.

Postconditions:

- The workout log entry is saved successfully.

- The data for workout type, duration, intensity, and notes is displayed correctly.

Expected Result: The system successfully logs the workout with the specified details, confirming the user can log various workout types.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: WLOG-002

Test Case Title: Verify that workout logs include duration, intensity, and notes.

Test Case Description: This test case verifies that mandatory fields like duration, intensity, and notes are included in every logged workout.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User is logged in to the fitness tracker app.
- User is on the workout logging screen.
- The user has logged a workout previously.

Test Data: View a pre-logged workout entry.

Test Steps:

1. Select a pre-logged workout entry from the list.
2. Verify that the duration, intensity, and notes are displayed for the workout.

Postconditions:

- The user views the logged workout details.

Expected Result: The system displays the duration, intensity, and notes associated with the logged workout, confirming data integrity.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: WLOG-003

Test Case Title: Verify that users can edit and delete workout entries.

Test Case Description: This test case verifies the functionality of editing and deleting existing workout entries.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User is logged in to the fitness tracker app.
- User has logged at least one workout.

Test Data: A previously added workout entry.

Test Steps:

1. Navigate to the workout log.
2. Select the entry to be edited.
3. Modify the duration, intensity, or notes of the workout.
4. Save the changes.

Postconditions:

- The workout entry is updated with the new details.

Expected Result: The system successfully enables editing of workout entries and correctly updates the data.

5. Select the entry to be deleted.
6. Confirm the deletion.

Postconditions:

- The workout entry is removed from the log.

Expected Result: The system successfully deletes the selected workout entry from the log.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: WLOG-004

Test Case Title: Verify that data is stored and viewable for tracking progress.

Test Case Description: This test case verifies that workout data is correctly stored and can be viewed for tracking progress over time.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged in to the fitness tracker app.
- User has logged multiple workouts on different dates.

Test Data: View all logged workouts.

Test Steps:

1. Access the workout log.
2. Observe the display of workouts listed chronologically.
3. Select different date ranges to view workout logs for specific periods.

Postconditions:

- The system displays the workout logs chronologically.
- The user can filter and view workout logs for selected date ranges.

Expected Result: The system accurately stores and displays workout data, allowing users to track their progress over time.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: WLOG-005

Test Case Title: Verify that users can filter logs by date and workout type.

Test Case Description: This test case verifies the functionality of filtering workout logs by date and type.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged in to the fitness tracker app.
- User has logged multiple workouts of different types on different dates.

Test Data: Filtered workout logs.

Test Steps:

1. Navigate to the workout log.

2. Use the filters to select a specific date range.

3. Verify that the log displays workouts only within the selected date range.

4. Apply a filter to select a specific workout type.

5. Verify that the log displays only workouts of the chosen type.

6. Apply both date range and workout type filters.

7. Verify that the log displays only workouts matching both criteria.

Postconditions:

- The workout log displays filtered results based on date and/or workout type.

Expected Result: The system successfully filters workout logs based on date and workout type, allowing users to retrieve specific workout information efficiently.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 82

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Goal Setting

Feature description -

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

Userstory title -

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

Acceptance criteria -

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

Generated Test Cases:

Fitness Tracker App: Goal Setting Test Cases

Test Case 1: Verify that users can set goals for various fitness metrics

Test Case Description: This test case verifies the user's ability to set goals for different fitness metrics offered by the app, including steps, calories, distance, and weight.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

* User is logged in to the Fitness Tracker App.

* The user has accessed the Goal Setting section.

Test Data: Realistic values for each fitness metric. (e.g., Steps: 10,000, Calories: 2000, Distance: 5 miles, Weight: 15 pounds)

Test Steps:

1. Navigate to the Goal Setting section within the app.

2. Select the ?Steps? metric and set a goal of 10,000 steps per day.

3. Select the ?Calories? metric and set a goal of 2000 calories burned per day.

4. Select the ?Distance? metric and set a goal of 5 miles run per week.

5. Select the ?Weight? metric and set a goal to lose 15 pounds.

Postconditions:

* The goals are saved in the user's profile.

Expected Result:

* The user successfully sets goals for all four fitness metrics.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2: Verify progress tracking for each goal

Test Case Description: This test case verifies that the app accurately tracks the user's progress towards their set goals and displays it within the app.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

* User has set valid goals (as per Test Case 1).

* Sufficient time has passed to allow some progress towards the goals.

Test Data: Simulated or actual user activity leading to some progress towards the set goals. (e.g., steps taken, calories burned, distances covered, weight changes)

Test Steps:

1. Go to the Goal Setting section within the app.

2. Select the ?Steps? goal.

3. Verify that the app displays the number of steps taken towards the target goal.

4. Repeat steps 2 and 3 for each of the other set goals (Calories, Distance, Weight).

Postconditions:

* Progress data is displayed for each goal.

Expected Result:

* The app accurately displays the user's progress toward each goal based on their activity.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3: Verify motivational feedback and reminders

Test Case Description: This test case verifies that the app provides motivating feedback and reminders to users based on their goal progress.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

* User has set the target goals (as per Test Case 1).

* User has not reached a significant milestone yet.

Test Data: User activity that results in moderate progress towards some goals but not completing them.

Test Steps:

1. View the Goal Setting section within the app.

2. Observe any motivational feedback provided based on the user's progress.

3. Check for any reminder notifications encouraging the user to continue working towards their goals.

Postconditions:

* Feedback and reminders are displayed or sent.

Expected Result:

* The app delivers motivating feedback and reminders to keep the user engaged and progressing towards their

goals.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4: Verify the ability to adjust or update goals

Test Case Description: This test case verifies that users can modify their existing fitness goals within the app.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

* User has set valid goals (as per Test Case 1).

Test Data: Modified values for existing fitness goals. (e.g., increase steps goal to 12,000, decrease calorie goal to 1800)

Test Steps:

1. Access the Goal Setting section within the app.

2. Select one of the existing goals.

3. Modify the goal value to a new target.

4. Save the changes to the goal.

Postconditions:

* The updated goal value is reflected in the user's profile.

Expected Result:

* Users can successfully adjust and update their existing fitness goals within the app.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5: Verify the summary of goal attainment

Test Case Description: This test case verifies that the app provides a clear summary of the user's goal attainment, indicating progress towards each goal.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

* User has set valid goals and made progress towards those goals (as per previous test cases).

Test Data: User activity leading to varying degrees of progress towards different goals.

Test Steps:

1. Open the ?Goal Summary? section within the app.

2. Review the summary for each goal, paying attention to the progress indicators and percentage completion.

Postconditions:

* Summary data for all goals is displayed.

Expected Result:

* The app displays a comprehensive summary of the user's goal attainment, clearly illustrating their progress towards each goal.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 83

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Plans

Feature description -

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

Userstory title -

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

Acceptance criteria -

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

Generated Test Cases:

Here are 5 detailed test cases for the Fitness Tracker App's "Workout Plans" feature based on your provided input:

Test Case 1:

Test Case ID: WT-001

Test Case Title: Verify that users can select workout plans based on fitness levels.

Test Case Description: This test case verifies that users can navigate through the workout plan selection process and choose plans based on their specified fitness level (beginner, intermediate, advanced).

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in to the Fitness Tracker App.
- User has successfully set their fitness level in the profile settings.

Test Data: Different workout plans categorized for various fitness levels (beginner, intermediate, advanced).

Test Steps:

1. Navigate to the "Workout Plans" section of the app.
2. Select the filter option for fitness level.
3. Choose a specific fitness level (e.g., Beginner).
4. Verify that only workout plans appropriate for the selected fitness level are displayed.
5. Select a plan from the displayed results.

Postconditions:

- The selected plan is loaded with its details.
- The user is directed to the plan's schedule and exercises.

Expected Result: The app successfully filters workout plans based on the selected fitness level, providing relevant options for the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: WT-002

Test Case Title: Verify that workout plans include detailed exercise routines and schedules.

Test Case Description: This test case verifies that a selected workout plan provides comprehensive information about exercises, their order, duration, and frequency.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in to the Fitness Tracker App.

- A workout plan is selected (preferably one with multiple exercises).

Test Data: N/A

Test Steps:

1. Navigate to the selected workout plan.

2. Review the plan's schedule, checking the number of days per week and the structure of the workouts.

3. Examine individual exercises within a workout, noting their names, descriptive details, sets, repetitions, and rest periods.

4. Verify that the exercises are logically sequenced and progress gradually.

Postconditions:

- User has gained information about the workout plan's structure and exercises.

Expected Result: The provided workout plan includes detailed information on exercises, their order (schedule), frequency, and duration, enabling the user to understand and follow it effectively.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: WT-003

Test Case Title: Verify users can track adherence and progress within a workout plan.

Test Case Description: This test case confirms that the app allows users to mark exercise completion, track their performance over time, and monitor progress towards their fitness goals set within the plan.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in to the Fitness Tracker App.
- A workout plan is selected and started.
- The user has completed at least one workout from the plan.

Test Data: Data related to completed workouts within the selected plan (e.g., timestamps, exercise results, calories burned).

Test Steps:

1. Navigate to the selected workout plan.
2. Locate the workout log or progress section.
3. Verify that completed workouts are listed with accurate timestamps and details.
4. Check if the app provides metrics tracking progress (e.g., total workouts completed, exercises mastered, weight lifted, calories burned).
5. Examine any visual aids or graphs displaying progress trends.

Postconditions:

- User can view accurate records of their completed workouts.
- User can assess their progress within the plan.

Expected Result: The app accurately tracks the user's adherence to the workout plan and provides clear visualizations or data insights into their progress towards their fitness goals.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: WT-004

Test Case Title: Verify the app provides notifications and reminders for workouts.

Test Case Description: This test case verifies that the app sends timely notifications or reminders to users about upcoming workouts in their selected plan.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in to the Fitness Tracker App and has a workout plan active.
- App notification permissions are granted.

Test Data: N/A

Test Steps:

1. Select a workout plan and set a notification preference (e.g., 30 minutes before the workout).
2. Advance to the time specified for the next scheduled workout within the plan.
3. Observe whether a notification is received on the device.
4. Verify the notification content (workout name, time, duration) is accurate and relevant.

Postconditions:

- User receives a notification reminder.

Expected Result: The app successfully delivers notifications or reminders to the user about upcoming workouts in their selected plan, ensuring timely awareness and adherence.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: WT-005

Test Case Title: Verify users can adjust workout plans based on progress and preferences.

Test Case Description: This test case confirms the user's ability to modify their workout plan, e.g., by skipping exercises, adjusting schedules, or changing the plan altogether, based on their progress, fitness level, or changing needs.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is logged in to the Fitness Tracker App.
- A workout plan is active.

Test Data:

- Modified workout schedule (e.g., adding rest days, extending durations).
- Selected exercises to add or remove from the routine.
- Different workout plans to switch to (upgrading or downgrading fitness level).

Test Steps:

1. Navigate to the active workout plan.
2. Utilize the app's options to adjust the plan (e.g., skip exercises, modify schedule).
3. For plan switching, select a new plan with different intensity or goals.
4. Observe if the changes are successfully reflected in the plan's schedule and exercise list.

Postconditions: User's workout plan has been adjusted to their specifications.

Expected Result:

The app allows users to make desired changes to their workout plan, providing flexibility and customization based on individual preferences and progress.

Data - 84

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Community Support

Feature description -

Users can connect with a community of like-minded individuals, share progress updates, participate in challenges, and offer support and encouragement to fellow users. Community support fosters accountability, motivation, and a sense of camaraderie among users striving towards similar fitness goals.

Userstory title -

As a user, I want to connect with a community so that I can share my progress and stay motivated.

Acceptance criteria -

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.', 'The app facilitates participation in community challenges.', 'Users can offer and receive support and encouragement.', 'Community activity and engagement are tracked.']

Generated Test Cases:

Fitness Tracker App: Community Support Test Cases

Test Case 1:

* **Test Case ID:** FIT-COMM-001

* **Test Case Title:** Verify that users can join and participate in community groups.

* **Test Case Description:** This test case aims to verify the functionality of joining and participating in community groups within the Fitness Tracker App.

* **Test Suite:** Community Support

* **Test Priority:** High

* **Preconditions:**

    * User is logged in to the Fitness Tracker App

    * There are existing community groups available within the app

* **Test Data:** Names and descriptions of existing community groups

* **Test Steps:**

    1. Navigate to the "Community" section of the app.

    2. Browse through the list of available community groups.

    3. Select a desired group and click the "Join Group" button.

    4. Verify that the user is successfully added to the selected group.

    5. Within the group, participate in discussions by posting a comment or message.

* **Postconditions:**

    * User is a member of the selected community group.

    * User can view and participate in discussions within the group.

* **Expected Result:** The user successfully joins the chosen community group and can engage in discussions within that specific group.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** FIT-COMM-002

* **Test Case Title:** Verify that users can share progress updates and achievements.

* **Test Case Description:** Confirm the functionality of sharing progress updates and achievements within the Fitness Tracker App's community features.

* **Test Suite:** Community Support

* **Test Priority:** High

* **Preconditions:**

    * User is logged in to the Fitness Tracker App.

    * User is a member of at least one community group.

* **Test Data:** Example progress updates (e.g., completed workout, reaching a step goal, personal best)

* **Test Steps:**

    1. Navigate to the chosen community group within the app.

    2. Compose a progress update message describing a recent achievement or milestone.

    3. Add optional media content (e.g., photo, video) to the update.

    4. Publish the progress update within the selected community group.

* **Postconditions:**

    * The progress update is visible to other members within the chosen community group.

    * Other users can interact with the progress update (e.g., like, comment).

* **Expected Result:** The user successfully shares their progress update, including optional media, within the designated community group, making it visible and interactive for other members.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** FIT-COMM-003

* **Test Case Title:** Verify that the app facilitates participation in community challenges.

* **Test Case Description:** This test case aims to assess the functionality of participating in community challenges within the Fitness Tracker App.

* **Test Suite:** Community Support

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in to the Fitness Tracker App.

    * At least one community challenge is available within the app.

* **Test Data:** Details of an existing community challenge (e.g., step-count competition, workout duration challenge)

* **Test Steps:**

    1. Navigate to the "Community" section of the app.

    2. Locate and select the chosen community challenge.

    3. Review the challenge rules and details.

    4. Enroll in the challenge by clicking the "Join Challenge" button.

    5. Track the progress within the challenge and interact with other participants.

* **Postconditions:**

    * The user is enrolled in the chosen community challenge.

    * The user's progress is tracked and displayed within the challenge.

    * The user can view the progress and rankings of other participants in the challenge.

* **Expected Result:** The user successfully joins a community challenge, their progress is tracked, and they can interact with other participants.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** FIT-COMM-004

* **Test Case Title:** Verify that users can offer and receive support and encouragement.

* **Test Case Description:** This test case will assess the functionality of offering and receiving support and encouragement within community groups within the Fitness Tracker App.

* **Test Suite:** Community Support

* **Test Priority:** High

* **Preconditions:**
    * User is logged in to the Fitness Tracker App.
    * User is a member of at least one community group.
    * There are existing posts or messages within the community group.

* **Test Data:** Positive and encouraging messages to be offered

* **Test Steps:**
    1. Navigate to the chosen community group within the app.
    2. Identify an existing post where a user might need support.
    3. Compose a supportive and encouraging message responding to the post.
    4. Send the message within the group's discussion thread.
    5. Observe and verify if other users engage with or acknowledge the given encouragement.

* **Postconditions:**
    * The supportive message is visible to other members within the group.
    * Other members interact with the message (e.g., like, reply, acknowledge).

* **Expected Result:** The user successfully offers support and encouragement within the community group, which is acknowledged and interacted with by other users.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** FIT-COMM-005
* **Test Case Title:** Verify that community activity and engagement are tracked.
* **Test Case Description:** This test case aims to confirm the tracking of community activity and engagement within the Fitness Tracker App.
* **Test Suite:** Community Support
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged in to the Fitness Tracker App.
    * Users are actively participating in a chosen community group.
* **Test Data:** Examples of community activity (e.g., new posts, comments, challenge participation)
* **Test Steps:**
    1. Navigate to the chosen community group within the app.
    2. Perform several actions within the group, such as making a post, commenting on existing posts, and participating in a challenge.
3. Observe the real-time updates or activity feed within the community group.
4. Refer to the activity log or statistics section to view a historical record of community engagement.
* **Postconditions:**
    * All performed actions are reflected in the real-time activity feed.
    * A comprehensive record of community activity is available in the activity log or statistics section.
* **Expected Result:** The app accurately tracks and displays community activity and engagement in real-time and through historical records.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Data - 85

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Wearable Devices

Feature description -

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

Userstory title -

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

Acceptance criteria -

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

Generated Test Cases:

Fitness Tracker App Test Cases: Wearable Integration

Test Case ID: ft-wd1

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: This test case aims to verify that the Fitness Tracker app can successfully connect and sync data with various popular wearable devices, ensuring broad compatibility.

Test Suite: Wearable Integration

Test Priority: High

Preconditions:

* The user has a compatible wearable device (e.g., a Fitbit, Garmin, Apple Watch).
* Bluetooth is enabled on the user's device and the wearable.
* The Fitness Tracker app is installed and running.

Test Data: List of compatible wearable devices.

Test Steps:

1. Open the Fitness Tracker app.
2. Navigate to the wearable device settings.
3. Select the desired wearable device from the list.
4. Follow the on-screen prompts to initiate the pairing process.
5. Verify that the app successfully connects to the wearable device.

Postconditions:

* The user is notified that the wearable device is successfully connected.

* The app displays the wearable device name and model.

Expected Result: The app recognizes and connects to the chosen wearable device, indicating a successful integration.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ft-wd2

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: This test case confirms that the Fitness Tracker app accurately retrieves and syncs activity data, such as steps taken, calories burned, heart rate, etc., from a user's connected wearable.

Test Suite: Wearable Integration

Test Priority: High

Preconditions:

* The user has successfully connected a wearable device with the app.

* The wearable device has recorded some activity data.

Test Data: A sample workout data set from the wearable device.

Test Steps:

1. Ensure the wearable device is recording activity data.
2. Open the Fitness Tracker app.
3. Navigate to the data syncing section.
4. Initiate a manual sync with the wearable device.
5. Observe the app for the successful transfer of data.

Postconditions:

* The app updates the user's activity dashboard with data from the wearable.

Expected Result: The app successfully syncs data from the wearable device, reflecting the user's activity accurately.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: ft-wd3
Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: This test case evaluates the Fitness Tracker app's ability to receive and display real-time data updates from connected wearable devices, ensuring a responsive and up-to-date user experience.

Test Suite: Wearable Integration
Test Priority: High
Preconditions:

* The user has a wearable device capable of transmitting real-time data (e.g., heart rate monitor, GPS tracker).
* The user has successfully connected the wearable device to the app.
* The wearable device is actively recording data.

Test Data: A scenario requiring real-time data updates (e.g., running with a GPS tracker or performing a workout with heart rate monitoring).

Test Steps:

1. Start a workout or activity that generates real-time data on the wearable device.
2. Monitor the Fitness Tracker app concurrently.

3. Observe the app's display of real-time data from the wearable device.
4. Verify that the data updates in the app are synchronized and consistent with the information from the wearable.

Postconditions:

* The app continues to display real-time data updates while the user engages in the activity.

Expected Result: The Fitness Tracker app seamlessly receives and displays real-time data from the connected wearable device, providing a smooth and interactive user experience.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: ft-wd4

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: This test case ensures that the Fitness Tracker app correctly retrieves and displays essential fitness metrics (steps, heart rate, and calories burned) from connected wearable devices.

Test Suite: Wearable Integration

Test Priority: High

Preconditions:

* The user has successfully connected a wearable device capable of recording steps, heart rate, and calories burned.
* The user has engaged in activities that generate data for these metrics.

Test Data: A range of activities with varying intensities and durations to generate representative data for steps, heart rate, and calories burned.

Test Steps:

1. Initiate an activity and monitor the data being recorded on the wearable device.
2. Open the Fitness Tracker app and navigate to the data display section for the chosen activity.
3. Observe the values displayed for steps, heart rate, and calories burned.

Postconditions:

* The app synchronizes and displays data for the chosen activity.

Expected Result: The app accurately retrieves and displays data for steps, heart rate, and calories burned recorded by the connected wearable device.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: ft-wd5
Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: This test case examines the app's ability to guide users through troubleshooting steps when experiencing connectivity problems with their wearable devices.

Test Suite: Wearable Integration
Test Priority: Medium
Preconditions:

* The user has activated Bluetooth on their device and the wearable.
* The user has successfully connected the wearable to the app previously.

Test Data:

* Simulated scenarios causing connectivity issues (e.g., intermittent Bluetooth connection, device range exceeding limit).

Test Steps:

1. Initiate a simulated connectivity issue while the wearable device is connected to the app.
2. Observe the app's response to the lost connection.
3. Follow any on-screen prompts or instructions provided by the app to troubleshoot the issue.
4. Attempt to reconnect the wearable device.

Postconditions:

* The app provides clear and concise error messages indicating the connectivity issue.
* The user is guided through appropriate troubleshooting steps.

Expected Result: The app detects the connectivity issue, provides relevant error messages, offers suitable troubleshooting solutions, and assists the user in re-establishing the connection with the wearable device.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive/Negative

Data - 86

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Water Intake Tracking

Feature description -

Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

Userstory title -

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

Acceptance criteria -

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

Generated Test Cases:

Here are 5 test cases for the Fitness Tracker App's Water Intake Tracking feature, following the provided template:

Test Case 1

Test Case ID: FT-WI-001

Test Case Title: Verify that users can log their daily water intake

Test Case Description: This test case verifies that users can successfully enter their daily water intake manually and save the record.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged into the Fitness Tracker App
- Day's water intake log is empty

Test Data: Different water intake values (e.g., 500 ml, 1 liter, 2.5 liters)

Test Steps:

1. Navigate to the water intake tracking section of the app.
2. Enter a specific water intake value using the app's input field.
3. Tap on the "Log Intake" or "Save" button.
4. Verify that the entered value is displayed in the water intake log for the current day.

Postconditions:

- Water intake record is saved successfully.

Expected Result: The entered water intake value is correctly logged and displayed in the app, accurately reflecting the user's consumed amount for the day.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: FT-WI-002

Test Case Title: Verify that the app calculates total water consumption for the day

Test Case Description: This test case verifies that the app accurately calculates the cumulative total water intake for the current day based on multiple logged entries.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged into the Fitness Tracker App
- Water intake log contains multiple entries for the day

Test Data: Different water intake values logged throughout the day

Test Steps:

1. Log several water intake entries at various times during the day using different values.
2. Verify that the "Total Water Consumed" or "Daily Intake" display accurately reflects the sum of all logged entries.

Postconditions:

- Total water consumption is correctly calculated and displayed.

Expected Result: The app flawlessly sums up all logged water intake entries, providing an accurate representation of the user's total water consumption for the day.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: FT-WI-003

Test Case Title: Verify that users can set daily hydration goals

Test Case Description: This test case verifies the functionality of setting personalized daily hydration goals and storing them within the app.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged into the Fitness Tracker App
- No existing daily hydration goal is set

Test Data: Different hydration goal values (e.g., 2 liters, 3 liters)

Test Steps:

1. Navigate to the water intake tracking section and locate the "Set Goal" or "Hydration Target" option.
2. Enter a desired daily hydration goal value.

3. Tap on the "Save Goal" or "Confirm" button.
4. Verify that the entered goal is displayed as the user's current hydration target.

Postconditions:

- Daily hydration goal is set successfully.

Expected Result: The app allows the user to set their personalized daily hydration goal, which is then displayed prominently within the water intake tracking section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: FT-WI-004

Test Case Title: Verify that reminders are provided to encourage regular water intake

Test Case Description: This test case verifies that the app delivers reminders throughout the day to prompt the user to drink water and stay hydrated.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged into the Fitness Tracker App
- A daily hydration goal is set

Test Data: Different reminder times (e.g., morning, afternoon, evening)

Test Steps:

1. Set a daily hydration goal within the app.
2. Enable reminder notifications.
3. Allow time for the app to trigger reminders based on the set schedule.
4. Receive and verify a reminder notification at a pre-determined time.

Postconditions:

- Reminder notifications are delivered according to the user's settings.

Expected Result: The app sends timely reminders throughout the day, prompting the user to drink water and stay on track with their hydration goals.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: FT-WI-005

Test Case Title: Verify that users can view hydration progress and trends

Test Case Description: This test case verifies the ability to view both daily and historical hydration progress, and the visibility of potential trends in water intake.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged into the Fitness Tracker App
- Water intake logs are populated for multiple days

Test Data: Several days of water intake data

Test Steps:

1. Navigate to the hydration tracking section.
2. View the daily hydration progress for the current day.
3. Review the historical water intake data for previous days.
4. Analyze the displayed data to identify potential hydration trends.

Postconditions:

- User can visualize daily and historical hydration progress.

Expected Result: The app provides a comprehensive view of the user's hydration history, enabling them to track their progress, spot potential patterns in water intake, and make informed decisions about their hydration habits.

Severity: Major

Type of Testing: Functional & Data Analysis

Test Case Approach: Positive

Data - 87

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Nutrition Tracking

Feature description -

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

Userstory title -

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

Acceptance criteria -

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

Generated Test Cases:

Fitness Tracker App - Nutrition Tracking Test Cases

Here are 5 test cases for the "Nutrition Tracking" feature of the Fitness Tracker App:

Test Case 1:

* **Test Case ID:** FT-NUT-001
* **Test Case Title:** Verify that users can log their daily meals and snacks
* **Test Case Description:** This test case verifies the ability of users to input information about meals and snacks into the food diary, including food name, quantity, and serving size.
* **Test Suite:** Nutrition Tracking
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the app.
* **Test Data:** List of common food items and their corresponding nutritional information (available via an online food database or API)
* **Test Steps:**
    1. Navigate to the Nutrition Tracking section of the app.
    2. Select the "Add Meal" or "Log Food" option.
    3. Search for a specific food item (e.g., "banana").
    4. Select the desired food item from the search results.
    5. Enter the quantity consumed and serving size.
    6. Click "Save" or "Add to Diary."
* **Postconditions:** The logged meal with its details (food name, quantity, serving size) is displayed in the food diary.
* **Expected Result:** The entered meal information is accurately recorded in the food diary.
* **Severity:** Critical
* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** FT-NUT-002
* **Test Case Title:** Verify that the app calculates macronutrient intake
* **Test Case Description:** This test case ensures the app correctly calculates the macronutrients (carbs, fats, proteins) based on the user-logged food items.
* **Test Suite:** Nutrition Tracking
* **Test Priority:** High
* **Preconditions:**
    * User has logged at least three meals with various macronutrient compositions.
* **Test Data:** Predetermined meals with specific macronutrient values.
* **Test Steps:**
    1. Navigate to the Nutrition Tracking section and select the "Daily Summary" or "Macronutrient Breakdown."
    2. Review the calculated values for carbs, fats, and proteins.
    3. Compare the calculated macronutrients with the expected values based on the logged food items.
* **Postconditions:** The user can view a summary of their macronutrient intake for the day.
* **Expected Result:** The calculated macronutrient values accurately reflect the totals from the logged food items.
* **Severity:** Critical
* **Type of Testing:** Functional, Verification
* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** FT-NUT-003

* **Test Case Title:** Verify that users can set nutritional goals

* **Test Case Description:** This test case verifies the functionality for users to set personalized goals for macronutrients (carbs, fats, proteins) and/or calories.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** High

* **Preconditions:**

    * User is logged into the app.

* **Test Data:** Different target values for macronutrient goals (e.g., 200g carbs, 50g fats, 100g protein).

* **Test Steps:**

    1. Navigate to the Nutrition Tracking section and locate the "Goals" or "Settings" section.

    2. Select the option to set nutritional goals.

    3. Choose the desired goal type (e.g., calorie, macronutrients).

    4. Enter the specific target values for each goal.

    5. Save the set goals.

* **Postconditions:** The user-defined nutritional goals are saved and displayed within the app.

* **Expected Result:** The app correctly stores the user's nutritional goals and presents them within the interface.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** FT-NUT-004

* **Test Case Title:** Verify that users can view their nutritional history and trends

* **Test Case Description:** This test case checks the ability to retrieve and analyze past nutritional data,

including meal logs, macronutrient intake, and potentially visualize trends over time.

* **Test Suite:** Nutrition Tracking
* **Test Priority:** Medium
* **Preconditions:**
    * User has logged meals for at least a week.
* **Test Data:** Data representing multiple days of user meals.
* **Test Steps:**
    1. Navigate to the Nutrition Tracking section and locate the "History" or "Chart" feature.
    2. Select a time frame for viewing nutritional history (e.g., weekly, monthly).
    3. Analyze the displayed historical data, including logged meals, macronutrient summaries, and any calculated trends or visualizations.
* **Postconditions:** The user can access and review their past nutritional records.
* **Expected Result:** The app accurately displays a comprehensive history of the user's nutrition data, allowing for trend analysis and insights.
* **Severity:** Major
* **Type of Testing:** Functional, Verification
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** FT-NUT-005
* **Test Case Title:** Verify the system handles invalid nutrient input
* **Test Case Description:** This test case ensures the app handles invalid data input for nutrients consistently and gracefully.
* **Test Suite:** Nutrition Tracking
* **Test Priority:** Medium
* **Preconditions:**

    * User is logged into the app.

* **Test Data:**

    * Non-numeric values (e.g., "text" instead of "quantity").

    * Negative values for nutrient intake.

    * Extremely large values for nutrient intake.

* **Test Steps:**

    1. Attempt to log a meal with invalid data for nutrient category fields (e.g., "quantity" or "serving size").

    2. Observe the app's response to the invalid input.

    3. Attempt to save the meal with the invalid data.

* **Postconditions:** User feedback is provided regarding the invalid input.

* **Expected Result:** The system should prevent the saving of invalid data and provide appropriate error messages or validation prompts to guide the user towards correcting the input.

* **Severity:** Minor

* **Type of Testing:** Functional, Input Validation

* **Test Case Approach:** Negative

Data - 88

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Progress Reports

Feature description -

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

Userstory title -

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

Acceptance criteria -

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

Generated Test Cases:

Here are 5 test cases for the Fitness Tracker App, based on the provided user story and acceptance criteria:

Test Case 1

Test Case ID: ft_pr_01

Test Case Title: Verify that users receive weekly progress reports

Test Case Description: This test case aims to verify that users receive a weekly progress report as expected according to the predetermined criteria.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

* A user account is created and logged in.

* The user has completed at least five activity entries within the week.

Test Data: User activity data across a whole week.

Test Steps:

1. Navigate to the "Reports" or "Progress" section of the app.
2. Check for the availability of a weekly progress report for the current week.
3. Open the weekly report and verify that it includes:
    * A summary of total activity levels (steps, calories burned, etc.).
    * A record of completed workouts during the week.
    * Progress towards any set goals during the week.

Postconditions:

* User has viewed the weekly progress report.
* Fitness data is reflected accurately within the report.

Expected Result: The user receives a comprehensive weekly progress report displaying accurate activity data, workout history, and goal progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: ft_pr_02

Test Case Title: Verify that progress reports include monthly goal attainment

Test Case Description: This test case verifies the system's capability to showcase monthly goal attainment within the progress reports.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* A user account is created and logged in.

* The user has set a specific fitness goal for the current month.

* The user has logged activity data for at least half the month.

Test Data: User goal settings + activity data from the current month.

Test Steps:

1. Navigate to the "Reports" or "Progress" section of the app.

2. Locate and open the most recent monthly progress report.

3. Verify that the report includes a dedicated section showing the user's progress towards their set monthly

goal.

Postconditions: The user has examined the monthly progress report, focusing on goal attainment.

Expected Result: The monthly progress report clearly indicates the user's progress towards their specific goal, possibly with a percentage completion or visual representation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: ft_pr_03

Test Case Title: Verify the display of workout trends in progress reports

Test Case Description: This test case focuses on confirming that the app accurately presents workout trends within progress reports.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* A user account is created and logged in.
* The user has logged multiple workouts over a period of at least two weeks.

Test Data: Log of recent workouts with varying intensity levels.

Test Steps:

1. Access the "Reports" or "Progress" section of the app.
2. Open the latest monthly progress report.
3. Locate the section dedicated to displaying workout trends.
4. Analyze the presented data for patterns: e.g., increase/decrease in workout frequency, change in average workout duration, shift in workout type.

Postconditions: User has analyzed the trend information in the report.

Expected Result: Reports provide a clear visualization or summary of workout trends, such as increased frequency on certain days, changes in average duration, or shifts in preferred workout types.

Severity: Medium

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: ft_pr_04

Test Case Title: Verify that users can customize report frequency

Test Case Description: This test case confirms that the user has the ability to adjust the frequency of their progress reports according to their preferences.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* A user account is created and logged in.
* Setting page is accessible within the app.

Test Data: No specific test data required.

Test Steps:

1. Inside the app, access the "Settings" or "Preferences" section.
2. Locate the "Progress Reports" or similar setting.

3. Modify the report frequency options (e.g., change from weekly to bi-weekly, or adjust the day of the week for delivery).

4. Verify that the changes to report frequency are successfully saved.

Postconditions:

* Report frequency setting has been modified within the app.

Expected Result: The app allows users to change their preferred report frequency (weekly, monthly, custom intervals), and the system acknowledges and applies these changes.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: ft_pr_05

Test Case Title: Verify that users can customize report content

Test Case Description: This test allows verifying if users can personalize their progress reports by selecting the type of data they want included.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* A user account is created and logged in.
* Setting page is accessible within the app.

Test Data: No specific test data required.

Test Steps:

1. Access the "Settings" or "Preferences" section within the app.
2. Locate the "Progress Reports" or similar setting.
3. Explore available options for customizing report content. For example, check boxes for:
* Steps taken

* Calories burned

* Workout summary

* Goal progress

* Sleep data (if applicable)

4. Select or deselect different content options based on preference.

5. Verify that the selected choices are saved within the report settings.

Postconditions: Report customization settings have been adjusted within the app.

Expected Result: The app provides options to personalize the content of progress reports. Users can choose which data points they want to see, ensuring reports are relevant to their specific fitness goals and interests.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 89

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

In-App Challenges

Feature description -

The app offers in-app challenges and competitions, encouraging users to set personal records, compete with friends, and earn rewards for achieving fitness milestones. In-app challenges add an element of fun, motivation, and friendly competition to the fitness tracking experience.

Userstory title -

As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals.

Acceptance criteria -

['Users can join various in-app challenges.', 'Challenges have clear objectives and rules.', 'Users can track their progress within the challenge.', 'Rewards are provided for achieving challenge milestones.', 'Users can compete with friends and view leaderboards.']

Generated Test Cases:

Here are 5 test cases for the In-App Challenges feature of the Fitness Tracker App, each addressing a specific acceptance criterion:

Test Case 1

Test Case ID: FcH7g3

Test Case Title: Verify that users can join various in-app challenges

Test Case Description: This test case will verify that the user interface allows users to browse and join different types of in-app challenges.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app
- The app has multiple active challenges available.

Test Data: Different challenge types (e.g., step count, distance, calorie burn, specific workouts)

Test Steps:

1. Navigate to the "Challenges" section of the app.
2. Browse through the list of available challenges.
3. Select a challenge from a different category than the user previously joined.
4. Read the challenge details, including objectives and rules.
5. Tap the "Join Challenge" button.

Postconditions:

- The user is successfully registered for the selected challenge.
- The challenge appears in the user's "My Challenges" section.

Expected Result: The user can join various in-app challenges with clear descriptions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: yW2qF8

Test Case Title: Verify that challenges have clear objectives and rules

Test Case Description: This test case verifies that challenge descriptions provide clear and concise information about the challenge goals, duration, and any specific rules.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app
- The app has a specific challenge with detailed rules available for joining.

Test Data: Details of an in-app challenge

Test Steps:

1. Navigate to the "Challenges" section of the app.
2. Select a challenge with clear rule sets.
3. Read the challenge description thoroughly.
4. Pay attention to the challenge objectives, duration, and any specific rules or requirements.

Postconditions:

- User understands the challenge objectives and rules.

Expected Result: Challenge descriptions are comprehensive and easy to understand.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: Grq5I9

Test Case Title: Verify users can track their progress within a challenge

Test Case Description: This test case verifies that users can view their progress within an ongoing challenge and track their performance towards the goal.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app
- The user has joined a challenge and has completed some tasks.

Test Data: Challenge progress data

Test Steps:

1. Navigate to the ?Challenges? section of the app.
2. Select "My Challenges" and choose the challenge the user is participating in.
3. Check the challenge progress tracker. Verify that progress is accurately reflected based on the tasks completed.

Postconditions:

- The user can see their progress in the challenge.

Expected Result: Progress tracking is accurate and provides clear visualizations of the user's performance.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: D7p2X4

Test Case Title: Verify that rewards are provided for achieving challenge milestones

Test Case Description: This test case verifies that the system awards rewards to users upon successfully completing challenge milestones.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app.
- The user has joined a challenge with predefined milestones and associated rewards.

Test Data: Challenge milestone rewards

Test Steps:

1. Navigate to the "My Challenges" section and join the challenge with defined rewards.
2. Perform the actions required to reach a milestone in the challenge.
3. Observe if the rewards are awarded immediately or upon successful verification by the system.

Postconditions:

- The user receives the expected reward for completing the milestone.

Expected Result: Rewards are accurately and timely awarded for achieving challenge milestones.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: f9a8o1

Test Case Title: Verify users can compete with friends and view leaderboards

Test Case Description: This test case ensures users can compare their progress with friends within a challenge and view a leaderboard.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app.
- The user has joined a challenge and friends have also joined the same challenge.

Test Data: Friendly users who joined the challenge

Test Steps:

1. Navigate to the "Challenges" section within the app.
2. Select the joined challenge and look for a leaderboard or friends' progress display section.

3. Verify the user can see their friends on the leaderboard and compare their performance against them.

Postconditions:

- Users can view a leaderboard or ranking system to compare friends' progress.

Expected Result: The feature allows users to view a leaderboard and compare their progress with their friends in real time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 90

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Customizable Dashboards

Feature description -

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

Userstory title -

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Acceptance criteria -

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

Generated Test Cases:

Here are 5 test cases generated based on your input data, each designed to cover one or more of the acceptance criteria for the Customizable Dashboards feature:

Test Case 1

* **Test Case ID:** FT001
* **Test Case Title:** Verify that users can select metrics to display on their dashboard.
* **Test Case Description:** This test case aims to confirm that users have the functionality to choose which fitness metrics they want to see on their personalized dashboard.
* **Test Suite:** Customizable Dashboards
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the Fitness Tracker App.
    * A default dashboard is loaded.
* **Test Data:** List of available fitness metrics (steps, calories burned, distance, heart rate, etc.)
* **Test Steps:**
    1. Navigate to the dashboard customization settings.

    2. Select "Metrics to Display" from the customization options.
    3. deselect at least 3 metrics that are currently pre-selected.
    4. Save the customized dashboard settings.
* **Postconditions:**
    * The dashboard is refreshed.
* **Expected Result:** The dashboard now displays only the selected metrics, excluding the deselected ones.

Test Case 2

* **Test Case ID:** FT002
* **Test Case Title:** Verify that the app allows customization of dashboard layout and theme.
* **Test Case Description:** This test case will ensure that users can rearrange the position of widgets on their dashboard and change the overall theme (e.g., dark mode, light mode).
* **Test Suite:** Customizable Dashboards
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the Fitness Tracker App.
    * A customized dashboard exists with multiple widgets.
* **Test Data:** Available widget layout options (vertical, horizontal, grid), theme options (dark, light)
* **Test Steps:**
    1. Enter dashboard customization settings.
    2. Select "Layout" and choose a different layout option from the available choices.
    3. Select "Theme" and choose a different theme.
    4. Save the dashboard customization settings.
* **Postconditions:**
    * The dashboard is refreshed.

* **Expected Result:** The dashboard's layout and theme are changed according to the user's selections.

Test Case 3

* **Test Case ID:** FT003
* **Test Case Title:** Verify that users can save and update their dashboard settings.
* **Test Case Description:** This test case will confirm that users can store their preferred dashboard configuration and modify it later.
* **Test Suite:** Customizable Dashboards
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the Fitness Tracker App.
    * A default dashboard is loaded.
* **Test Data:** Data representing modifications to the dashboard (added/removed widgets, layout changes, theme swap)
* **Test Steps:**
    1. Customize the dashboard layout, theme, and select metrics to display.
    2. Save the newly configured dashboard.
    3. Log out of the app and log back in.
    4. Navigate to the dashboard.
* **Postconditions:**
    * Dashboard is displayed.
* **Expected Result:** The user's saved dashboard customizations are loaded, reflecting the changes made earlier.

Test Case 4

* **Test Case ID:** FT004

* **Test Case Title:** Verify that dashboard displays are dynamic and update in real-time.

* **Test Case Description:** This test case checks if the dashboard dynamically reflects changes in the user's fitness data as they occur.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

    * User is logged in and has an active wearable device or fitness tracker syncing with the app.

* **Test Data:** User starts exercising (walking, running, cycling), data will be tracked automatically.

* **Test Steps:**

    1. Navigate to the dashboard displaying metrics such as steps taken, distance, and calories burned.

    2. Initiate a workout or physical activity that will affect these metrics.

    3. Observe the dashboard display while the activity is ongoing.

* **Postconditions:**

    * Workout continues for a few minutes.

* **Expected Result:** The relevant metrics (steps, distance, calories) on the dashboard update in real-time as the user exercises.

Test Case 5

* **Test Case ID:** FT005

* **Test Case Title:** Verify that users can reset the dashboard to default settings.

* **Test Case Description:** This test case ensures the ability to revert to the app's pre-defined dashboard layout, theme, and metrics selection.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in.

    * The dashboard has been customized.

* **Test Data:** Default dashboard configuration (pre-set metrics, layout, theme)

* **Test Steps:**

    1. Navigate to the dashboard customization settings.

    2. Find the option to "Reset to Default" or "Restore Default Settings."

    3. Select the option to reset.

* **Postconditions:**

    * The dashboard updates.

* **Expected Result:** The dashboard reverts to its default configuration, displaying the originally included metrics and layout.

Data - 91

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Social Sharing

Feature description -

Users can share workout summaries, achievements, and progress updates on social media platforms, inspiring others and fostering accountability. Social sharing enables users to celebrate their successes, seek support from their social networks, and stay motivated on their fitness journey.

Userstory title -

As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

Acceptance criteria -

['Users can share workout summaries and achievements.', 'The app supports sharing on multiple social media platforms.', 'Users can customize the content and format of shared posts.', 'Sharing features are integrated seamlessly with social media.', 'Users can track the engagement of their shared posts.']

Generated Test Cases:

Here are 5 test cases for the "Social Sharing" feature of the Fitness Tracker App, based on the provided acceptance criteria and template:

Test Case ID: FCSA_1

Test Case Title: Verify that users can share workout summaries.

Test Case Description: This test case verifies the functionality of sharing workout summaries to social media platforms.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

* User is logged into the app.

* User has completed at least one workout.

* A social media account is connected to the app.

Test Data:

* Details of completed workout (date, duration, type, calories burned, etc.)

Test Steps:

1. Navigate to the workout history section.

2. Select a completed workout to view its summary.

3. Tap the 'Share' button on the workout summary.

4. Choose a social media platform from the list (e.g., Facebook, Twitter).

5. Review and optionally customize the pre-populated sharing text and/or images.

6. Click 'Share'.

Postconditions:

* The workout summary is posted to the selected social media platform.

* The user sees a confirmation message indicating successful sharing.

Expected Result:

The workout summary, with any customization, is successfully shared on the chosen social media platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FCSA_2

Test Case Title: Verify sharing on multiple social media platforms.

Test Case Description: This test case ensures that the app supports sharing workout achievements to various social media platforms.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

* User is logged into the app.
* User has achieved a fitness milestone (e.g., reached a weight loss target).
* The user has connected at least two different social media accounts.

Test Data:

* Details of the fitness milestone

Test Steps:

1. Navigate to the achievements section within the app.
2. Select the desired fitness milestone to share.
3. Tap the 'Share' button associated with the achievement.
4. Choose the first connected social media platform (e.g., Facebook).
5. Share the achievement.
6. Repeat steps 3-5, choosing a different connected social media platform (e.g., Twitter).

Postconditions:

* The achievement is posted successfully to both selected platforms.

Expected Result:

The workout achievement is shared successfully on both social media platforms, with no errors or crashes.

Severity: Major

Type of Testing: Functional, Cross-Platform

Test Case Approach: Positive

Test Case ID: FCSA_3

Test Case Title: Verify custom sharing functionality.

Test Case Description: This test case tests the user's ability to customize the content and format of shared posts.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

* User is logged into the app.
* User has completed a workout.

Test Data:

* Pre-written text and images for sharing.

Test Steps:

1. Navigate to the workout history section.
2. Select a completed workout and tap the 'Share' button.

3. Choose a social media platform.

4. Modify the pre-populated sharing text and add custom images.

5. Click 'Share'.

Postconditions:

* The shared post contains the customized text and images.

Expected Result:
The user's customized message and images are successfully incorporated into the shared post on the selected platform.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: FCSA_4

Test Case Title: Verify seamless integration with social media.

Test Case Description: This test case evaluates the seamless integration of social sharing features with connected social media accounts.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

* User is logged into the app.

* User is logged into their connected social media accounts (e.g., Facebook, Twitter).

Test Data:

* No dedicated test data, relies on account access.

Test Steps:

1. Navigate to the feature where sharing is enabled (e.g., workout summary, achievement).

2. Select 'Share'

3. Choose a connected social media platform. 4. Attempt to share a post without encountering unexpected pop-ups, error messages, or account authentication failures.

Postconditions:

* Sharing is completed successfully within the platform.

Expected Result:

The app seamlessly interacts with the logged-in social media accounts, allowing for smooth sharing of information without requiring additional authentication or causing errors.

Severity: Major
Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: FCSA_5
Test Case Title: Verify tracking of post engagement.
Test Case Description: This test case checks if the app tracks user engagement metrics for shared posts.
Test Suite: Social Sharing
Test Priority: Medium

Preconditions:
* User is logged into the app.
* User has successfully shared a post to a social media platform.

Test Data:

* No dedicated test data, relies on live social media interactions.

Test Steps:

1. Share a post (e.g., a workout summary) to a social media platform (Facebook or Twitter).

2. Perform actions on the shared post (e.g., like, comment, share).
3. Log out of the selected social media platform for a while (1-5 minutes).
4. Return to the fitness tracker app.
5. Go to the post sharing history

Postconditions:

* The user can view their shared post's engagement metrics (likes, comments, shares) within the app.

Expected Result:

The app accurately reflects user engagement with the shared post in its internal tracking system.

Severity: Major
Type of Testing: Functional

Test Case Approach: Positive/Negative

Data - 92

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Rest and Recovery Tracking

Feature description -

The app includes features for tracking rest days, recovery activities, and recovery metrics such as resting heart rate and sleep quality. Rest and recovery tracking helps users prioritize recovery, prevent overtraining, and optimize their overall health and performance.

Userstory title -

As a user, I want to track my rest and recovery so that I can optimize my fitness performance and prevent overtraining.

Acceptance criteria -

['Users can log rest days and recovery activities.', 'The app tracks metrics such as resting heart rate and sleep quality.', 'Users receive recommendations for recovery based on their activities.', 'The app provides alerts for insufficient recovery time.', 'Users can view trends and history of their recovery data.']

Generated Test Cases:

Test Cases for Rest and Recovery Tracking Feature

Based on the provided information, here are 5 detailed test cases for the Rest and Recovery Tracking feature of the Fitness Tracker App:

Test Case 1:

* **Test Case ID:** rtr_001
* **Test Case Title:** Verify that users can log rest days

* **Test Case Description:** This test case will ensure that users can successfully log a rest day within the app and that this information is recorded accurately.

* **Test Suite:** Rest and Recovery Tracking
* **Test Priority:** High
* **Preconditions:**
  * User is logged into the app
  * User has completed at least one exercise session

* **Test Data:** No Test data needed
* **Test Steps:**
  1. Navigate to the "Rest & Recovery" section of the app.
  2. Select the "Log Rest Day" option.
  3. Choose the date for the rest day.
  4. Confirm the log.

* **Postconditions:**
  * Rest day is displayed in the user's activity log
  * User activity on selected date reflects as a rest day

* **Expected Result:** The user successfully logs a rest day, and it appears in their activity log as a designated rest day.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** rtr_002
* **Test Case Title:** Verify that the app tracks resting heart rate

* **Test Case Description:** This test case will confirm that users can input their resting heart rate and that the app accurately stores and displays this data.

* **Test Suite:** Rest and Recovery Tracking
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the app with wearable device connection setup
    * User has inputted at least one resting heart rate measurement

* **Test Data:** Sample resting heart rate values
* **Test Steps:**
    1. Open the app and navigate to the "Rest & Recovery" section.
    2. Record a resting heart rate measurement.
    3. Repeat the step for various resting heart rate values over several days.

* **Postconditions:**
    * User's entered resting heart rate values are saved
    * Average resting heart rate trend is displayed

* **Expected Result:** The app accurately records and displays the entered resting heart rate measurements, allowing users to track trends over time.
* **Severity:** Major
* **Type of Testing:** Functional
* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** rtr_003
* **Test Case Title:** Verify that the app provides recovery recommendations

* **Test Case Description:** This test case verifies that the app provides relevant recovery recommendations based on the user's activity data, including rest days and exercise intensity.

* **Test Suite:** Rest and Recovery Tracking
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the app
    * User has logged recent exercise sessions with varying intensities
    * User has logged at least one rest day

* **Test Data:** Sample exercise data with different intensities
* **Test Steps:**

1. Perform several simulated exercise sessions within the app.

2. Ensure the user's logged exercise data shows both high-intensity and low-intensity workouts.

3. Log a rest day.

* **Postconditions:**

    * App displays recovery recommendations under "Rest & Recovery" section

    * The recommendations are specific to the logged exercise data

* **Expected Result:** Based on the logged data, the app provides personalized recovery recommendations tailored to the user's recent activity levels.

* **Severity:** Minor

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** rtr_004

* **Test Case Title:** Verify that the app issues alerts for insufficient recovery time

* **Test Case Description:** This test case examines the app's ability to alert users when they haven't taken enough rest days and risk overtraining.

* **Test Suite:** Rest and Recovery Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the app

    * User has a personalized recovery plan configured (e.g., number of needed rest days per week)

* **Test Data:** No Test data needed

* **Test Steps:**

    1. Continuously log exercise sessions without logging any rest days for a set period, exceeding the configured rest requirement.

    2. Observe notifications or warnings within the app

* **Postconditions:**

    * An alert notification is received indicating insufficient recovery time.

* **Expected Result:** The app prompts the user with an alert or notification reminding them to prioritize rest and recovery to avoid potential overtraining risks.

* **Severity:** Major

* **Type of Testing:** Functional

* **Test Case Approach:** Negative

Test Case 5:

* **Test Case ID:** rtr_005

* **Test Case Title:** Verify that users can view trends and history of their recovery data

* **Test Case Description:** This test case ensures that users can effectively view historical data and trends related to their rest days, resting heart rate, and other relevant metrics.

* **Test Suite:** Rest and Recovery Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the app
    * User has logged at least one week's worth of activity data including rest days, exercise sessions, and resting heart rate.

* **Test Data:** No Test data needed
* **Test Steps:**
    1. Navigate to the "Rest & Recovery" section and access the "History" or "Trends" feature.
    2. Review the displayed data for rest days, resting heart rate trends, and sleep quality (if applicable).

* **Postconditions:**
    * The app displays a comprehensive history of logged data over a selected period.
    * Visual representations of trends are clear and interpretable.
    * Users can download or export their recovery data.

* **Expected Result:** The app presents clear and insightful visualizations or summaries of the user's recovery data, allowing them to track progress and identify patterns.
* **Severity:** Minor
* **Type of Testing:** Functional, Data-Driven
* **Test Case Approach:** Positive

Data - 93

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Audio Coaching

Feature description -

Users can access audio-guided workouts, coaching cues, and motivational messages during their workouts, enhancing engagement and adherence. Audio coaching provides real-time feedback, encouragement, and guidance, simulating the experience of having a personal trainer or workout buddy.

Userstory title -

As a user, I want to access audio coaching during workouts so that I can stay motivated and receive real-time guidance.

Acceptance criteria -

['Users can access a variety of audio-guided workouts.', 'Audio coaching includes motivational messages and coaching cues.', 'Users can choose coaching based on workout type and intensity.', 'Audio coaching integrates seamlessly with other app features.', 'Users can control audio settings and preferences.']

Generated Test Cases:

Here are 5 test cases covering the acceptance criteria for the Audio Coaching feature in the Fitness Tracker App:

Test Case 1

Test Case ID: FC-001

Test Case Title: Verify that users can access a variety of audio-guided workouts.

Test Case Description: This test case aims to ensure the app offers a diverse selection of audio-guided workouts to cater to different user needs and preferences.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in.

- App has a pre-populated library of audio workouts.

Test Data: List of various workout types (e.g., Yoga, HIIT, Cardio, Strength Training)

Test Steps:

1. Navigate to the "Workouts" section of the app.

2. Filter workouts by type (e.g., HIIT).

3. Select an audio-guided workout from the filtered list.

4. Confirm that the selected workout features audio coaching.

Postconditions:

- Workout is loaded successfully.

- Audio coaching tracks begin.

Expected Result: A variety of audio-guided workouts across different categories become accessible to users.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: FC-002

Test Case Title: Verify that audio coaching includes motivational messages and coaching cues.

Test Case Description: This test case ensures that the app delivers motivational messages and timely coaching cues during the audio-guided workouts.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in.

- An audio-guided workout is selected and in progress.

Test Data: Sample motivational messages and coaching cues specific to the chosen workout type.

Test Steps:

1. Start an audio-guided workout.

2. Listen attentively to the audio coaching throughout the workout.

3. Note the frequency and appropriateness of motivational messages and coaching cues delivered.

Postconditions:

- Workout is completed.

Expected Result: The audio track seamlessly integrates motivational messages and coaching cues that are relevant to the workout type and intensity, keeping the user engaged and informed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: FC-003

Test Case Title: Verify that users can choose coaching based on workout type and intensity.

Test Case Description: This test case confirms the app's ability to provide tailored audio coaching based on a user's selected workout type and intensity level, ensuring a personalized experience.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

- User is logged in.

- App offers different intensities (e.g., beginners, intermediate, advanced) for selected workout types.

Test Data:

- Different workout types (e.g., Running, Yoga).

- Different intensity levels (beginner, intermediate, advanced)

Test Steps:

1. Select a workout type (e.g., Running).

2. Choose a specific intensity level (beginner).

3. Start the audio-guided workout.

4. Repeat steps 1-3, but select a different intensity level (e.g., intermediate).

5. Note the differences in coaching cues, motivational messages, and pacing throughout both workouts.

Postconditions:

- Both workouts are completed.

Expected Result:

The audio coaching adapts to the selected workout type and intensity, providing appropriate guidance and encouragement for each level.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: FC-004

Test Case Title: Verify seamless integration of audio coaching with other app features.

Test Case Description: This test case assesses how audio coaching functionality interacts with other app features like workout tracking, heart rate monitoring, and progress reporting.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

- User is logged in.
- Workout tracking is enabled.
- Heart rate monitoring is connected or simulated.

Test Data:

- Data about user's workout history (e.g., distances covered, calories burned).

Test Steps:

1. Start an audio-guided workout.
2. Monitor the workout performance data (e.g., distance, heart rate) in real-time.
3. Note how audio coaching provides feedback and guidance based on the user's performance data.
4. After finishing the workout, review the workout summary and progress charts (if available).

Postconditions:

- Workout is completed and data is recorded.

Expected Result: The app seamlessly integrates audio coaching with other features, providing real-time feedback, guidance, and progress updates based on user performance and various metrics.

Severity: Major

Type of Testing: Integration, Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: FC-005

Test Case Title: Verify that users can control audio settings and preferences.

Test Case Description: This test case ensures that users have options to customize their audio coaching experience, like adjusting volume, selecting voice, and enabling/disabling features.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

- User is logged in.

- An audio-guided workout is selected.

Test Data: Adjust

- Volume level, toggling voice between available options.

Test Steps:

1. Access audio settings within the workout interface.

2. Adjust the volume level to different increments.

3. Select different voice options available (if any).

4. Toggle on/off specific audio features (e.g., motivational messages, real-time guidance).

5. Observe how the chosen settings affect the audio playback during the workout.

Postconditions:

- Workout is playing with the new settings.

Expected Result: Users can modify audio settings and preferences to personalize their coaching experience, enabling a more comfortable and enjoyable workout.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

These test cases cover all the acceptance criteria and provide a structured approach to ensure a high-quality audio coaching feature.

Data - 94

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Health Apps

Feature description -

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

Userstory title -

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

Acceptance criteria -

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

Generated Test Cases:

Here are 5 test cases for the Fitness Tracker App:

Test Case 1:

* **Test Case ID:** FIT-001
* **Test Case Title:** Verify that the app integrates with popular fitness calorie counter apps.
* **Test Case Description:** This test case aims to verify the functionality of integrating the Fitness Tracker App with a popular calorie counter application. It will focus on successful data syncing between the two apps.
* **Test Suite:** Health App Integration
* **Test Priority:** High
* **Preconditions:**
    * User has a registered account on the Fitness Tracker App.
    * User has an account on a popular calorie counter app (e.g., MyFitnessPal).
    * Both apps are installed on the device.
* **Test Data:** Sample calorie intake and expenditure data from the calorie counter app.
* **Test Steps:**
    1. On the Fitness Tracker App, navigate to the Health Integration settings.
    2. Select the preferred calorie counter app from the list of available integrations.
    3. Authorize the Fitness Tracker App to access the calorie counter app data.
    4. Observe if the calorie counter app data (e.g., daily calorie intake, calories burned) is successfully synchronized into the Fitness Tracker App.
* **Postconditions:**
    * Data from the calorie counter app is synced with the Fitness Tracker App.

* **Expected Result:** The calorie data from the selected calorie counter app is successfully integrated into the Fitness Tracker App, providing a consolidated view of the user's calorie intake and expenditure.
* **Severity:** Critical
* **Type of Testing:** Functional, Integration Testing
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** FIT-002
* **Test Case Title:** Verify that users can manage integration settings for health apps.
* **Test Case Description:** This test case aims to ensure users can easily configure and modify integrations with different health apps. It will test adding, removing, and reconfiguring connections to health platforms within the Fitness Tracker App.
* **Test Suite:** Health App Integration
* **Test Priority:** High
* **Preconditions:**
    * User has a registered account on the Fitness Tracker App.
    * User has connected at least one health app (e.g., a meditation app)
* **Test Data:** Details for a new health app integration (name, authorization credentials).
* **Test Steps:**
    1. Open the Health Integration settings within the Fitness Tracker App.
    2. Select the option to add a new integration.
    3. Choose a health app from the list of supported integrations.
    4. Enter the required authorization credentials for the selected health app.
    5. Verify the successful connection of the new health app.
    6. Repeat steps 1-5 to remove the previously connected health app.
    7. Attempt to re-add the removed health app.

* **Postconditions:**
    * The user can successfully add and remove health app integrations.
* **Expected Result:** Users are able to add, remove, and reconfigure health app integrations seamlessly within the Fitness Tracker App settings.
* **Severity:** Critical
* **Type of Testing:** Functional, Usability Testing
* **Test Case Approach:** Positive, Negative

Test Case 3:
* **Test Case ID:** FIT-003
* **Test Case Title:** Verify that the app provides a consolidated view of health metrics from integrated apps.
* **Test Case Description:** This test case aims to ensure the Fitness Tracker App effectively combines data from various integrated health apps into a unified dashboard.
* **Test Suite:** Health App Integration
* **Test Priority:** High
* **Preconditions:**
    * User has a registered account on the Fitness Tracker App
    * User has integrated at least three different health app types: calorie counter, meditation app, and sleep tracking app.
    * Data is available in all integrated health apps.
* **Test Data:** Sample data from the integrated calorie counter, meditation, and sleep tracking apps.
* **Test Steps:**
    1. Log in to the Fitness Tracker App and navigate to a dedicated dashboard or overview screen.
    2. Verify that data from all integrated apps (calorie intake/expenditure, meditation sessions, sleep duration/quality) is displayed cohesively in the dashboard.
    3. Identify if the dashboard provides visualizations or charts representing the consolidated data.

* **Postconditions:**
    * Dashboard displays combined data from various health apps.
* **Expected Result:** The Fitness Tracker App's dashboard provides a clear and comprehensive view of the user's overall health data by integrating data from different apps, including graphs, charts, or summary statistics.
* **Severity:** Major
* **Type of Testing:** Functional, Usability Testing
* **Test Case Approach:** Positive

Test Case 4:
* **Test Case ID:** FIT-004
* **Test Case Title:** Verify that the app provides insights based on combined health data.
* **Test Case Description:** This test case focuses on ensuring the app offers meaningful insights or recommendations derived from the combined data of integrated apps, promoting user awareness about their health patterns.
* **Test Suite:** Health App Integration
* **Test Priority:** High
* **Preconditions:**
    * User has a registered account on the Fitness Tracker App.
    * User has integrated a calorie counter app, a sleep tracking app, and a meditation app.
    * Data has been consistently logged for at least a week.
* **Test Data:** Sample data from the integrated health apps covering one week.
* **Test Steps:**
    1. Log in to the Fitness Tracker App and access the health insights or recommendations section.
    2. Examine the insights offered by the app based on the integrated data.
    3. Ensure the insights relate to the user's health patterns and provide actionable advice or suggestions.
* **Postconditions:**

    * Insights are generated based on the combined health data.
* **Expected Result:** The app analyzes the integrated health data and provides personalized insights, such as identifying patterns in calorie intake and exercise, sleep quality affecting meditation performance, or the correlation between stress levels and sleep duration.
* **Severity:** Major
* **Type of Testing:** Functional, Usability Testing
* **Test Case Approach:** Positive

Test Case 5:
* **Test Case ID:** FIT-005
* **Test Case Title:** Verify that integration settings are user-friendly and easy to manage.
* **Test Case Description:** This test case aims to assess the ease of use and clarity of the health app integration settings within the Fitness Tracker App.
* **Test Suite:** Health App Integration
* **Test Priority:** Medium
* **Preconditions:**
    * User has a registered account on the Fitness Tracker App
* **Test Data:** N/A
* **Test Steps:**
    1. Navigate to the Health integration settings within the Fitness Tracker App.
    2. Observe the layout and organization of the settings menu.
    3. Assess the clarity and readability of the labels and instructions for each setting.
    4. Attempt to add, remove, and modify health app integrations, noting the ease of the process.
* **Postconditions:**
    * User can manage health app integrations without difficulty.
* **Expected Result:** Users can easily understand and navigate the Health Integration settings, with clear

instructions and a user-friendly interface for managing connected apps.

Data - 95

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Body Composition Tracking

Feature description -

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and the effectiveness of exercise and nutrition interventions.

Userstory title -

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed decisions about my training and nutrition.

Acceptance criteria -

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set goals for body composition metrics.']

Generated Test Cases:

Fitness Tracker App: Body Composition Tracking - Test Cases

Test Case ID: BFCT-001

Test Case Title: Verify that users can log body composition metrics.

Test Case Description: This test case verifies the functionality of logging weight, body fat percentage, and muscle mass in the body composition tracking feature.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged into the app.
- User has access to the body composition tracking feature.

Test Data:

- Weight: 75kg
- Body fat percentage: 15%
- Muscle mass: 40kg

Test Steps:

1. Navigate to the body composition tracking section.
2. Tap on the "Add New Entry" button.
3. Enter the weight, body fat percentage, and muscle mass values.
4. Tap on the "Save" button.

Postconditions:

- New entry is added to the body composition history.

Expected Result: The app accurately records the entered body composition data and displays it in the user's history.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BFCT-002

Test Case Title: Verify that the app calculates and displays BMI.

Test Case Description: This test case verifies the app's ability to calculate and display BMI based on the logged weight and height.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User has logged weight data.
- User has logged height data.

Test Data:

- Weight: 75kg
- Height: 1.80m

Test Steps:

1. Navigate to the body composition tracking section.
2. View the calculated BMI value displayed on the screen.

Postconditions:

- BMI value is displayed

Expected Result: The app accurately calculates the BMI using the user's logged weight and height and displays it correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BFCT-003

Test Case Title: Verify that users can view trends and history of body composition changes.

Test Case Description: This test case verifies the functionality of viewing trends and history of body composition data logged by the user.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User has logged multiple body composition entries.

Test Data:

- Multiple body composition entries with varying weights, body fat percentages, and muscle mass.

Test Steps:

1. Navigate to the body composition tracking section.
2. Tap on the "History" tab.
3. Verify the chronological order of body composition entries.

Postconditions:

- Body composition history is accessible.

Expected Result: The app displays a clear and chronological history of the user's logged body composition data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BFCT-004

Test Case Title: Verify that the app provides insights and recommendations based on body composition data.

Test Case Description: This test case verifies the app's ability to analyze body composition data and provide relevant insights and recommendations.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User has logged consistent body composition data for at least two weeks.

Test Data:

- Weight, body fat percentage, and muscle mass data logged consistently over two weeks.

Test Steps:

1. Navigate to the body composition tracking section.
2. Review the insights and recommendations provided by the app based on the logged data.

Postconditions:

- Insights and recommendations are displayed.

Expected Result: The app analyzes user's body composition trends and suggests personalized insights, including potential areas for improvement and tailored recommendations for training, nutrition, or both.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BFCT-005

Test Case Title: Verify that users can set goals for body composition metrics.

Test Case Description: This test case verifies the functionality of setting weight, body fat percentage, muscle mass, and BMI goals.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged into the app.
- User has access to the goals setting feature.

Test Data:

-Desired weight: 70kg

-Desired body fat percentage: 12%

-Desired muscle mass: 45kg

-Desired BMI: 22.5

Test Steps:

1. Navigate to the body composition tracking section.

2. Tap on the "Set Goals" button.

3. Enter the desired values for each metric.

4. Save the goals.

Postconditions:

- Goals are saved in the app.

Expected Result: The app allows the user to set individual goals for each desired body composition metric and stores them securely within the user profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 96

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Analysis

Feature description -

The app offers detailed workout analysis tools, including performance metrics, trends, and insights into exercise intensity, duration, and calorie burn. Workout analysis helps users assess the effectiveness of their workouts, identify areas for improvement, and optimize training strategies for better results.

Userstory title -

As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

Acceptance criteria -

['Users can view detailed performance metrics for each workout.', 'The app provides trends and insights into workout intensity and duration.', 'Users can track calorie burn and other health metrics.', 'The app offers recommendations based on workout analysis.', 'Users can compare current workouts with past performance.']

Generated Test Cases:

Here are five test cases for the Fitness Tracker App's Workout Analysis feature, adhering to the provided template and addressing all acceptance criteria:

Test Case 1

Test Case ID: FIT-001

Test Case Title: Verify that users can view detailed performance metrics for each workout

Test Case Description: This test case verifies the display of detailed performance metrics for a single workout recorded by the user.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User is logged in
- User has completed and saved at least one workout

Test Data: A workout with recorded data on duration, distance, heart rate, and calories burned.

Test Steps:

1. Navigate to the workout history section of the app.
2. Select the previously completed workout.
3. Verify that the detailed performance metrics (duration, distance, heart rate, calories burned) are displayed accurately.

Postconditions:

- Workout details are displayed

Expected Result: The app displays all the relevant performance metrics for the selected workout accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: FIT-002

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration

Test Case Description: This test case ensures the app correctly analyzes workout data to provide trends and insights regarding intensity and duration over time.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User is logged in
- User has completed and saved multiple workouts over a period of time.

Test Data: Workout history with varying intensity levels and durations.

Test Steps:

1. Access the workout analysis or trend section of the app.
2. Examine the displayed trends for workout intensity (e.g., average heart rate, max heart rate) and duration.
3. Verify that the trends accurately reflect the patterns in the user's workout history.

Postconditions:

- Trends are displayed

Expected Result: The app generates trend lines and insights that clearly demonstrate changes in workout intensity and duration over the specified period.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: FIT-003

Test Case Title: Verify that users can track calorie burn and other health metrics

Test Case Description: This test case checks the accuracy of calorie burn tracking and the display of other relevant health metrics associated with each workout.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User is logged in
- The user's profile has accurate weight and other relevant health data inputs.

Test Data: A workout with tracked calories burned, heart rate, and steps taken.

Test Steps:

1. Complete a workout recording heart rate, steps, and other relevant metrics.
2. Review the workout summary, focusing on the calorie burn estimate and other tracked health metrics.
3. Compare the app's calorie burn estimate with known values (if applicable or using a separate fitness tracker).

Postconditions:

- Calorie burn and other metrics are displayed

Expected Result: The app accurately calculates and displays the estimated calorie burn for the workout based on the user's input and profile data. Other health metrics (like steps and time) are also displayed accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: FIT-004

Test Case Title: Verify that the app offers recommendations based on workout analysis

Test Case Description: This test case evaluates if the app provides useful and relevant recommendations based on the user's workout data.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- User is logged in
- User has a training history with varying workout types.

Test Data: Workout history with diverse types of exercise (cardio, strength training, etc.).

Test Steps:

1. Access the workout analysis or suggestions section.
2. Review the recommendations provided by the app based on the user's workout analysis.
3. Assess if the recommendations are relevant, accurate, and offer valuable insights for improving training.

Postconditions:

- Recommendations are provided to the user

Expected Result: The app generates personalized recommendations aligned with the user's workout history and goals, suggesting improvements, new exercises, or adjustments to training intensity.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: FIT-005

Test Case Title: Verify that users can compare current workouts with past performance

Test Case Description: This test case ensures users can effectively compare their current workout metrics against their historical performance.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- User is logged in
- User has completed multiple workouts.

Test Data: Recent and past workout data with varying performance levels.

Test Steps:

1. Complete a new workout
2. Access the workout history or analysis section to review the recent workout.
3. Optionally select a past workout for comparison.
4. Compare the performance metrics of the recent workout with the selected past workout (e.g., distance, heart rate, calories burned).

Postconditions:

- Performance comparison is accessible to the user.

Expected Result: The app allows users to choose from saved workouts for comparison and displays visual representations or side-by-side comparisons of pertinent metrics, highlighting improvements or areas for development.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 97

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Weather Integration

Feature description -

Integrated weather forecasts and conditions allow users to plan outdoor workouts and activities based on current weather conditions and forecasts. Weather integration helps users make informed decisions, stay safe, and optimize their outdoor exercise routines according to weather conditions.

Userstory title -

As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

Acceptance criteria -

['The app provides current weather conditions and forecasts.', 'Users can view weather details for different locations.', 'Weather data is updated in real-time.', 'The app offers recommendations based on weather conditions.', 'Users can receive weather alerts for extreme conditions.']

Generated Test Cases:

Here are 5 test cases based on the provided input data, each covering one or more acceptance criteria:

Test Case 1

Test Case ID: WTH82I

Test Case Title: Verify that the app provides current weather conditions

Test Case Description: This test case aims to verify that the app correctly displays current weather conditions for the user's current location.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- Location services are enabled

Test Data: No test data needed

Test Steps:

1. Launch the Fitness Tracker app.
2. Navigate to the weather section.
3. Observe the displayed weather conditions (temperature, humidity, precipitation).

Postconditions: User can see the current weather information

Expected Result: The app displays accurate and up-to-date weather conditions for the user's current location.

Severity: Major

Type of Testing: Functional Test

Test Case Approach: Positive

Test Case 2

Test Case ID: R7V39S

Test Case Title: Verify that users can view weather details for different locations

Test Case Description: This test case verifies the ability to view weather information for locations other than the user's current location.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in

Test Data: City or Zip codes of different locations

Test Steps:

1. Launch the Fitness Tracker app
2. Navigate to the weather section.
3. Search for a specific location using city or zip code.
4. Observe the displayed weather details for the selected location.

Postconditions:

- Location settings are updated

Expected Result: The app accurately displays weather details for the chosen location.

Severity: Major

Type of Testing: Functional Test

Test Case Approach: Positive

Test Case 3

Test Case ID: H9D45W

Test Case Title: Verify real-time weather data updates

Test Case Description: This test case confirms that weather data in the app is refreshed in real time.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in

Test Data:

Test Steps:

1. Launch the Fitness Tracker app and navigate to the weather section.
2. Note the displayed weather conditions.
3. Wait for a period of time (e.g., 5-10 minutes).
4. Observe the changes in weather conditions if any.

Postconditions: Weather information should be current

Expected Result: The displayed weather conditions have updated to reflect any changes over time, demonstrating real-time data updates.

Severity: Major

Type of Testing: Functional Test

Test Case Approach: Positive

Test Case 4

Test Case ID: J5W28R

Test Case Title: Verify weather recommendation based on conditions

Test Case Description: This test case checks that the app offers relevant workout recommendations based on current weather conditions.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- User has set fitness goals or preferences

Test Data:

- Rainy weather conditions
- Sunny weather conditions with heat alert

Test Steps:

1. Launch the Fitness Tracker app.
2. Observe the displayed weather conditions.
3. Select the "Fitness Recommendations" or similar section within the app.
4. Analyze the recommendations provided for different weather scenarios (e.g., indoor activities for rain, water-based workouts for heat).

Postconditions: User receives personalized suggestions

Expected Result: The app generates workout recommendations that are aligned with the current weather conditions and the user's fitness profile.

Severity: Major

Type of Testing: Functional Test

Test Case Approach: Positive

Test Case 5

Test Case ID: N1P76Q

Test Case Title: Verify weather alert for extreme conditions

Test Case Description: This test case confirms that the app delivers alerts for potentially harmful weather conditions, such as extreme heat or severe storms.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- App is configured to receive weather alerts

Test Data:

Test Steps:

1. Launch the Fitness Tracker app.
2. Check the app settings to ensure weather alerts are enabled.
3. Simulate or set up a scenario where extreme weather conditions are occurring (using test data or background services).
4. Observe if the app displays a weather alert for the simulated condition.

Postconditions: User is notified about weather changes

Expected Result: When extreme weather conditions are detected, the app provides clear and timely alerts to the user.

Severity: Critical

Type of Testing: Functional Test

Test Case Approach: Positive

--

Data - 98

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Mood Tracking

Feature description -

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

Userstory title -

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

Acceptance criteria -

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and energy levels.']

Generated Test Cases:

Fitness Tracker App - Mood Tracking Test Cases

Here are 5 detailed test cases for the "Mood Tracking" feature of the Fitness Tracker App:

Test Case ID: mood-track-1

Test Case Title: Verify that users can log their mood, energy levels, and motivation

Test Case Description: This test case verifies the functionality of logging mood, energy levels, and motivation before and after workouts.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

* User is logged in to the app.

* A workout has been initiated.

Test Data:

* Predefined mood options (e.g., Happy, Sad, Neutral, Anxious)

* Energy level options (e.g., High, Medium, Low)

* Motivation options (e.g., Very Motivated, Motivated, Neutral, Demotivated)

Test Steps:

1. Start a new workout.

2. Select from the predefined mood options for "Before Workout"

3. Select from the predefined energy level options for "Before Workout"

4. Select from the predefined motivation options for "Before Workout"

5. Complete the workout.

6. Select from the predefined mood options for "After Workout"

7. Select from the predefined energy level options for "After Workout"

8. Select from the predefined motivation options for "After Workout"

9. Save the logged data.

Postconditions:

* Workout data is saved including logged mood, energy and motivation.

* The user can view their logged mood, energy, and motivation for the completed workout.

Expected Result: The system successfully logs the user's chosen mood, energy levels, and motivation for both before and after the workout. Data is stored and accessible.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case ID: mood-track-2
Test Case Title: Verify that the app provides insights into mood and energy trends over time.

Test Case Description: This test case verifies if the app generates meaningful insights into the user's mood and energy trends based on their logged data.

Test Suite: Mood Tracking
Test Priority: High
Preconditions:
* User has logged at least 5 workouts with mood, energy, and motivation data.

Test Data: Refer to test data from previous test case.

Test Steps:

1. Navigate to the "Mood & Energy" tracking section of the app.
2. View the generated insights, such as mood trends across workouts, average energy levels before and after workouts.

Postconditions:

* The app display provides a visual representation (e.g., graphs, charts) of the logged data.

Expected Result: The app displays clear and insightful visualizations depicting mood and energy trends. Trends are apparent over time, showing any correlations or patterns.

Severity: Major
Type of Testing: Functional & Analytics
Test Case Approach: Positive

Test Case ID: mood-track-3
Test Case Title: Verify that users receive recommendations based on their mood and energy data.

Test Case Description: This test case verifies the app's ability to provide personalized recommendations based on the user's mood and energy levels.

Test Suite: Mood Tracking
Test Priority: High
Preconditions:
* User has logged at least 3 workouts with mood, energy, and motivation data.
* Goal setting feature is enabled and set.

Test Data:
* Refer to test data from previous test cases.
* A defined system of personalized recommendations based on mood and energy levels.

Test Steps:
1. Log a workout with specific mood, energy, and motivation choices.

2. Review the app's suggested recommendations based on the logged data.

Postconditions:
* The app displays relevant recommendations for workout types, activities, or resources

Expected Result: The app provides contextually relevant recommendations tailored to the user's current mood and energy levels. Recommendations align with user goals and preferences.

Severity: Major
Type of Testing: Functional & Recommendation Engine
Test Case Approach: Positive

Test Case ID: mood-track-4
Test Case Title: Verify that the app supports journaling for personal reflections on mood and workouts.

Test Case Description: This test case confirms the functionality of the journaling feature for users to add personal insights and reflections.

Test Suite: Mood Tracking
Test Priority: Medium
Preconditions:
* User is logged into the app.
* A workout has been completed.

Test Data:
* User-generated text for journaling.

Test Steps:

1. Access the workout details after completion.

2. Navigate to the journaling section within the workout details.

3. Enter personal reflections on the mood, energy levels, and overall experience during the workout.

4. Save the journal entry.

Postconditions:

* The user's journal entry is saved and associated with the specific workout.

Expected Result: The app effectively captures user's written reflections, allowing for deeper self-assessment and reflection on their fitness journey and its impact on their well-being.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: mood-track-5

Test Case Title: Verify that users can set goals for improving mood and energy levels.

Test Case Description: This test case examines the functionality of setting goals for mood and energy improvement within the fitness tracker app.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

* User is logged into the app.

Test Data:

* Realistic and measurable goal values for mood and energy levels (e.g., increase average post-workout mood by 1 point)

Test Steps:

1. Navigate to the "Goals" or "Settings" section of the app.
2. Select the "Mood & Energy" goal category.
3. Set specific and measurable goals for improving mood and energy levels (e.g., average mood score before and after workouts).
4. Save the desired goals.

Postconditions:

* The user's set goals are securely stored within the app.

Expected Result: The app enables users to define personalized goals for enhancing their mood and energy levels, allowing them to track progress and stay motivated.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 99

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Creation

Feature description -

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

Userstory title -

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

Acceptance criteria -

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', "Recipes are saved in the user's profile.", 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

Generated Test Cases:

Here are 5 test cases for the Recipe Creation feature of the Recipe Sharing Platform, following the provided template:

Test Case 1

Test Case ID: rcptC1

Test Case Title: Verify that users can input and save ingredients

Test Case Description: This test case verifies the functionality of adding ingredients to a recipe and saving them successfully.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in

Test Data: List of ingredients

Test Steps:

1. Access the recipe creation page.
2. Enter ingredients one by one in the designated ingredient field.
3. Save the recipe.

Postconditions:

- Ingredient list is displayed on the saved recipe page.

Expected Result: The system successfully saves the recipe with the entered ingredients.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: rcptC2

Test Case Title: Verify that users can input and save cooking instructions

Test Case Description: This test case verifies the functionality of adding cooking instructions to a recipe and saving them successfully.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in

Test Data: Multi-step cooking instructions

Test Steps:

1. Access the recipe creation page.
2. Enter cooking instructions step-by-step in the designated instructions field.
3. Save the recipe.

Postconditions:

- Instruction steps are displayed on the saved recipe page.

Expected Result: The system successfully saves the recipe with the entered cooking instructions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: rcptC3

Test Case Title: Verify that users can upload photos for their recipes

Test Case Description: This test case verifies the functionality of uploading images to accompany a recipe.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in

- A valid image file is available

Test Data: Image file (e.g., JPG) less than 10MB

Test Steps:

1. Access the recipe creation page.
2. Select the image file and upload it.
3. Save the recipe.

Postconditions:

- Uploaded image is displayed on the saved recipe page.

Expected Result: The system successfully uploads the image and displays it with the saved recipe.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: rcptC4

Test Case Title: Verify that created recipes are accessible to other users

Test Case Description: This test case verifies that recipes created by a user are visible and accessible to other users on the platform.

Test Suite: Recipe Search and Browsing

Test Priority: High

Preconditions:

- User A creates a recipe

- User B is logged in

Test Data: A recipe created by User A

Test Steps:

1. User B browses the recipe listings on the platform.
2. Locate the recipe created by User A.
3. View the recipe details.

Postconditions:

- User B can access all details of the recipe.

Expected Result: User B can successfully view and access the recipe created by User A.

Severity: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: rcptC5

Test Case Title: Verify that users receive confirmation upon successful recipe submission

Test Case Description: This test case verifies that the user receives appropriate feedback after successfully submitting a new recipe.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Recipe with ingredients and instructions

Test Steps:

1. Access the recipe creation page.
2. Enter recipe details.
3. Click the "Save Recipe" button.

Postconditions:

- A confirmation message is displayed.

Expected Result: The system displays a confirmation message indicating that the recipe was successfully saved, and it is available in the user's profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 100

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Discovery

Feature description -

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

Userstory title -

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance criteria -

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

Generated Test Cases:

Test Cases for Recipe Discovery Feature

Based on User Story: As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance Criteria:

* Users can search for recipes using keywords and ingredients.
* Search results are relevant and accurately match the search criteria.
* Users can filter search results by cuisine type, dietary preferences, and other criteria.
* Recipes in the search results are accessible and detailed.

* Users can save search results for future reference.

Test Case 1:

Test Case ID: rsd123

Test Case Title: Verify that users can search for recipes using keywords.

Test Case Description: This test case verifies if the recipe search functionality accurately retrieves recipes based on specific keywords entered by the user.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

* User is logged in to the platform.

* A sample recipe with specific keywords is already uploaded and published on the platform.

Test Data:

* Targeted keywords relevant to the sample recipe.

Test Steps:

1. Navigate to the recipe search page.

2. Enter the targeted keywords in the search bar.

3. Click the "Search" button.

4. Verify that the search results display the sample recipe containing the entered keywords.

5. Open the search result and verify that the recipe details match the expected information.

Postconditions:

* Search results list is displayed.

Expected Result: The search should return relevant recipes containing the specified keywords.

Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case 2:

Test Case ID: ftr456
Test Case Title: Verify that users can filter search results by cuisine type.
Test Case Description: This test case checks if the filtering functionality by cuisine type correctly narrows down the search results.

Test Suite: Recipe Discovery
Test Priority: Medium
Preconditions:

* User is logged in.
* Recipes with various cuisine types are uploaded and published.

Test Data:
* A specific cuisine type (e.g., Italian).

Test Steps:

1. Navigate to the recipe search page.

2. Enter a general search term (e.g., "pasta").

3. Click the 'Cuisine' filter.

4. Select the chosen cuisine type (e.g., Italian).

5. Verify that the search results display recipes predominantly from the selected cuisine type.

Postconditions:

* Filtered results list displays only the chosen cuisine type

Expected Result: The search results should be filtered to show recipes matching the specified cuisine type.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: sdg789

Test Case Title: Verify that users can search for recipes using ingredients.

Test Case Description: This test case tests the functionality of searching for recipes based on specific ingredients.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

* User is logged in.

* Recipes containing diverse ingredients are uploaded and published.

Test Data:

* A list of ingredients (e.g., "chicken", "broccoli", "cheese").

Test Steps:

1. Navigate to the recipe search page.

2. Click the "Ingredients" search option.

3. Enter the list of ingredients.

4. Click the "Search" button.

5. Verify that the search results display recipes containing all the entered ingredients.

Postconditions:

* Search results list displays recipes containing the entered ingredients.

Expected Result: The search should return recipes that include all specified ingredients.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: hjk123

Test Case Title: Verify that search results are relevant and accurate.

Test Case Description: This test case ensures that the search algorithm delivers accurate and relevant results matching the user's search criteria.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

* User is logged in.

* A variety of recipes with distinct characteristics, cuisines, and ingredients are uploaded.

Test Data:

* Various search terms with different levels of specificity (e.g., " chocolate cake", "recipe with cinnamon", "vegan dinner").

Test Steps:

1. For each search term, perform a search on the platform.

2. Analyze the displayed search results and ensure they are relevant to the entered search term.

3. Within each search result, verify that the recipe details accurately reflect the search criteria.

Postconditions:

* Search result list displays relevant recipes based on the entered terms.

Expected Result: The search results should be precise and retrieve recipes matching the user's intent.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: lkj456

Test Case Title: Verify that search results can be saved.

Test Case Description: This test case checks if the platform allows users to save search results for later reference.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

* User is logged in.
* A complex search query with specific criteria is established.

Test Data:

* Complex search query with specific keywords, cuisine type, and dietary preferences.

Test Steps:

1. Perform the complex search on the platform.
2. Locate the "Save Search" option on the search results page.
3. Click "Save Search" and provide a name for the saved search.
4. Navigate away from the search results page.
5. Re-access the saved search from the "Saved Searches" section.

Postconditions:

* Saved search is accessible under the "Saved Searches" section.

Expected Result: The platform should successfully save the search query and allow its retrieval from the "Saved Searches" section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive
