Generated Test Cases via gemma2-9b-it LLM Model and Zero-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:

## Test Cases for User Profile Creation Feature

Project Name: Social Media Platform

Feature Name: User Profiles

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

### Test Case 1

* **Test Case ID:** TC_UP_001
* **Test Case Title:** Verify that users can add and edit their name.
* **Test Case Description:** This test case verifies the functionalities of adding and editing a user's name in their profile.
* **Test Suite:** User Profile
* **Test Priority:** High
* **Preconditions:**
    * A new user account should be created.
    * User is logged in.
* **Test Data:**
    * Valid name: "John Doe"
    * Invalid name: "12345"
    * Updated name: "Jane Doe"
* **Test Steps:**
    1. Navigate to the user profile creation page.
    2. Enter a valid name "John Doe" in the name field and click "Save".
    3. Verify that the name "John Doe" is displayed on the profile.
    4. Navigate to the profile editing page.

    5. Change the name to "Jane Doe" and click "Save".

    6. Verify that the name "Jane Doe" is displayed on the profile.

* **Postconditions:** No postconditions

* **Expected Result:** The user should be able to successfully add and edit their name in the profile.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive Testing

### Test Case 2

* **Test Case ID:** TC_UP_002

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

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

* **Test Suite:** User Profile

* **Test Priority:** High

* **Preconditions:**

    * A new user account should be created.

    * User is logged in.

* **Test Data:**

    * Valid bio: "Software Engineer passionate about building innovative solutions."

    * Invalid bio: "This is a bio that is too long and exceeds the character limit." (exceeding the allowed length)

    * Updated bio: "Data Scientist exploring the world through data"

* **Test Steps:**

    1. Navigate to the user profile creation page.

    2. Enter a valid bio "Software Engineer passionate about building innovative solutions" and click "Save".

    3. Verify that the bio is displayed correctly on the profile.
    4. Navigate to the profile editing page.
    5. Try to enter an invalid bio exceeding the character limit. Verify that an error message is displayed.
    6. Change the bio to "Data Scientist exploring the world through data" and click "Save".
    7. Verify that the updated bio is displayed on the profile.
* **Postconditions:** No postconditions
* **Expected Result:** The user should be able to successfully add and edit their bio within the allowed character limit.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive and Negative Testing

### Test Case 3

* **Test Case ID:** TC_UP_003
* **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 updating a user's profile picture.
* **Test Suite:** User Profile
* **Test Priority:** High
* **Preconditions:**
    * A new user account should be created.
    * User is logged in.
* **Test Data:**
    * Valid image file: "profile_picture.jpg" (supported format)
    * Invalid image file: "profile_picture.txt" (unsupported format)

  * Updated image file: "new_profile_picture.png"

* **Test Steps:**

  1. Navigate to the user profile editing page.

  2. Upload a valid image file "profile_picture.jpg" and click "Save".

  3. Verify the profile picture is displayed correctly.

  4. Attempt to upload an invalid image file "profile_picture.txt". Verify that an error message is displayed.

  5. Upload a new image file "new_profile_picture.png" and click "Save".

  6. Verify the new profile picture is displayed correctly.

* **Postconditions:** No postconditions

* **Expected Result:** The user should be able to upload and update their profile picture with supported image formats.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, GUI Testing

* **Test Case Approach:** Positive and Negative Testing

### Test Case 4

* **Test Case ID:** TC_UP_004

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

* **Test Case Description:** This test case verifies the functionality of setting different privacy levels for various profile fields.

* **Test Suite:** User Profile

* **Test Priority:** High

* **Preconditions:**

  * A new user account should be created.
  * User is logged in and on the profile editing page
* **Test Data:**
  * Various privacy settings: (public, private, friends only)
* **Test Steps:**
  1. Access the privacy settings section for each field (Name, Bio, Location, Interests).
  2. Change the privacy setting for each field to "Public".
  3. Verify that the field is visible to all users.
  4. Change the privacy setting for each field to "Private".
  5. Verify that the field is only visible to the user who created the profile.
  6. Change the privacy setting for each field to "Friends only".
  7. Verify that the field is visible only to the user's friends.
* **Postconditions:** No postconditions
* **Expected Result:** The user should be able to successfully set different privacy levels for each profile field.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive Testing

### Test Case 5

* **Test Case ID:** TC_UP_005
* **Test Case Title:** Verify that user receives confirmation after saving their profile.
* **Test Case Description:** This test case validates that the system provides a confirmation message to the user upon successfully saving their profile changes.

* **Test Suite:** User Profile

* **Test Priority:** Medium

* **Preconditions:**

    * A new user account should be created.

    * User is logged in and on the profile editing page.

* **Test Data:**

    * Any valid changes to the user profile

* **Test Steps:**

    1. Make changes to any field in the profile editing page.

    2. Click the "Save" button.

    3. Verify that a success confirmation message is displayed to the user.

* **Postconditions:** No postconditions

* **Expected Result:** The user should see a confirmation message indicating that their profile changes have been saved successfully.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, UI Testing

* **Test Case Approach:** Positive Testing

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:

## Test Cases for Social Media Platform - News Feed Feature

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

Test Case 1:

* **Test Case ID:** TC_NEWSFEED_001
* **Test Case Title:** Verify that the news feed displays posts from followed accounts.
* **Test Case Description:** This test case verifies that the news feed displays posts from users the logged-in user is following.
* **Test Suite:** User Interface Testing
* **Test Priority:** High
* **Preconditions:**
    * A user account needs to be created and logged in.
    * At least two other user accounts need to be created and the logged-in user needs to follow them.
* **Test Data:**
    * Usernames of accounts the logged-in user is following.
* **Test Steps:**
    1. Log into the social media platform as the created user.
    2. Navigate to the news feed section.
    3. Verify that the news feed displays posts made by the followed users.
* **Postconditions:**
    * No specific post-conditions.
* **Expected Result:** The news feed should display posts only from the followed accounts.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing, UI Testing
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TC_NEWSFEED_002
* **Test Case Title:** Verify that content is shown in chronological or algorithmic order.
* **Test Case Description:** This test case verifies that the news feed displays posts in either chronological order or algorithmic order as specified by the platform's settings.

* **Test Suite:** User Interface Testing
* **Test Priority:** High
* **Preconditions:**
    * A user account needs to be created and logged in.
    * Multiple users should post content at varying times.
* **Test Data:**
    * Timestamps of posts made by different users.
* **Test Steps:**
    1. Log into the social media platform as the created user.
    2. Navigate to the news feed section.
    3. Observe the order in which posts from different users are displayed.
    4. Compare the order with the timestamps of the posts.
* **Postconditions:**
    * No specific post-conditions.
* **Expected Result:**
    * If chronological order is enabled, posts should be displayed based on the time they were created, with the most recent post appearing first.
    * If algorithmic order is enabled, posts should be displayed based on an algorithm that considers factors like user engagement, relevance, and timeliness.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, UI Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_NEWSFEED_003

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

* **Test Case Description:** This test case verifies that users can like, comment on, and share posts in the news feed.

* **Test Suite:** User Interface Testing

* **Test Priority:** High

* **Preconditions:**

    * A user account needs to be created and logged in.

    * A post needs to exist in the news feed.

* **Test Data:**

    * Comments to be added to the post.

* **Test Steps:**

    1. Log into the social media platform as the created user.

    2. Navigate to the news feed section and locate the post.

    3. Click the "like" button and verify that the like counter increments.

    4. Click the "comment" button, enter a comment, and verify that the comment is displayed under the post.

    5. Click the "share" button and verify that the post is shared to the user's feed or social network.

* **Postconditions:**

    * No specific post-conditions.

* **Expected Result:** The user should be able to successfully like, comment on, and share the post on their feed or social network.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, UI Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TC_NEWSFEED_004

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

* **Test Case Description:** This test case verifies the real-time updating functionality of the news feed.

* **Test Suite:** Performance Testing

* **Test Priority:** Medium

* **Preconditions:**

    * A user account needs to be created and logged in.

    * Another user needs to be logged in and actively posting content.

* **Test Data:**

    * Data regarding post timestamps.

* **Test Steps:**

    1. Log into the social media platform as the created user.

    2. Navigate to the news feed section.

    3. Observe the news feed for a predefined time period.

    4. Capture the timestamps of posts displayed in the feed.

    5. Have the other user post new content repeatedly.

  6. Monitor the news feed for updates and capture the timestamps of newly displayed posts.
* **Postconditions:**
  * No specific post-conditions.
* **Expected Result:** Newly posted content from the other user should appear in the news feed within a reasonable time frame, indicating real-time updates.

* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Performance Testing, Functional Testing
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_NEWSFEED_005
* **Test Case Title:** Verify that users can filter content based on type or topic.
* **Test Case Description:** This test case verifies the filtering functionality of the news feed, allowing users to select specific content types or topics.

* **Test Suite:** User Interface Testing
* **Test Priority:** Medium
* **Preconditions:**
  * A user account needs to be created and logged in.
  * Content of various types (e.g., posts, photos, videos) and topics needs to be available.
* **Test Data:**
  * Types or topics to be filtered by.
* **Test Steps:**

1. Log into the social media platform as the created user.

2. Navigate to the news feed section.

3. Utilize the filtering options to select specific content types or topics.

4. Observe the displayed content and verify that it aligns with the selected filter criteria.

* **Postconditions:**

    * No specific post-conditions.

* **Expected Result:**

The news feed should display only content that matches the selected filter options.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, UI Testing

* **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:

## Social Media Platform - Test Cases for Messaging Feature

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

Test Suite: Messaging Feature

Test Cases:

Test Case ID: TC_MS_001

Test Case Title: Verify that users can send text messages

Test Case Description: This test case aims to verify that users can successfully send text messages to other users through the platform's messaging feature.

Test Priority: High

Preconditions:

* A logged-in user exists with a valid profile.

* Another logged-in user exists (recipient) for sending the test message.

* Both users have network connectivity.

Test Data:

* Predefined test messages (text content)

Test Steps:

1. Navigate to the messaging section of the application.

2. Select the recipient user from the contact list.

3. Compose a test message using the provided test text.

4. Click the "Send" button to transmit the message.

5. Verify that the message successfully appears in the recipient's messaging inbox.

Postconditions:

* The message should be displayed in the recipient's messaging inbox.

* The sender should see "Sent" status for the message.

Expected Result: The test message should be successfully sent to the recipient's inbox.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_MS_002

Test Case Title: Verify that users can send images

Test Case Description: This test case aims to verify that users can successfully attach and send images to other users through the platform's messaging feature.

Test Priority: High

Preconditions:

* A logged-in user exists with a valid profile (sender).
* Another logged-in user exists (recipient) to receive the image.
* Both users have network connectivity.

Test Data:

* Predefined test image file

Test Steps:

1. Navigate to the messaging section of the application.
2. Select the recipient user from the contact list.
3. Click the "Attach" button or equivalent icon.
4. Select the predefined test image file from the device storage.
5. Click the "Send" button to transmit the image.
6. Verify that the image successfully appears in the recipient's messaging inbox.

Postconditions:

* The image should be displayed in the recipient's messaging inbox.

* The sender should see "Sent" status for the image.

Expected Result: The test image should be successfully sent to the recipient's inbox.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_MS_003

Test Case Title: Verify that messages show read receipts

Test Case Description: This test case aims to verify that the platform correctly displays read receipts for sent messages, indicating when a recipient has viewed the message.

Test Priority: Medium

Preconditions:

* A logged-in user exists as the sender.

* Another logged-in user exists as the recipient.

* Both users have network connectivity.

Test Data:

* Predefined test message.

Test Steps:

1. Navigate to the messaging section and send the predefined test message to the recipient.

2. Observe the sender's view of the message and note its status (sending, delivered, read).

3. Direct or simulate the recipient to open the inbox and view the message.

4. Return to the sender's view and verify that the message status changes to indicate "Read".

Postconditions:

* The recipient should see the message as read.

* The sender should see a read receipt for the message.

Expected Result: The message status should change to "Read" on the sender's side after the recipient views the message.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_MS_004

Test Case Title: Verify that users are notified of new messages

Test Case Description: This test case aims to verify that the platform correctly notifies users of new incoming private messages through the notification system.

Test Priority: High
Preconditions:
* A logged-in user exists (receiver).
* Another logged-in user exists (sender) who will send a test message.
* The receiver has enabled notifications for private messages.
* Both users have network connectivity.

Test Data:
* Predefined test message.

Test Steps:
1. Ensure the receiver user profile's notification settings are enabled for private messages.
2. The sender navigates to the messaging section and sends the predefined test message to the receiver.
3. Verify that the receiver receives a notification alert on their device indicating a new private message.
4. (Optional) Check if the notification content accurately reflects the sender's information and the message content.

Postconditions:

* The receiver's inbox should display the new message from the sender.

Expected Result: The receiver should be notified of the new private message through an appropriate notification mechanism.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_MS_005

Test Case Title: Verify that users can start private conversations or group chats

Test Case Description: This test case aims to verify that users can successfully initiate both private one-on-one conversations and group chats with other users through the platform's messaging feature.

Test Priority: High

Preconditions:

* A logged-in user exists (initiator).

* Other logged-in users exist that the initiator wants to involve in a conversation (personal & group chat).

* All users have network connectivity.

Test Data:

* List of recipient usernames for group chat

Test Steps:

1. Navigate to the messaging section of the application.

2. **Private conversation:**

   * Select a recipient from the contact list.
   * Click the "Start Chat" button (or equivalent) to initiate a private conversation.

3. **Group chat:**
   * Click the "New Group" button (or equivalent).
   * Search and select multiple users from the contact list to add to the group chat.
   * Provide a name for the group chat (if required).
   * Click the "Create Group" button to initiate the group chat.
4. Verify that for both private and group chats, a new chat window opens with the designated participants.
5. Verify that the chat window displays the appropriate message icons/panels for sending text, images, or videos.

Postconditions:

* The new private or group chat should be visible in the messaging section.
* The chat should remain active and allow for ongoing communication.

Expected Result: The user should be able to start both private conversations and group chats successfully.

Severity: Critical
Priority: High
Type of Testing: Functional Testing

Test Case Approach: Positive

These test cases cover the acceptance criteria identified for the "Messaging" feature in the Social Media Platform project.

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 "Content Sharing" Feature

Here are 5 detailed test cases for the "As a user, I want to share posts so that I can express my thoughts and interests with others" user story.

Test Case ID: TC_CS_01

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

Test Case Description: This test case verifies the user's ability to create and publish a text-based post to their feed.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

* User has a valid account on the social media platform.

* User is logged in.

* User has access to create posts.

Test Data:

* Sample text for the post.

Test Steps:

1. Navigate to the "Create Post" section of the user interface.

2. Enter the sample text into the designated text field.

3. Optionally add a relevant hashtag and tag other users.

4. Click on the "Share" button to publish the post.

5. Verify that the shared post appears on the user's feed and is visible to the intended audience (followers or public).

Postconditions:

* Post should remain visible on the user's feed.

Expected Result: The user should successfully create and share a text post. The post will display on their feed and be accessible to the chosen audience.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CS_02

Test Case Title: Verify that users can create and share images

Test Case Description: This test case verifies the user's ability to upload and share images with their followers or publicly.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

* User has a valid account on the social media platform.

* User is logged in.

* User has access to create posts.

* User has an image file accessible on their device.

Test Data:

* An image file in a supported format (e.g., JPG, PNG).

Test Steps:

1. Navigate to the "Create Post" section of the user interface.

2. Select the image file from the device.

3. Optionally add a caption, hashtags, or tags.

4. Click on the "Share" button to publish the image post.

5. Verify that the shared image post appears on the user's feed and is visible to the intended audience.

Postconditions:

* Image should remain visible on the user's feed.

Expected Result: The user should successfully upload and share an image post. The image will display on their feed and be accessible to the chosen audience.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CS_03

Test Case Title: Verify that users can share videos

Test Case Description: This test case verifies the user's ability to upload and share videos with their followers or publicly.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

* User has a valid account on the social media platform.

* User is logged in.

* User has access to create posts.

* User has a video file accessible on their device.

Test Data:

* A video file in a supported format (e.g., MP4, MOV).

Test Steps:

1. Navigate to the "Create Post" section of the user interface.

2. Select the video file from the device.

3. Optionally add a caption, hashtags, or tags.

4. Click on the "Share" button to publish the video post.

5. Verify that the shared video post appears on the user's feed and is playable.

6. Verify that the video is visible to the intended audience.

Postconditions:

* Video should remain visible on the user's feed.

Expected Result: The user should successfully upload and share a video post. The video will display on their feed, be playable, and be accessible to the chosen audience.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CS_04

Test Case Title: Verify that users can edit or delete their posts

Test Case Description: This test case verifies the user's ability to modify or remove their own previously shared posts.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

* User has a valid account on the social media platform.
* User is logged in.
* User has created a post (text, image, or video).

Test Data:

* Newly created post by the user.

Test Steps:

1. Locate the user's previously created post on their feed.

2. Click on the "Edit" option available for their own post.

3. Modify the post content (text, caption, hashtags, or tags).

4. Click on the "Save" button to update the post.

5. Verify that the post content has been successfully updated.

6. Repeat steps 1-5, this time clicking on the "Delete" option.

7. Confirm the post deletion when prompted.

8. Verify that the post is no longer visible on the user's feed.

Postconditions:

* The post should no longer be visible on the user's feed if deleted.

Expected Result: The user should successfully edit and update their post content. Additionally, they should be able to delete their posts permanently.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive, Negative

Test Case ID: TC_CS_05

Test Case Title: Verify that users receive feedback on their posts

Test Case Description: This test case verifies that users receive appropriate feedback on their shared posts through likes, comments, and shares.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

* User has a valid account on the social media platform.

* User is logged in.

* User has shared a post (text, image, or video).

Test Data:

* Another user account.

Test Steps:

1. Share a post using the created user account.
2. Log in with the second user account.
3. Navigate to the shared post.
4. Click the "Like" button on the post.
5. Verify that a "Like" notification appears on the creating user's account.
6. Leave a comment on the post.
7. Verify that the creating user receives a notification about the comment.
8. Use the second user account to "Share" the post to their own feed.
9. Verify that the creating user receives a notification about the share.

Postconditions:

* Sharing notification appears on the user's account for each liked and shared post.

Expected Result: Users should receive notifications for likes, comments, and shares on their posts.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

These comprehensive test cases address all the acceptance criteria and ensure that the "Content Sharing" feature functions as intended, providing a high-quality user experience.

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:

## Test Cases for Privacy Settings Feature

Here are 5 detailed test cases for the "Privacy Settings" feature in the Social Media Platform project, following the provided template:

Test Case ID: TC_PS_001

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

Test Case Description: This test case verifies the functionality of setting a user profile to public visibility, allowing all users to view the user's information.

Test Suite: User Account Management
Test Priority: High
Preconditions:
* A new user account is created.
* The user is logged in.
* The user is navigating the platform.

Test Data: -
Test Steps:
1. Navigate to the user's profile settings.
2. Select the "Public" privacy setting option.
3. Save the changes.
4. Access the user's profile from a different user account.
5. Verify that all profile information is visible to the other user.

Postconditions:
* User profile is set to public.

Expected Result: The user's profile is successfully set to public visibility, allowing all users to view their information.

Severity: Critical
Priority: High
Type of Testing: Functional Testing

Test Case Approach: Positive Testing

Test Case ID: TC_PS_002

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

Test Case Description: This test case verifies the functionality of setting a user profile to private visibility, restricting access to only approved friends.

Test Suite: User Account Management

Test Priority: High

Preconditions:

* A new user account is created.

* The user has logged in.

* The user has added at least one friend.

Test Data: -

Test Steps:

1. Navigate to the user's profile settings.

2. Select the "Private" privacy setting option.

3. Save the changes.

4. Attempt to access the user's profile from a non-friend user account.

5. Verify that the profile is not accessible to the non-friend user.

Postconditions:

* User profile is set to private.

Expected Result: The user's profile is successfully set to private visibility, restricting access to only their approved friends.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive Testing

Test Case ID: TC_PS_003

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

Test Case Description: This test case verifies the functionality of setting a user profile to restricted visibility, allowing customization of who can see their content and interact with them.

Test Suite: User Account Management

Test Priority: High

Preconditions:

* A new user account is created.
* The user has logged in.

Test Data: -

Test Steps:

1. Navigate to the user's profile settings.
2. Select the "Restricted" privacy setting option.
3. Configure the restricted settings to allow only specific friends or groups to view content.
4. Save the changes.
5. Attempt to access the user's profile from different user accounts: approved friends, non-approved friends, and non-users.
6. Verify that content visibility aligns with the restricted settings configured.

Postconditions:

* User profile is set to restricted.

Expected Result: The user's profile is successfully set to restricted visibility, allowing them to control who can view their content and interact with them based on customizable settings.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive Testing

Test Case ID: TC_PS_004

Test Case Title: Verify that privacy settings can be changed at any time.

Test Case Description: This test case verifies the functionality of modifying privacy settings after they have been initially set.

Test Suite: User Account Management

Test Priority: Medium

Preconditions:

* A new user account is created.

* The user has logged in.

* The user has set their privacy settings to either public, private, or restricted.

Test Data: -

Test Steps:

1. Navigate to the user's profile settings.

2. Change the privacy settings to a different option.
3. Save the changes.
4. Verify that the profile visibility changes according to the new settings.

Postconditions:

* User profile privacy settings are updated.

Expected Result: The user is able to modify their privacy settings at any time, and the changes are reflected in the account's visibility.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive Testing

Test Case ID: TC_PS_005

Test Case Title: Verify that users are notified of changes in privacy settings.

Test Case Description: This test case verifies if the system sends notifications to the user when their privacy settings are changed.

Test Suite: User Notifications

Test Priority: High

Preconditions:

* A new user account is created.
* The user has logged in.

Test Data:

* Notification preferences for the user account need to be configured.

Test Steps:

1. Log in to the user account.

2. Modify the user's privacy settings from one option to another.

3. Verify that a notification is received within the platform (e.g., via a pop-up or notification center).

4. Alternatively, check for an email notification if email notifications are enabled for privacy setting changes.

Postconditions:

* User receives a notification about the privacy setting change.

Expected Result: The user receives a clear notification within the platform or via email, informing them about the recent change in their privacy settings.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive Testing

These test cases provide a solid foundation for ensuring the quality and robustness of the "Privacy Settings" feature. Remember to adapt, expand, and refine these test cases based on specific platform requirements and implementation details.

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:

## Test Cases for Social Media Platform - Notifications Feature

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

Test Case ID: TC-NOT-001

Test Case Title: Verify that users receive notifications for likes on their posts.

Test Case Description: This test case verifies that a user receives a notification when another user likes a post they have created.

Test Suite: Notification Feature

Test Priority: High

Preconditions:

* A user account is created and logged in.

* A user has created at least one post.

* Notifications are enabled for the user.

Test Data:

* Test User Account

* Test User Account (Different from Test User Account)

* A Sample Post created by the Test User Account

Test Steps:

1. Have the Test User Account create a post.

2. Have the Second Test User Account visit the Test User Account's post and click the "Like" button.

3. Verify that the Test User Account receives a notification informing them that the Second Test User Account liked their post.

4. Click on the notification and verify that it directs the Test User Account to the post that was liked.

Postconditions:

* The Test User Account remains logged in.

Expected Result: The Test User Account receives a notification indicating that they have received a like on their post. The notification provides a link to the post.

Severity: Major

Priority: High

Type of Testing: Functional, User Acceptance

Test Case Approach: Positive

Test Case ID: TC-NOT-002

Test Case Title: Verify that users receive notifications for comments on their posts.

Test Case Description: This test case verifies that when a user comments on a post, both the original poster and the commenter receive a notification.

Test Suite: Notification Feature

Test Priority: High

Preconditions:

* A user account is created and logged in.

* A user account is created and logged in.

* A user has created at least one post.

* Notifications are enabled for both the Test User Account and another user account.

Test Data:

* Test User Account

* Another User Account

* A Sample Post created by the Test User Account

Test Steps:

1. Have the Test User Account create a post.

2. Have the Second Test User Account visit the post and leave a comment.

3. Verify that the Test User Account receives a notification informing them that they have received a comment on their post.

4. Verify that the Second Test User Account also receives a notification indicating that their comment was posted.

5. Click on the notifications for both users and verify that they are directed to the correct post and comment.

Postconditions:

* Both Test User Accounts remain logged in.

Expected Result: Both the Test User Account and the Second Test User Account receive notifications informing them about the comment activity.

Severity: Major

Priority: High

Type of Testing: Functional, User Acceptance

Test Case ID: TC-NOT-003

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

Test Case Description: This test case verifies that a user receives a notification when another user sends them a friend request.

Test Suite: Notification Feature

Test Priority: High

Preconditions:

* Two user accounts are created: the Test User Account and a Second Test User Account.

Test Data:

* Test User Account

* Second Test User Account

Test Steps:

1. Have the Second Test User account send a friend request to the Test User account.

2. Verify that the Test User account receives a notification informing them of the friend request.

3. Click on the notification and verify that it displays information about the friend request from the Second Test User Account.

Postconditions:

* Both user accounts remain logged in.

Expected Result: The Test User account promptly receives a notification about the friend request, enabling them to review and accept or decline it.

Severity: Major

Priority: High

Type of Testing: Functional, User Acceptance

Test Case ID: TC-NOT-004

Test Case Title: Verify that users can customize their notification preferences.

Test Case Description: This test case ensures that users have the ability to modify their notification settings to control the type and frequency of notifications they receive.

Test Suite: Notification Feature

Test Priority: Medium

Preconditions:

* A user account is created and logged in.

Test Data:

* Test User Account

* Various notification settings (e.g., likes, comments, mentions, friend requests)

Test Steps:

1. Navigate to the user's profile settings page.

2. Locate the notification settings section.

3. Toggle various notification options on and off to adjust preferences.

4. Save the changes to the notification settings.

Postconditions:

* User remains logged in.

Expected Result: The user's selected notification settings are applied, ensuring they receive only the desired notifications.

Severity: Minor

Priority: Medium

Type of Testing: Functional, User Acceptance

Test Case ID: TC-NOT-005

Test Case Title: Verify that notifications are displayed in real-time.

Test Case Description: This test case ensures that notifications are displayed to users promptly as they occur, providing a real-time experience.

Test Suite: Notification Feature

Test Priority: High

Preconditions:

* A user account is created and logged in.

* Another user account is created and logged in.

* Notifications are enabled for both users.

Test Data:

* Test User Account

* Another User Account

Test Steps:

1. Have both users be actively viewing their notification feeds.

2. Have the Second Test User Account perform an action that triggers a notification (e.g., like a post, comment on a post, send a friend request).

3. Observe the Test User Account's notification feed and verify that the notification appears instantly.

4. Repeat the process with different actions and verify real-time notification delivery.

Postconditions:

* Both Test User Accounts remain logged in.

Expected Result: Notifications are displayed in real-time, minimizing any delay between the action and its notification.

Severity: Critical

Priority: High

Type of Testing: Functional, Performance

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:

## Test Cases for Social Media Platform - Search Functionality

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

Test Case for Search Functionality:

Test Case ID: TC_SEARCH_001

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: Social Media Platform - Search Functionality

Test Priority: High

Preconditions:

* A logged-in user is on the platform's home page.

* The search bar is accessible.

Test Data:

* Sample keywords: "social media tips", "trending topics", "funny cat videos"

Test Steps:

1. Navigate to the platform's home page.
2. Enter a sample keyword into the search bar.
3. Click the search button or press Enter.
4. Verify that search results are displayed, including content related to the entered keyword.
5. Ensure that search results are displayed in a clear and organized manner.

Postconditions:

* The user remains on the search results page.

Expected Result:

Relevant search results are displayed, encompassing posts, profiles, and topics matching the entered keyword.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_SEARCH_002

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

Test Case Description: This test case verifies that users can search for content using hashtags within the search bar.

Test Suite: Social Media Platform - Search Functionality

Test Priority: High

Preconditions:

* A logged-in user is on the platform's home page.
* The search bar is accessible.

Test Data:

* Sample hashtags: #technology, #travelphotography, #foodie

Test Steps:

1. Navigate to the platform's home page.

2. Enter a sample hashtag into the search bar, preceded by a "#" symbol.

3. Click the search button or press Enter.

4. Verify that search results are displayed, including content associated with the entered hashtag.

5. Ensure that search results are displayed in a clear and organized manner.

Postconditions:

* The user remains on the search results page.

Expected Result:

Relevant search results are displayed, encompassing posts, profiles, and topics containing the entered hashtag.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_SEARCH_003

Test Case Title: Verify that search results include profiles, posts, and topics.

Test Case Description: This test case verifies that a single search query retrieves results encompassing all three categories: profiles, posts, and topics

Test Suite: Social Media Platform - Search Functionality

Test Priority: High

Preconditions:

* A logged-in user is on the platform's home page.
* The search bar is accessible.

Test Data:

* Sample keyword: "Technology innovation"

Test Steps:

1. Navigate to the platform's home page.
2. Enter the sample keyword "Technology innovation" into the search bar.
3. Click the search button or press Enter.
4. Verify that search results include a mix of profiles (related to tech influencers or companies), posts (discussing technology trends), and topics (focused on innovation in different sectors).

Postconditions:

* The user remains on the search results page.

Expected Result:

The search results should display a balanced representation of profiles, posts, and topics relevant to the entered keyword "Technology innovation".

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_SEARCH_004

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

Test Case Description: This test case verifies that search results are prioritized based on relevance to the user's query.

Test Suite: Social Media Platform - Search Functionality

Test Priority: High

Preconditions:

* A logged-in user is on the platform's home page.

* The search bar is accessible.

Test Data:

* Sample keyword: "Latest music releases"
* Expected higher relevance results:
  * Recent posts about new music albums
  * Music-related trending topics
  * Profiles of popular musicians with recent activity
* Expected lower relevance results:
  * Older posts discussing general music genres
  * Profiles of musicians with inactive accounts

Test Steps:

1. Navigate to the platform's home page.
2. Enter the sample keyword "Latest music releases" into the search bar.
3. Click the search button or press Enter.
4. Review the search results and compare them to the expected relevance order.
5. Verify that the most relevant results appear first, followed by progressively less relevant ones.

Postconditions:

* The user remains on the search results page.

Expected Result:

Search results are displayed in a descending order of relevance, with the most relevant content appearing at the top.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_SEARCH_005

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

Test Case Description: This test case verifies that the search bar provides real-time suggestions as a user types a query.

Test Suite: Social Media Platform - Search Functionality

Test Priority: Medium

Preconditions:

* A logged-in user is on the platform's home page.
* The search bar is accessible.

Test Data:

* Sample keywords: "So", "Soc", "Socie", "Societi", "Social"

Test Steps:

1. Navigate to the platform's home page.
2. Click into the search bar.
3. Type the progressively longer keywords while observing the dropdown suggestions.

Postconditions:

* The user remains on the platform's home page within the search bar.

Expected Result:

As the user types each keyword, the search bar should display a dropdown of relevant suggestions, including profiles, hashtags, and potential search topics.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

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:

## Test Cases for Hashtags and Tagging Feature

Project: Social Media Platform

Feature: Hashtags and Tagging

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

### Test Case 1: Verify that users can add hashtags to their posts.

Test Case Description: This test case verifies the functionality of adding hashtags to a post. It checks if a user can successfully add multiple hashtags to a post, separated by spaces, and if they are properly displayed.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the platform.
* User has created a post.

Test Data: Test hashtag values like #socialmedia, #technology, #newpost

Test Steps:

1. Compose a new post with content.

2. Add multiple hashtags to the post, separated by spaces.

3. Publish the post.

4. Verify that all added hashtags are displayed correctly below the post content.

5. Verify that the hashtags are clickable and lead to a dedicated page for that hashtag.

Postconditions:

* The post with hashtags is published successfully.

Expected Result: Hashtags are added to the post and displayed correctly, and are clickable.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

### Test Case 2: Verify that users can tag other profiles, brands, or locations.

Test Case Description: This test case verifies the functionality of tagging other users, brands, or locations in a post.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the platform.
* List of existing profiles, brands, or locations is accessible.

Test Data: Test user profiles, brand names, and location names

Test Steps:

1. Compose a new post with content.
2. Start typing the username, brand name, or location name in the post.
3. Select the appropriate profile, brand, or location from the suggestions.
4. Verify that the tag is created correctly, including the "@" symbol and the chosen username, brand name, or location name.

Postconditions:

* The post with tags is published successfully.

Expected Result: Tags are added to the post and displayed correctly, and are clickable and lead to the respective tagged profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

### Test Case 3: Verify that hashtags and tags are clickable and lead to related content.

Test Case Description: This test case verifies that clicking on hashtags and tags leads users to relevant content containing those hashtags or tags.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the platform.
* Posts containing test hashtags and tags exist.

Test Data: Test hashtags and tags used in previous test cases.

Test Steps:

1. Publish a post with test hashtags and tags.

2. Click on one of the hashtags in the published post.

3. Verify that the search results page displays posts containing the clicked hashtag.

4. Click on a profile tag in the published post.

5. Verify that the user profile page of the tagged user opens.

Postconditions:

* The user is redirected to the relevant pages containing the tagged content.

Expected Result: Clicking on hashtags directs users to a page with posts containing that hashtag, while clicking on tags directs users to the respective user's profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

### Test Case 4: Verify that users can search for content using hashtags.

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

Test Suite: Search Functionality Testing

Test Priority: High

Preconditions:

* User is logged into the platform.

* Posts containing test hashtags exist.

Test Data: Test hashtag values defined in earlier test cases

Test Steps:

1. Go to the platform's search bar.

2. Type a test hashtag in the search bar.

3. Click the search button.

4. Verify that the search results page displays posts containing the searched hashtag.

Postconditions:

* The user is presented with relevant search results.

Expected Result: The search results should return posts that contain the entered hashtag.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

### Test Case 5: Verify that the system suggests popular hashtags as users type.

Test Case Description: This test case verifies the functionality of hashtag suggestion as a user types in the post.

Test Suite: User Interface Testing

Test Priority: Medium

Preconditions:

* User is logged into the platform.
* There are popular hashtags in the system.

Test Data: Test users typing various relevant keywords.

Test Steps:

1. Go to the post composer.
2. Start typing a keyword related to popular hashtags.
3. Observe the suggestion list appearing below the input field.
4. Select a suggested hashtag from the list.
5. Verify that the selected hashtag is added to the post correctly.

Postconditions:

* The user has typed relevant keywords and selected a hashtag suggestion.

Expected Result: As the user types, the system should suggest relevant and popular hashtags based on the typed keywords.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

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:

## Test Cases for Feature: Emojis and Reactions

Test Case ID: TC_EMOJ_01

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

Test Case Description: This test case verifies that users can select from a range of emojis to react to posts and that these reactions are properly displayed on the post.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

  - A logged-in user is on the platform.

   - A post exists with content.

Test Data:

- List of valid emojis provided by the platform.

Test Steps:

1. Navigate to the post.

2. Click on the emoji reaction button.

3. Select a valid emoji from the list.

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

5. Repeat steps 2-4 with different emojis from the list.

Expected Result: The selected emojis will be displayed correctly as reactions on the post.

Test Case ID: TC_EMOJ_02

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

Test Case Description: This test case verifies that users can insert emojis within their comments and messages to enhance their communication.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

   - A logged-in user is on the platform.

   - A post or user profile is open to allow commenting.

Test Data:

- A set of valid emojis.

Test Steps:

1. Initiate a comment or send a private message.

2. Use the emoji picker or keyboard shortcuts to insert a valid emoji into the message.

3. Submit the comment or message.

4. Verify that the inserted emoji is displayed correctly within the comment or message.

Expected Result: The emoji will be correctly displayed within the comment or message.

Test Case ID: TC_EMOJ_03

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

Test Case Description: This test case verifies that the platform displays a clear and accurate count of reactions received by a post.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

  - A logged-in user is on the platform.
  - A post with at least one reaction exists.

Test Data:

- Different number of reactions on a post (e.g., 1, 10, 100).

Test Steps:

1. Navigate to the post with reactions.

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

3. Refresh the page and verify the count remains consistent.

Expected Result: The reaction count will be accurately displayed and updated accordingly.

Test Case ID: TC_EMOJ_04

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

Test Case Description: This test case verifies the platform supports a variety of emoji characters as defined by industry standards.

Test Suite: Usability Testing

Test Priority: Medium

Preconditions:

   - A logged-in user is on the platform.

Test Data:

- A comprehensive list of Unicode emoji characters.

Test Steps:

1. Search for different emoji characters using the platform's search function or keyboard shortcuts (if available).

2. Attempt to use each emoji in a post or comment.

3. Verify that each emoji is successfully displayed and interpreted according to its meaning.

Expected Result: The platform will display all valid Unicode emoji characters without errors.

Test Case ID: TC_EMOJ_05

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

Test Case Description: This test case verifies that users can modify or remove their previously given reactions to posts.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

- A logged-in user has already reacted to a post.

Test Data:

- None

Test Steps:

1. Navigate to the post where the user has already reacted.

2. Click on the initially selected emoji reaction.

3. Verify the option to remove or change the reaction appears.

4. Choose a new emoji reaction or remove the existing reaction.

5. Confirm that the reaction is successfully updated or removed.

Expected Result: The user's reaction history will be updated accordingly.

Severities are **based on impact,**

Blocker: Feature unusable

Critical: Major functional impairment

Major: Significant functionality loss

Minor: Small cosmetic issue or minor functional issue

Trivial: Usability problem, no functional impact

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:

## Test Cases for Live Streaming Feature

Test Case ID: LCS-001

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

Test Case Description: This test case aims to verify the functionality of starting and stopping live streams from a user's perspective. It will involve initiating a live stream, allowing it to run for a defined period, and then successfully stopping the stream.

Test Suite: Live Streaming Feature

Test Priority: High

Preconditions:

* User has successfully logged into the platform.

* User has a profile with necessary permissions for live streaming.

* A stable internet connection is available.

* Camera and microphone are properly configured.

Test Data: No test data needed

Test Steps:

1. Navigate to the live streaming section within the platform.

2. Click the "Start Live Stream" button.

3. Observe that the camera and microphone are active, and the live stream initiates successfully.

4. Allow the stream to run for a predetermined duration (e.g., 2 minutes).

5. Click the "Stop Live Stream" button.

6. Verify that the stream ends promptly and the user is returned to the live streaming section.

Postconditions:

* No active live streams are running from the user's account.

* The live stream recording is saved as expected if the feature is enabled.

Expected Result:

The user can successfully start and stop live streams without encountering errors or unexpected behavior.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case ID: LCS-002

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

Test Case Description: This test case verifies the user's ability to control the visibility of their live streams by selecting either a public audience or a follow-only audience.

Test Suite: Live Streaming Feature

Test Priority: High

Preconditions:

* User has successfully logged into the platform.

* User has a profile with necessary permissions for live streaming.

* Stable internet connection is available.

* Camera and microphone are properly configured.

Test Data: No test data needed

Test Steps:

1. Navigate to the live streaming preparation section within the platform.

2. Ensure the "Audience" setting is available and allows for "Public" or "Followers Only" selection.

3. Select "Public" as the audience for a live stream.

4. Initiate the live stream and verify that it is accessible to all platform users.

5. Repeat steps 1-4, but select "Followers Only" as the audience.

6. Initiate a second live stream and verify that it is only accessible to the user's followers.

Postconditions: The live stream settings are saved for future broadcasts.

Expected Result: The user can successfully choose between ?Public? and ?Followers Only? audience visibility options for their live streams.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

... Continue building the remaining 3 test cases according to the template, ensuring they cover the remaining acceptance criteria. ...

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:

## Test Cases for "Share Stories" User Story

Test Case ID: TC_ST_001

Test Case Title: Verify that users can create stories with photos.

Test Case Description: This test case verifies the functionality for creating stories with photos. It will confirm that users can successfully upload a photo from their device and add it to their story.

Test Suite: Social Media Features

Test Priority: High

Preconditions:

* User is logged into the social media platform.

* User has access to the Stories feature.

* User has at least one photo saved on their device.

Test Data: A sample photo from the user's device.

Test Steps:1. Navigate to the Stories creation screen.

2. Select the "Upload Photo" option.

3. Choose the desired photo from the device gallery.

4. Verify that the selected photo is displayed in the story creation interface.

5. Proceed to publish the story.

6. Verify that the posted story is visible on the user's profile and to their followers.

Postconditions:

* The story remains visible on the user's profile for 24 hours.

Expected Result: The user should successfully create and publish a story containing the selected photo.

Severity: High

Priority: High

Type of Testing: Functional, System

Test Case Approach: Positive

Let's create 4 more test cases

Test Case ID: TC_ST_002

Test Case Title: Verify that users can create stories with videos.

Test Case Description:

This test case verifies the functionality for creating stories with videos. It will confirm that users can successfully upload a video from their device and add it to their story.

Test Suite: Social Media Features

Test Priority: High

Preconditions:

* User is logged into the social media platform.

* User has access to the Stories feature.

* User has at least one video saved on their device.

Test Data: A sample video from the user's device.

Test Steps:1. Navigate to the Stories creation screen.

2. Select the "Upload Video" option.

3. Choose the desired video from the device gallery.

4. Verify that the selected video preview is displayed in the story creation interface.

5. Proceed to publish the story.

6. Verify that the posted story is visible on the user's profile and to their followers.

Postconditions:

* The story remains visible on the user's profile for 24 hours.

Expected Result:

The user should successfully create and publish a story containing the selected video.

Severity: High

Priority: High

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case ID: TC_ST_003

Test Case Title: Verify that users can add text overlays to stories.

Test Case Description: This test case verifies the functionality for adding text overlays to stories.

It will confirm that users can type and style text within their stories.

Test Suite: Social Media Features

Test Priority: High

Preconditions:

* User is logged into the social media platform.

* User has access to the Stories feature.

Test Data: Sample text for the overlay.

Test Steps:1. Navigate to the Stories creation screen.

2. Create a new story (either with a photo or video)

3. Select the "Add Text" option in the story creation interface.

4. Type the desired text into the overlay field.

5. Use the styling options (font, size, color, alignment) to modify the text appearance.

6. Verify that the text overlay is correctly displayed on the story.

Postconditions:

* The text overlay remains on the story for 24 hours.

Expected Result: The user should be able to add and style text overlays to their stories.

Severity: High

Priority: High

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case ID: TC_ST_004

Test Case Title: Verify that stories expire after 24 hours.

Test Case Description: This test case verifies the functionality of story expiry.

It will confirm that stories are automatically removed after 24 hours.

Test Suite: Social Media Features

Test Priority: Medium

Preconditions:

* User is logged into the social media platform.

* User has created a story.

Test Data: N/A

Test Steps:1. Create a new story.

2. Wait for exactly 24 hours.

3. Access the user's story archive or profile to view if the story is still present.

4. Verify that the story is no longer visible.

Postconditions: None

Expected Result: The story should be successfully removed after 24 hours.

Severity: Major

Priority: High

Type of Testing: Functional, System

Test Case Approach: Negative

Test Case ID: TC_ST_005

Test Case Title: Verify that users can share stories publicly or with specific followers

Test Case Description:

This test case verifies the functionality of story privacy settings. It will confirm that users can choose to make stories public or share them only with specific followers.

Test Suite: Social Media Features

Test Priority: High

Preconditions:

* User is logged into the social media platform.

* User has access to the Stories feature.

* The user has at least one follower.

Test Data:

* User must be able to create a new story

Test Steps:1. Navigate to the Stories creation screen.

2. Select "Public" sharing option.

3. Publish the story.

4. Verify that the story is visible to all users on the platform, including those who are not following the user.

Postconditions:

* The story is visible to the public for 24 hours

Expected Result: The user should successfully publish a story with public visibility.

5. Navigate to the Stories creation screen.

6. Select "Selected Followers" sharing option.

7. Choose a specific group of followers to share the story with.

8. Publish the story.

9. Log into accounts of the chosen followers (and also accounts that were not chosen).

10. Verify that the story is only visible to the accounts that were explicitly designated in the sharing options.

Severity: High

Priority: High

Type of Testing: Functional, System

Test Case Approach: Positive, Negative

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:

## Analytical Insights Test Cases:

Test Case ID: TC-001

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

Test Case Description: This test case verifies that users can access and view information about the demographics of their followers, including age ranges, gender distribution, and geographic location.

Test Suite: User Analytics

Test Priority: High

Preconditions:

* A user account is created and has at least five followers.
* The user has logged in to the platform.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Analytics" tab.
3. Verify that a section displaying follower demographics is present.
4. Verify that the demographics data includes at least three of the following: age ranges, gender distribution, and geographic location.

Postconditions:

* The user remains on the Analytics tab of their profile page.

Expected Result:

The user will be able to view a clear and concise summary of their follower demographics.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC-002

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

Test Case Description: This test case verifies that users can access and view data on the reach and engagement of their posts, including likes, comments, shares, and impressions.

Test Suite: User Analytics

Test Priority: High

Preconditions:

* A user account is created and has made at least two posts.
* The user has logged in to the platform.

Test Data: Two posts created by the user with different content types (e.g., text post, image post).

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Analytics" tab.
3. Verify that a section displaying post reach and engagement metrics is present.
4. Select one of the posts and verify that the metrics displayed include: likes, comments, shares, and impressions for that specific post.

Postconditions:

* The user remains on the Analytics tab of their profile page.

Expected Result:

The user will be able to access and view detailed insights on the reach and engagement of their individual posts.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC-003

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

Test Case Description: This test case verifies that users can view historical data on the growth of their audience over time, represented as charts and graphs.

Test Suite: User Analytics

Test Priority: High

Preconditions:

* A user account is created and has a history of follower growth over a period of at least one week.
* The user has logged in to the platform.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Analytics" tab.
3. Verify that a section displaying audience growth trends is present.
4. Verify that the trends are represented visually using charts or graphs.
5. Verify that the chart or graph allows for selecting different time ranges (e.g., weekly, monthly, yearly).

Postconditions:

* The user remains on the Analytics tab of their profile page.

Expected Result:

The user will be able to see a clear and concise visualization of their audience growth over time, broken down by selected time ranges.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC-004

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

Test Case Description: This test case verifies that the displayed analytics data is presented in a clear, concise, and easy-to-understand manner for users.

Test Suite: User Interface Testing

Test Priority: Medium

Preconditions:

* A user account is created and has access to the Analytics section.
* The user has logged in to the platform.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user's profile page.

2. Click on the "Analytics" tab.

3. Review the layout and design of the analytics dashboard.

4. Assess the clarity of the presented data, including headings, labels, and units of measurement.

5. Verify that the data is presented in a logical and intuitive manner.

Postconditions:

* The user remains on the Analytics tab of their profile page.

Expected Result:

The analytics data is displayed in a user-friendly format, using clear language, appropriate visual representations, and an intuitive layout.

Severity: Major

Priority: Medium

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC-005

Test Case Title: Verify data can be filtered by time range and content type

Test Case Description: This test case verifies that users can filter the analytics data by specific time ranges and content types, allowing for focused analysis.

Test Suite: User Analytics

Test Priority: Medium

Preconditions:

* A user account is created and has a variety of posts across different time periods.
* The user has logged in to the platform.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user's profile page.
2. Click on the "Analytics" tab.
3. Identify the filter options for time range and content type.
4. Select a specific time range (e.g., last week, last month).
5. Verify that the displayed data reflects the selected time range.
6. Select a specific content type (e.g., images, videos).
7. Verify that the displayed data reflects the selected content type.

Postconditions:

* The user remains on the Analytics tab of their profile page.

Expected Result:

Users can effectively filter the analytics data by time range and content type, allowing for focused analysis and insights.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

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:

Here are 5 test cases for the "Join Events and Groups" user story, following your provided template:

## Test Case 1

Test Case ID: TC_EventsGroups_01

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

Test Case Description: This test case verifies the functionality for a user to create a new event and subsequently join that event. It will also assess the information displayed for the created event.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* A registered user with valid credentials is logged in.

Test Data:

* Event name: "Test Event"

* Event description: "A test event for QA purposes"

* Event date: [Select a future date]

* Event time: [Select a valid time]

* Event location: [Provide a valid location]

Test Steps:

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

2. Click on the "Create Event" button.

3. Fill in the event details (name, description, date, time, location) using the provided test data.

4. Click on the "Create Event" button to finalize the event creation.

5. Verify that the newly created event is displayed on the user's profile and on the "Events" page.

6. Attempt to join the newly created event. Ensure that the user successfully joins the event and their status reflects as "Attending."

Postconditions:

* The created event remains visible on the platform.

* The user can view the event details on their profile.

Expected Result: The user successfully creates and joins an event. All provided data is correctly displayed.

Severity: High

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TC_EventsGroups_02

Test Case Title: Verify that users can join interest-based groups.

Test Case Description: This test case examines the feature of joining groups based on user interests.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* A registered user with valid credentials is logged in.

Test Data:

* Group name: "Test Interest Group"

* Group description: "A group for people interested in testing."

Test Steps:

1. Navigate to the "Groups" section of the platform.

2. Search for a pre-existing group that matches the user's interests (or create a new group using the test data provided above).

3. Click on the "Join Group" button for the selected group.

Postconditions:

* The user is added to the selected group.

* The user's profile reflects the membership in the group.

Expected Result: The user successfully joins the group.

Severity: High

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TC_EventsGroups_03

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

Test Case Description: This test case assesses the functionality of creating and organizing discussions within a group by topic.

Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

* The user is a member of a group.

Test Data:

* Group name: [Name of the relevant test group]

Test Steps:

1. Navigate to the selected group.

2. Within the group, access the discussion forum or area.

3. Create a new discussion thread.

4. Provide a relevant and specific topic title for the new discussion thread.

5. Post a sample message within the new thread.

Postconditions:

* The new discussion thread is visible in the group forum.

* The topic title is clearly displayed for the new thread.

Expected Result: The user successfully creates and organizes a discussion by topic within the group.

Severity: Medium

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TC_EventsGroups_04

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

Test Case Description: This test case ensures that all essential event information is correctly displayed to users.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* A user has created an event (as in Test Case 1).

Test Data:

* Event name: "Successfully Created Event"
* Event description: "This event is for testing purposes"
* Event date: "2024-03-15"
* Event time: "19:00"
* Event location: "Park Street Cafe"

Test Steps:

1. Navigate to the details page of the event.

2. Verify that all the following details are displayed correctly:

    * Event Name: Successfully Created Event

    * Event Description: This event is for testing purposes

    * Event Date: 2024-03-15

    * Event Time: 19:00

Postconditions:

* Event details remain accessible.

Expected Result: All event details are accurately displayed.

Severity: High

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: TC_EventsGroups_05

Test Case Title: Verify that users can RSVP to events and see the list of attendees.

Test Case Description: This test case tests the RSVP functionality for events and the guest list visibility to confirm successful integration.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* The user is logged in.

* A previously created event exists (as in Test Case 1).

Test Data:

* Event name: [Name of the event from Test Case 1].

Test Steps:

1. Navigate to the details page of the event.

2. Click on the "RSVP" button.

3. Verify the user's RSVP status changes to "Attending."

4. Check that the user's name appears in the list of attendees for the event.

Postconditions:

* The user's RSVP state is persistent across page refreshes.

Expected Result: The user successfully RSVPs to the event, and their name is reflected in the attendee list.

Severity: High

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

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:

## Test Cases for Monetization Options Feature

Test Case ID: TC_MON_001

Test Case Title: Verify that users can apply for monetization options.

Test Case Description: This test case aims to validate the functionality of the application process for monetization on the social media platform. It will ensure users can access the monetization application form, submit required information, and receive a confirmation message.

Test Suite: Feature - Monetization Options

Test Priority: High

Preconditions:

* User has a registered account on the platform.
* User has a sufficient number of followers as per platform policy.
* User has created at least one public post.

Test Data: User details, follower count, sample post content.

Test Steps:

1. Log in as a registered user.
2. Navigate to the "Monetization" section of the user profile.
3. Click on the "Apply for Monetization" button.

4. Fill out the required information on the application form accurately.

5. Review the application details and submit the form.

6. Verify that a confirmation message is displayed, indicating successful submission of the application.

Postconditions:

* Application is marked as submitted in the user profile.

Expected Result: The user should be able to successfully apply for monetization options without any errors.

Severity: Critical

Priority: High

Type of Testing: System, Functional, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_MON_002

Test Case Title: Verify that monetization options include advertising revenue and sponsored content.

Test Case Description: This test case verifies the availability of advertising revenue and sponsored content as monetization options on the platform.

Test Suite: Feature - Monetization Options

Test Priority: High

Preconditions:

* User has successfully applied and been approved for monetization.

Test Data: Sample post content for advertising and sponsorships

Test Steps:

1. Log in as a monetized user.

2. Create a new post with relevant content.

3. Navigate to the monetization settings for the post.

4. Verify that the options for "Enable Advertising" and "Accept Sponsored Content" are available.

5. If applicable, explore the settings and configurations for each monetization option.

Postconditions:

* User can access settings for both advertising revenue and sponsored content.

Expected Result: Both advertising revenue and sponsored content options should be available to monetized users.

Severity: Major

Priority: Medium

Type of Testing: System, Functional, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_MON_003

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

Test Case Description: This test case evaluates the functionality of the payment system for monetized users.

Test Suite: Feature - Monetization Options

Test Priority: High

Preconditions:

* User has generated a significant amount of revenue through earnings.

* User has linked a valid payment method (e.g., bank account, PayPal).

Test Data: Test transaction amount, payment details.

Test Steps:

1. Log in as a monetized user with accumulated earnings.

2. Access the earnings dashboard or payment section.

3. Initiate a withdrawal request for the accumulated earnings.

4. Verify that the platform correctly deducts the transaction fees (if applicable).

5. Confirm the payment details and submit the withdrawal request.

Postconditions:

* A confirmation message is displayed outlining the withdrawal request.

Expected Result: The payment should be successfully processed and credited to the linked payment method within the specified timeframe.

Severity: Critical

Priority: High

Type of Testing: System, Functional, Acceptance Testing

Test Case Approach: Positive, Negative (Test scenarios with insufficient funds, invalid payment details)

---

* **Test Case ID:** TC_MON_004

Test Case Title: Verify that detailed earnings reports are available.

Test Case Description: This test case ensures that monetized users can access comprehensive and detailed reports of their earnings.

Test Suite: Feature - Monetization Options

Test Priority: Medium

Preconditions:

* User has generated earnings through various monetization methods (e.g., advertising, sponsorships).

Test Data:

Test Steps:

1. Log in as a monetized user.
2. Navigate to the earnings dashboard or reports section.
3. Verify that detailed reports are available, displaying information such as:
   * Total earnings
   * Earnings breakdown by monetization method (advertising, sponsorships)
   * Time period covered by the report
   * Payments processed
   * Transaction details

Postconditions:
* User can filter and download reports.

Expected Result: Users should be able to access detailed and customizable earnings reports, providing a clear overview of their revenue generated through different monetization avenues.
Severity: Major
Priority: Medium
Type of Testing: System, Functional, Acceptance Testing
Test Case Approach: Positive, Edge Case (Test reports for different time periods, zero earnings)

---

* **Test Case ID:** TC_MON_005

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

Test Case Description: This test case aims to ensure that the platform clearly communicates its monetization eligibility requirements to users.

Test Suite: Feature - Monetization Options

Test Priority: Medium

Preconditions:

* User is not yet monetized.

Test Data: N/A

Test Steps:

1. Navigate to the "Monetization" section of the user profile.
2. Review the information provided on monetization eligibility.
3. Verify that the following points are clearly stated:
   * Minimum follower count requirement
   * Content quality and originality guidelines
   * Any age restrictions or other platform policy compliance requirements

Postconditions:

* User has access to the information regarding monetization requirements.

Expected Result: The platform should provide a comprehensive and easily understandable explanation of the monetization eligibility requirements, leaving no room for ambiguity or confusion.

Severity: Minor

Priority: Low

Type of Testing: System, Functional, Usability Testing

Test Case Approach: Positive, Exploratory (Probe for clarity and completeness of information)

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:

## Test Cases for Social Media Platform - Explore Tab Feature

Test Case ID: TC_EXP_001

Test Case Title: Verify that trending content is displayed on the Explore tab.

Test Case Description: This test case verifies that the Explore tab displays a selection of trending content from across the platform, encompassing popular posts, accounts, and topics.

Test Suite: Explore Tab Functionality

Test Priority: High

Preconditions:

- A user is logged into the platform.
- The platform has a sufficient amount of recent user activity to generate trending content.

Test Data: No test data needed

Test Steps:

1. Navigate to the Explore tab of the platform.
2. Observe the displayed content for various post formats (text, images, videos).
3. Verify that the content appears to be trending, indicating popularity or recent surge in engagement.
4. Check for clear indicators of trendiness, such as "Trending Now" labels or trending hashtags.

Postconditions:

- No expectations for specific post content changes.

Expected Result:

The Explore tab displays a curated selection of trending content across various formats, indicating their popularity and recent engagement.

Severity: Major

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TC_EXP_002

Test Case Title: Verify that content recommendations are personalized.

Test Case Description: This test case verifies that the Explore tab recommends content personalized to user interests, engagement history, and social connections.

Test Suite: Explore Tab Functionality

Test Priority: High

Preconditions:

- The user has created a profile with interests and followed specific accounts.
- The user has engaged with certain types of content (liked, commented, shared).

Test Data: User's profile information, interests, followed accounts, engagement history

Test Steps:

1. Log in as the user with established profile data.
2. Navigate to the Explore tab.
3. Observe the recommended content and analyze its relevance to the user's interests, followed accounts, and past engagement.
4. Compare the recommended content to the user's feed to ensure distinct content suggestions.

Postconditions:

- No expectations for specific content changes or removal.

Expected Result:

The Explore tab displays personalized content recommendations relevant to the user's profile, interests, past interactions, and social connections.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Behaviour Testing

Test Case Approach: Positive

Test Case ID: TC_EXP_003

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

Test Case Description: This test case verifies that users have the option to filter the content displayed on the Explore tab by various categories or topics.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

- A user is logged into the platform.
- The platform has categorized content based on specific topics or interests.

Test Data: List of available filter categories or topics on the platform

Test Steps:

1. Navigate to the Explore tab of the platform.
2. Identify the filter options available for content categorization (e.g., "News," "Technology," "Entertainment," etc.).

3. Select one or more filter categories to narrow down the displayed content.

4. Observe the content changes to ensure it aligns with the selected filter(s).

5. Repeat steps 2-4 with different filter combinations.

Postconditions:

- Reset filter options to default after completion of test steps.

Expected Result:

The selected filters successfully refine and display content relevant to the chosen categories, allowing for targeted exploration.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TC_EXP_004

Test Case Title: Verify that the Explore Tab updates regularly with new content.

Test Case Description: This test case verifies if the Explore tab dynamically updates with fresh content at a regular interval, reflecting current trends and user engagement.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

- A user is logged into the platform.

- The platform has a mechanism for displaying trending/updated content.

Test Data: No test data needed

Test Steps:

1. Navigate to the Explore tab and note the time and content displayed.
2. Wait for a predetermined interval (e.g., 30 minutes) specified by the platform or project requirements.
3. Return to the Explore tab and observe if new content is displayed.
4. Compare the new content with the previously viewed content for differences in posts, accounts, or trends.

Postconditions:

- Refrain from manually refreshing the page during the observation period.

Expected Result:

The Explore tab refreshes and updates with new trending content within the specified interval, reflecting real-time platform activity.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, User Experience Testing

Test Case Approach: Positive

Test Case ID: TC_EXP_005

Test Case Title: Verify the "Save" functionality for Explore Tab content.

Test Case Description: This test case verifies the ability to save content from the Explore tab, enabling users to revisit it later or manage their saved items.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

- A user is logged into the platform.

Test Data: A sample post from the Explore tab

Test Steps:

1. Navigate to the Explore tab and locate the test post.
2. Click on the "Save" or corresponding button to save the post.
3. Verify that the post is successfully added to the user's saved list or designated area.
4. Navigate to the user's saved content section.
5. Confirm that the saved post is present and accessible within the saved items.

Postconditions:

- Remove the saved post from the user's saved list.

Expected Result:

Successful saving of content from the Explore tab to the user's saved list, allowing for easy retrieval and management later.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

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:

## Test Cases for Accessibility Features

Test Case ID: TC-ACC-001

Test Case Title: Verify that alt text is provided for all images

Test Case Description: This test case will verify that the platform provides descriptive alt text for all images displayed throughout the user interface.

Test Suite: Accessibility Testing

Test Priority: High

Preconditions:

* User is logged into the platform.
* Assorted pages with various images are accessible.

Test Data: None

Test Steps:

1. Navigate to different pages on the platform, including the news feed, user profile page, and community pages.
2. Identify all images displayed on each page.

3. For each image, inspect the HTML source code to locate the `alt` attribute.
4. Verify that each `alt` attribute contains a meaningful and descriptive text alternative for the image.

Postconditions:

* None

Expected Result: All images on the platform have corresponding `alt` attributes with descriptive text alternatives.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Accessibility Testing

Test Case Approach: Positive

Test Case ID: TC-ACC-002

Test Case Title: Verify that the platform is compatible with screen readers

Test Case Description: This test case will verify that the platform can be navigated and understood by users who rely on screen reader technology.

Test Suite: Accessibility Testing

Test Priority: High

Preconditions:

* User is logged into the platform using a compatible screen reader (e.g., NVDA, JAWS).

Test Data: None

Test Steps:

1. Using the screen reader, navigate through the platform's main functionalities, including the news feed, profile navigation, search bar, and posting options.
2. Verify that all interactive elements (buttons, links, form fields) are announced clearly and accurately by the screen reader.
3. Evaluate if the screen reader conveys the hierarchical structure of the page content effectively, allowing users to understand the relationships between different elements.

Postconditions:

* None

Expected Result: The platform is fully navigable and understandable by screen readers, providing clear and accurate audio feedback.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Accessibility Testing

Test Case Approach: Positive

You'll need to create more test cases to cover the remaining acceptance criteria: Keyboard shortcuts, color contrast and font size options, and accessibility documentation. Remember to adapt the template and tailor the test steps and scenarios to effectively address each specific feature.

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:

## Test Cases for Social Media Platform - Safety and Moderation Tools

Test Case ID: SM-S-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 filtering inappropriate content such as hate speech, profanity, and violence.

Test Suite: Safety & Moderation

Test Priority: High

Preconditions:

* A user account is created.

* The platform is configured with active content moderation settings.

Test Data:

* List of pre-defined inappropriate content examples (hate speech, profanity, violence)

Test Steps:

1. Log in as a user.

2. Attempt to post the pre-defined inappropriate content examples.

3. Observe the system's response to each attempt.

4. Verify that the platform successfully detects and either removes or restricts the visibility of inappropriate content.

5. Note any error messages or warnings displayed.

Postconditions:

* No inappropriate content is visible on the platform.

Expected Result: Content moderation algorithms successfully identify and filter inappropriate content, preventing its display to users.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing, Security Testing

Test Case Approach: Negative

Test Case ID: SM-S-002

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

Test Case Description: This test case ensures that users can effectively report content flagged as harmful or inappropriate.

Test Suite: Safety & Moderation

Test Priority: High

Preconditions:

* A user account is created.

* An instance of potentially inappropriate content is available on the platform.

Test Data:

* Information about the reported content and reason for reporting, e.g., hate speech, harassment, spam.

Test Steps:

1. Log in as a user.

2. Locate the pre-defined potentially inappropriate content.

3. Navigate to the reporting mechanism (e.g., flag button, report menu).

4. Select the appropriate category for the reported content and provide a clear explanation.

5. Submit the report.

Postconditions:

* A confirmation message is received indicating the report was submitted successfully.

Expected Result: Users can effectively report inappropriate content through the designated mechanism, triggering further review by the platform's moderation team.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Acceptance Testing

Test Case Approach: Positive

Test Case ID: SM-S-003

Test Case Title: Verify that blocking capabilities are available to users.

Test Case Description: This test case verifies that users can successfully block other users on the platform and that blocked users cannot interact with them.

Test Suite: Safety & Moderation

Test Priority: High

Preconditions:

* Two user accounts are created.

Test Data:

* Usernames of both accounts.

Test Steps:

1. Login as User A.
2. Identify User B on the platform.
3. Navigate to User B's profile.
4. Utilize the platform's blocking mechanism to block User B.
5. Attempt to send a message or comment to User B from User A's account.

Postconditions:

* A message stating that User B has been blocked will be displayed.

Expected Result: Blocking functionality successfully inhibits interaction from the blocking user with the blocked user, ensuring a safer experience.

Severity: Critical
Priority: High
Type of Testing: Functional Testing, User Acceptance Testing
Test Case Approach: Positive/Negative

Test Case ID: SM-S-004
Test Case Title: Verify that comment filters prevent offensive language.
Test Case Description: This test case ensures that comment filters effectively prevent the posting of offensive language on the platform.

Test Suite: Safety & Moderation
Test Priority: Medium
Preconditions:
*   A user account is created.
* The platform is configured with active comment filters.

Test Data:
* List of offensive words or phrases that should be blocked by the filters.

Test Steps:

1. Log in as a user.

2. Navigate to a post or discussion thread.

3. Attempt to post a comment containing the pre-defined offensive words or phrases.

4. Observe the system's response to each attempt.

5. Verify that the comment filters successfully block the offensive language and prevent its display.

Postconditions:

* No offensive comments appear on the platform.

Expected Result: Comment filters accurately identify and block offensive language, creating a less hostile environment for users.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, System Testing

Test Case Approach: Negative

Test Case ID: SM-S-005

Test Case Title: Verify that community guidelines are enforced consistently.

Test Case Description: This test case assesses the platform's consistency in enforcing the defined community guidelines across various user interactions and content.

Test Suite: Safety & Moderation

Test Priority: Medium

Preconditions:

* Community guidelines are clearly documented and accessible to users.

* Two or more identical violations of community guidelines are simulated by different users.

Test Data:

* Examples of violations specific to the community guidelines (e.g., harassment, spamming, impersonation).

Test Steps:

1. Simulate two or more users violating the same community guideline.

2. Document the nature of the violation for each user.

3. Monitor the platform's response to each violation.

4. Observe if appropriate action (e.g., warnings, temporary suspension, permanent ban) is taken for each violation.

5. Compare the actions taken for each violation and ensure consistency.

Postconditions:

* Appropriate actions are taken against users violating community guidelines.

* Documentation of actions taken is maintained.

Expected Result: The platform consistently enforces community guidelines, ensuring a fair and equitable experience for all users.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Negative

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:

## Social Media Platform: Multi-Language Support Test Cases

Here are five test cases for "As a user, I want multi-language support so that I can use the platform in my preferred language" user story:

Test Case ID: MSL-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 and confirms that the platform interface displays accordingly.

Test Suite: UI Testing

Test Priority: High

Preconditions:

* User has registered and logged in to the platform.

* At least three different languages are supported by the platform.

Test Data:

* Valid user account credentials.

* List of supported languages (e.g., English, Spanish, French, etc.)

Test Steps:

1. Navigate to the user profile settings.

2. Locate the language selection option.

3. Select a language from the dropdown list (different from the currently displayed language).

4. Observe if the platform interface updates and displays content in the selected language.

5. Attempt to access various platform features (news feed, messaging, profile, etc.) and verify that the language is consistent.

Postconditions:

* User profile settings are saved.

Expected Result: The platform interface successfully changes to the selected language, and all displayed content reflects the language choice.

Severity: Critical

Priority: High

Type of Testing: Functional, UI Testing

Test Case Approach: Positive

Test Case ID: MSL-002

Test Case Title: Verify that content translation features are available

Test Case Description: This test case verifies the functionality of translating posts and messages into different languages supported by the platform.

Test Suite: Functional Testing

Test Priority: High

Preconditions:

* User has registered and logged in to the platform.
* Language support for desired translation languages is implemented.

Test Data:
* A post or message content in the default language.
* List of supported translation languages.

Test Steps:
1. Create a post or send a message in the default platform language.
2. Locate the translation option within the post/message view.
3. Select a target language from the list of available translations.
4. Observe if the post/message content is correctly translated into the chosen language.

Postconditions:
* Translated post/message is displayed.

Expected Result: The platform successfully translates the content into the selected target language, maintaining accuracy and context.

Severity: Critical
Priority: High
Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: MSL-003

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

Test Case Description: This test case verifies that the user's chosen language preference is saved across different sessions and platform interactions.

Test Suite: User Session Management

Test Priority: High

Preconditions:

* User has registered and logged in to the platform.
* User has selected a language preference.

Test Data:

* Valid user account credentials.

Test Steps:

1. Login to the platform.
2. Set the language preference to a specific language.
3. Logout of the platform.
4. Login back to the platform using the same credentials.
5. Observe the platform interface language and verify it matches the previously selected preference.

Postconditions:

* User remains logged in.

Expected Result: Upon logging back in, the platform interface language displays the previously selected language preference.

Severity: Major

Priority: High

Type of Testing: Functional, Regression Testing

Test Case Approach: Positive

Test Case ID: MSL-004

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

Test Case Description: This test case verifies the platform's ability to support a sizable number of languages, covering a global user base.

Test Suite: Multi-Lingual Support

Test Priority: Medium

Preconditions:

* The platform supports a minimum of 10 languages.

* User has registered and logged in to the platform.

Test Data:

* List of supported languages.

Test Steps:

1. Login to the platform.

2. Navigate to the language selection settings.

3. Iterate through the list of available languages.

4. Select each language and verify if the platform interface seamlessly displays content in that language.

Postconditions:

* User remains logged in.

* No errors or crashes occur during language switching.

Expected Result: The platform successfully displays the UI and platform content in all supported languages without errors or inconsistencies.

Severity: Major

Priority: Medium

Type of Testing: Functional, Cross-Browser Testing

Test Case Approach: Positive

Test Case ID: MSL-005

Test Case Title: Verify Language Support for Special Characters and Formatting

Test Case Description: This test case verifies that the platform correctly handles and displays special characters and various formatting elements when content is translated.

Test Suite: Localization Testing

Test Priority: Medium

Preconditions:

* The platform supports a minimum of 3 languages with different character sets.
* User has registered and logged in to the platform.

Test Data:
* Content with special characters (e.g., emojis, accented characters, symbols) and diverse formatting (bold, italics, lists).

Test Steps:
1. Create content in the primary platform language containing special characters and formatting.
2. Translate the content into different supported languages.
3. Observe the translated content and verify that special characters are correctly rendered and formatting elements are preserved.

Postconditions:
* Translated content is viewable and readable.

Expected Result: The platform accurately translates content with special characters and preserves formatting, ensuring a consistent user experience across languages.

Severity: Major
Priority: Medium
Type of Testing: Functional, Localization Testing

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:

## Test Case Examples for Cross-Platform Integration Feature

Test Case ID: TC_XPI_001

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

Test Case Description: This test case verifies the functionality of linking user accounts with external social media platforms like Facebook, Twitter, and Google.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User account is created on the social media platform.

* User has granted necessary permissions for account linking.

Test Data: Valid login credentials for the user and selected social media platform.

Test Steps:

1. Navigate to the account settings page.

2. Locate the "Connect Accounts" section and select the desired social media platform.

3. Enter the login credentials for the selected platform.

4. Authorize the platform to access the user's account information.

5. Verify that the user's social media account is successfully linked.

Postconditions:

* User can manage connected accounts from the account settings page.

Expected Result: The user's account is successfully linked with the selected social media platform.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TC_XPI_002

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

Test Case Description: This test case verifies the functionality of cross-posting user-generated content, such as text posts, images, and videos, to other selected social media platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User account is logged in.

* User has created content (text post, image, or video).

* Social media accounts are successfully linked.

Test Data: Sample user-generated content (text, image, or video).

Test Steps:

1. Create a new text post, image post, or video post.

2. Navigate to the share options for the post.

3. Select the desired social media platform from the list of linked accounts.

4. Configure any optional settings for cross-posting (e.g., privacy settings).

5. Initiate the cross-posting process.

6. Verify that the content is successfully posted to the selected social media platform.

Postconditions:

* The content is visible on the linked social media platform.

* The user can edit or remove the cross-posted content from the linked platform.

Expected Result: The user-generated content is successfully cross-posted to the selected social media platform.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Please note that these are just examples. You will need to create additional test cases to cover all aspects of the cross-platform integration feature, including negative scenarios, edge cases, and performance testing.

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:

Here are five detailed test cases for the "Gamification and Rewards" feature:

Test Case ID: TC_GAM_01

Test Case Title: Verify that users can earn points for posting content.

Test Case Description: This test case verifies that users receive points as per predefined rules when they create a new post (text, photo, or video). It will also check for correct point attribution based on post type and engagement.

Test Suite: User Engagement

Test Priority: High

Preconditions:

* User has a registered account.

* The system is configured with point-earning rules for different post types.

Test Data:

* Test user with no initial points

* Sample content for different post types (text, image, video)

Test Steps:

1. Log in as the test user.

2. Create a text post with a minimum character count.

3. Create an image post.

4. Create a video post.

5. Verify points are awarded to the user's account after each post creation (as defined in point-earning rules).

Expected Result: Points are correctly awarded and reflected in the user's profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case ID: TC_GAM_02

Test Case Title: Verify that users can earn badges for completing specific tasks.

Test Case Description: This test case ensures that users receive badges when they achieve specific milestones or complete predefined actions within the platform, such as reaching a follower count, posting consistently, or participating in challenges.

Test Suite: User Engagement

Test Priority: Medium

Preconditions:

* User has a registered account.

* The system is configured with badges and their associated criteria.

Test Data:

* Test user with no badges

* Tasks that lead to badge awarding (e.g., achieving a follower count, creating a set number of posts)

Test Steps:

1. Log in as the test user.

2. Perform the tasks required to earn each badge (as specified in configuration).

3. Verify that the user receives the corresponding badges in their profile.

Expected Result: Badges are correctly awarded and visually displayed in the user's profile.

Severity: Critical

Priority: Medium

Type of Testing: Functional Testing

Test Case ID: TC_GAM_03

Test Case Title: Verify that leaderboards display users ranked by points correctly.

Test Case Description: This test case verifies that the leaderboard accurately displays users ranked based on their total points earned. It will also check for sorting options (e.g., by points, achievements, or recent activity) and display of relevant user information (e.g., username, points, badges).

Test Suite: User Engagement

Test Priority: High

Preconditions:

* Multiple users have registered accounts with varying points.

* The leaderboard functionality is active.

Test Data:

* User accounts with different point sums.

* Configuration for leaderboard sorting options (e.g., total points, achievements).

Test Steps:

1. Access the leaderboard section.

2. Verify that users are sorted by points in descending order by default.

3. If applicable, use sorting options to filter the leaderboard by achievements or recent activity.

4. Verify that the leaderboard correctly displays user names, points, and badges.

Expected Result:

* Leaderboard displays users correctly ranked based on points (or selected criteria)

* Display of user information (username, points, badges) is accurate.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case ID: TC_GAM_04

Test Case Title: Verify that challenges are available and users can earn rewards.

Test Case Description: This test case ensures that challenges are presented to users, users can actively participate in them, and they receive rewards upon successful completion.

Test Suite: User Engagement

Test Priority: High

Preconditions:

* The system has assigned challenges with specific tasks and reward criteria.

Test Data:

* Test user with no current rewards from challenges. Data related to challenge parameters (tasks, time limits, reward types).

Test Steps:

1. Log in as the test user.

2. Access the challenges section.

3. Select a challenge and view its details (tasks, time limit, rewards).

4. Complete all the tasks required for the challenge within the stipulated time.

5. Verify that the user receives the specified rewards (e.g., points, badges, virtual goods).

Expected Result:

* Challenges are accessible to users.

* Users can successfully complete challenges and earn the associated rewards.

* Rewards are correctly awarded and reflected in the user account.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case ID: TC_GAM_05

Test Case Title: Verify that users can view their progress and achievements.

Test Case Description: This test case ensures that users can access their individual achievement history, track their points accumulation, and view awarded badges.

Test Suite: User Profile

Test Priority: Medium

Preconditions:

* User has earned points, badges, and possibly completed challenges.

Test Data:

* Test user with a history of achieved points, badges, and potentially completed challenges.

Test Steps:

1. Log in as the test user.

2. Access the user profile or an achievements/progress section

3. Verify that the user can view their total points earned.

4. Verify that awarded badges are listed and displayed correctly.

5. Verify that the user's achievements history is accurately recorded (including challenges completed).

Expected Result: The user profile or achievements section displays accurate and up-to-date information about their points, badges, and achievements.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

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:

## Test Cases for Task Management Tool - Task Creation Feature

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

Test Case 1:

* **Test Case ID:** TC-TASKMAN-001
* **Test Case Title:** Verify that a task can be created with a title and description.
* **Test Case Description:** This test case verifies the functionality of creating a task with both a title and description fields.
* **Test Suite:** Task Management - Feature
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the Task Management Tool.
    * There are no existing tasks in the system.
* **Test Data:** "Title: Project Launch", "Description: Complete all pre-launch preparations."
* **Test Steps:**

    1. Navigate to the task creation page.
    2. Enter "Project Launch" in the task title field.
    3. Enter "Complete all pre-launch preparations." in the task description field.
    4. Click the "Create Task" button.
    5. Verify that a new task with the entered title and description is displayed on the task list.
* **Postconditions:**
    * The newly created task remains visible on the task list.
* **Expected Result:** A new task is successfully created and displayed with the specified title and description.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TC-TASKMAN-002
* **Test Case Title:** Verify that a task can be assigned to a specific team member.
* **Test Case Description:** This test case verifies the ability to assign a newly created task to a specific user.
* **Test Suite:** Task Management - Feature
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the Task Management Tool.
    * A new task has been created (see Test Case 1)
    * There is at least one other user in the system.

* **Test Data:** A task created with title "Project Launch" and assigned to "John Doe"

* **Test Steps:**

   1. Navigate to the newly created task details.

   2. Select "John Doe" from the dropdown list of team member options.

   3. Click the "Update Task" or "Save Changes" button.

   4. Verify that the task details page shows "John Doe" as the assigned team member.

* **Postconditions:**

   * The task assignment remains persistent.

* **Expected Result:** The task is successfully assigned to "John Doe".

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC-TASKMAN-003

* **Test Case Title:** Verify that a task can be created with a due date.

* **Test Case Description:** This test case verifies the functionality of setting a due date for a new task.

* **Test Suite:** Task Management - Feature

* **Test Priority:** High

* **Preconditions:**

   * User is logged into the Task Management Tool.

* **Test Data:** "Due Date: 2024-03-15"

* **Test Steps:**

   1. Navigate to the task creation page.

  2. Enter the task title, description, and a due date in the format "2024-03-15".

  3. Click the "Create Task" button.

  4. Verify that the task details page displays the entered due date.

* **Postconditions:**

  * The assigned due date remains visible in the task details.

* **Expected Result:** The task is created with the specified due date.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TC-TASKMAN-004

* **Test Case Title:** Verify that a task can be created with a priority level.

* **Test Case Description:** This test case verifies the ability to set a priority level for a new task.

* **Test Suite:** Task Management - Feature

* **Test Priority:** Medium

* **Preconditions:**

  * User is logged into the Task Management Tool.

* **Test Data:** "Priority: High"

* **Test Steps:**

  1. Navigate to the task creation page.

  2. Enter the task title, description, and select "High" from the priority level dropdown.

  3. Click the "Create Task" button.

  4. Verify that the task details page displays "High" as the priority.

* **Postconditions:**

    * The set priority level remains visible.

* **Expected Result:** The task is created with the selected priority level ("High").

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC-TASKMAN-005

* **Test Case Title:** Verify that the system validates required fields during task creation.

* **Test Case Description:** This test case checks if the system accurately identifies and handles missing required fields during task creation.

* **Test Suite:** Task Management - Feature

* **Test Priority:** High

* **Preconditions:**

    * User is logged into the Task Management Tool.

* **Test Data:**

    * Attempt to create a task without a title.

    * Attempt to create a task without a description.

* **Test Steps:**

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

    2. Observe the system's response (e.g., error message, validation alerts).

    3. Repeat the process for the description field.

* **Postconditions:**

    * All required fields are clearly identified as missing.

* **Expected Result:** The system displays an error message or validation alert for missing required fields (title and description) and prevents task creation until they are filled.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

These are just a few examples, and you would likely need to create more detailed and specific test cases depending on the specific requirements and functionalities of your Task Management Tool.

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:

## Test Cases for Task Assignment Feature

Here are five detailed test cases for the "Task Assignment" feature of your Task Management Tool:

Test Case ID: TC_TASK_ASSIGNMENT_001

Test Case Title: Verify that users can select team members to assign tasks.

Test Case Description: This test case verifies the functionality of selecting team members during task creation and ensures the assigned team member is correctly displayed on the task details page.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* A user with the necessary permissions to create tasks is logged in.

* A list of team members is available in the system.

* At least one team member exists in the system.

Test Data:

* A new task with a description

Test Steps:

1. Navigate to the task creation page.

2. Enter a task title and description.

3. Select a team member from the list of available team members.

4. Submit the task.

5. Verify that the selected team member is displayed as the assignee on the task details page.

Postconditions:

* The task should be created and stored in the system.

Expected Result: The task should be successfully assigned to the selected team member.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_ASSIGNMENT_002

Test Case Title: Verify that assignees receive notifications about their tasks.

Test Case Description: This test case ensures that assignees receive notifications when a task is assigned to them.

Test Suite: Notification Integration Testing

Test Priority: High

Preconditions:

* A user with the necessary permissions to assign tasks is logged in.

* A list of team members is available in the system.

* At least two team members exist in the system, including the assignee.

* Notification system is functioning.

Test Data:

* A new task requiring assignment

Test Steps:

1. Navigate to the task creation page.

2. Enter a task title and description.

3. Select a team member from the list of available team members.

4. Submit the task.

5. Log into the user account of the assignee.

6. Verify that a notification about the newly assigned task is received.

Postconditions:

* The assignee has viewed the notification.

Expected Result: The assignee should receive a notification about the task assignment.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_ASSIGNMENT_003

Test Case Title: Verify that the system displays assigned tasks in the assignee's task list.

Test Case Description: This test case ensures that tasks assigned to a specific user are displayed correctly in their task list.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* A user with the necessary permissions to create tasks is logged in.

* A list of team members is available in the system.

* At least two team members exist in the system, including the assignee.

* A task has been assigned to a team member.

Test Data:

* A new task assigned to a specific team member.

Test Steps:

1. Log into the user account of the assignee.

2. Navigate to the task list.

3. Verify that the assigned task is displayed in the assignee's task list.

Postconditions:

* The assignee has reviewed their task list.

Expected Result: The assigned task should be displayed in the assignee's task list.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_ASSIGNMENT_004

Test Case Title: Verify that users can reassign tasks to different team members.

Test Case Description: This test case ensures that users can change the assignment of a task to a different team member.

Test Suite: User Permissions Testing

Test Priority: Medium

Preconditions:

* A user with the necessary permissions to assign and reassign tasks is logged in.

* A task has already been created and assigned to a team member.

Test Data:

* An existing task with an assigned team member

Test Steps:

1. Navigate to the task details page.

2. Locate the option to reassign the task.

3. Select a different team member from the list of available team members.

4. Confirm the task reassignment.

5. Verify that the task status and assignee are updated in the system.

6. Log into the user account of the original assignee and verify that the task is no longer in their task list.

Postconditions:

* The task is reassigned.

Expected Result: The task should be successfully reassigned to the selected team member.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive/Negative

Test Case ID: TC_TASK_ASSIGNMENT_005

Test Case Title: Verify that task assignment history is tracked.

Test Case Description: This test case verifies the functionality of recording the history of task assignments.

Test Suite: Data Integrity Testing

Test Priority: Medium

Preconditions:

* A user with the necessary permissions to access task history is logged in.

* A task has been assigned to a team member. This team member should also be assigned to the task again.

Test Data:

* An existing task with a history of assignment changes.

Test Steps:

1. Navigate to the task details page.

2. Locate the section displaying the assignment history.

3. Verify that the history of assignments including the original assignee and the current assignee are accurately recorded in chronological order.

Postconditions:

* The task history has been reviewed.

Expected Result: The task assignment history should accurately reflect all previous assignments, including dates and assignee details.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Data Integrity 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:

## Test Cases for Task Deadline Reminders Feature

Test Case ID: TC_TASK_REMINDERS_01

Test Case Title: Verify that users can set reminders for tasks based on deadlines.

Test Case Description: This test case verifies the functionality of setting reminders for tasks based on their defined deadlines. It will ensure that users can input a date and time for the reminder and the system accurately stores and utilizes that information.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user account is created and logged in.

* A new task is created with a defined deadline.

Test Data:

* Valid date and time for a reminder.

Test Steps:

1. Navigate to the task details page for the newly created task.
2. Enter a desired date and time for the reminder in the dedicated field.
3. Save the task changes.
4. Verify that the reminder is correctly displayed in the task details and the user's reminder list.
5. Check system logs or notifications to confirm that the reminder is scheduled for the specified date and time.

Postconditions:

* The reminder should be saved and remain active for the specified task.

Expected Result:

The user is able to successfully set a reminder for the task based on its deadline.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_REMINDERS_02

Test Case Title: Verify that reminders can be customized by urgency level.

Test Case Description: This test case verifies the functionality of customizing reminder urgency levels. Different urgency levels should trigger different notification mechanisms or frequency.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user account is created and logged in.
* A new task is created with a defined deadline.

Test Data:

* Different urgency levels (e.g., High, Medium, Low)

Test Steps:

1. Navigate to the task details page for the newly created task.
2. Select different urgency levels for the reminder.
3. Verify that the system differentiates the urgency levels based on notification settings (e.g., email alerts for High urgency, in-app notification for Medium urgency, simple reminder on the task list for Low urgency).

Postconditions:

* The chosen urgency level should be reflected in the reminder settings and subsequent notifications.

Expected Result:

The user can successfully customize the urgency level of the reminder, and the system delivers notifications accordingly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_REMINDERS_03

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

Test Case Description: This test case verifies that the system accurately sends notifications for upcoming task deadlines, based on the set reminder schedules.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user account is created and logged in.
* A new task is created with a defined deadline and a reminder set.
* Email/notification channel is set up for the user.

Test Data:

* A future date for the task deadline.

Test Steps:

1. Wait for the task deadline to approach the reminder trigger point.
2. Verify that the user receives a notification (email/in-app) regarding the upcoming task deadline.
3. Check the notification content for accuracy, including the task name, deadline, and any additional information.

Postconditions:

* Notification should be delivered successfully based on the configured reminder settings.

Expected Result:

The system delivers a notification to the user upon approaching the task deadline according to the set reminder.

Severity: Critical
Priority: High
Type of Testing: Functional Testing, System Testing
Test Case Approach: Positive

Test Case ID: TC_TASK_REMINDERS_04
Test Case Title: Verify that users can choose the delivery method for reminders (e.g., email, in-app).
Test Case Description: This test case confirms that users can select their preferred method for receiving

reminders.

Test Suite: Task Management Feature Suite

Test Priority: Medium

Preconditions:

* A user account is created and logged in.

* The user has access to settings or notification preferences.

Test Data:

* Different notification delivery methods (e.g., email, push notifications, in-app message)

Test Steps:

1. Access the user's account settings or notification preferences.

2. Change the preferred delivery method for task reminders.

3. Create a new task and set a reminder for it.

4. Verify that the reminder is delivered through the selected method.

Postconditions:

* System should persist the user's chosen reminder delivery method.

Expected Result:

The user can successfully choose their preferred reminder delivery method and receive notifications accordingly.

Severity: Major
Priority: Medium
Type of Testing: Functional Testing, User Interface Testing
Test Case Approach: Positive

Test Case ID: TC_TASK_REMINDERS_05
Test Case Title: Verify that reminder settings can be adjusted or disabled.
Test Case Description: This test case examines the user's ability to modify existing reminder settings and disables reminders altogether.

Test Suite: Task Management Feature Suite
Test Priority: Medium
Preconditions:

* A user account is created and logged in.
* A task with an active reminder is created.

Test Data:

* Changes to the reminder schedule, urgency level, or delivery method.

Test Steps:

1. Navigate to the task details page with the active reminder.

2. Adjust the reminder schedule, urgency level, or delivery method.

3. Save the changes.

4. Verify that the reminder settings are updated accordingly.

5. Disable the reminder for the task.

6. Confirm that no further reminder notifications are triggered for this task.

Postconditions:

* User-made changes to the reminder settings are reflected in the system.

Expected Result:

The system allows users to modify and disable reminders as needed.

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:

## Test Cases for Task Prioritization Feature

Project Name: Task Management Tool

Feature Name: Task Prioritization

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

Test Case 1:

Test Case ID: TC_TP_001

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

Test Case Description: This test case verifies the functionality of assigning priority levels to tasks. It will create new tasks and assign them each priority level (High, Medium, and Low) independently.

Test Suite: User Acceptance Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* There are no existing tasks.

Test Data:

* Task name: Test Task 1, Priority: High

* Task name: Test Task 2, Priority: Medium

* Task name: Test Task 3, Priority: Low

Test Steps:

1. Navigate to the "Tasks" section.

2. Click on the "Create New Task" button.

3. Enter the task name "Test Task 1" and select the "High" priority level.

4. Click on the "Create Task" button.

5. Repeat steps 2-4 for "Test Task 2" with "Medium" priority and "Test Task 3" with "Low" priority.

Postconditions:

* Three tasks are created with assigned priority levels: High, Medium, and Low.

Expected Result: Tasks are created successfully and displayed with assigned priority levels.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_TP_002

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

Test Case Description: This test case verifies the functionality of sorting tasks based on their assigned priority levels.

Test Suite: User Acceptance Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* Three tasks are created (as in Test Case 1)

Test Data:

* The three tasks created in TC_TP_001.

Test Steps:

1. Navigate to the "Tasks" section.

2. Click on the "Priority" column header to sort tasks.

3. Verify that tasks are displayed in order of priority (High to Low).

Postconditions: Tasks are sorted based on priority.

Expected Result: Tasks are sorted correctly by priority level (High, Medium, Low).

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_TP_003

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

Test Case Description: This test case verifies the functionality of changing the priority level of an existing task.

Test Suite: User Acceptance Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* Three tasks are created (as in Test Case 1).

Test Data:

* "Test Task 2"

Test Steps:

1. Navigate to the "Tasks" section.

2. Select "Test Task 2".

3. Edit the task and change the priority level from "Medium" to "High".

4. Click on the "Save" button.

Postconditions: The task priority is updated.

Expected Result: "Test Task 2" is successfully updated to "High" priority.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_TP_004

Test Case Title: Verify visual distinction of tasks by priority level

Test Case Description: This test case verifies the visual clarity of priority levels for tasks.

Test Suite: User Acceptance Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* Three tasks are created (as in Test Case 1).

Test Data:

* The three tasks created in TC_TP_001 (High, Medium, Low)

Test Steps:

1. Navigate to the "Tasks" section.

2. Observe the visual representation of each task (e.g., color coding, icon, font size).

3. Verify that each priority level (High, Medium, Low) is visually distinct.

Postconditions: Tasks are visually differentiated by priority.

Expected Result: Each task is visually marked with a clear indicator representing its priority level, e.g., High tasks appear in red, Medium in blue, and Low in green.

Severity: Major

Priority: High

Type of Testing: UI Testing, Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_TP_005

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

Test Case Description: This test case verifies the default priority for tasks when no explicit priority is assigned.

Test Suite: User Acceptance Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

Test Data:

* Create a new task without specifying a priority level. Name it "Test Task 4".

Test Steps:

1. Navigate to the "Tasks" section.

2. Create a new task named "Test Task 4" without setting a priority level.

3. Confirm that the task has been created.

Postconditions:

* The new task is created.

Expected Result: The newly created task "Test Task 4" is assigned the default priority level of "Medium".

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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: TC-SUBTASK-01

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

Test Case Description: This test case verifies the functionality of creating subtasks under existing tasks. It will ensure that users can add new subtasks with distinct titles and descriptions, effectively breaking down complex tasks into smaller, manageable components.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user is logged into the system.
* A main task exists in the system.

Test Data:

* New subtask title: "Write User Stories"
* New subtask description: "Create clear and concise user stories for the application."

Test Steps:

1. Navigate to the main task view.
2. Click on the "Add Subtask" button.
3. Enter "Write User Stories" as the subtask title.
4. Enter "Create clear and concise user stories for the application." as the subtask description.
5. Click "Save" to create the subtask.

6. Verify that the new subtask is listed under the parent task with the specified title and description.

Postconditions:

* The new subtask remains visible under the parent task.

Expected Result: The subtask "Write User Stories" is successfully created and displayed under the main task with the given title and description.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-SUBTASK-02

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

Test Case Description: This test case ensures the user can set independent due dates and assign different users to subtasks. This reinforces the independent management capability of subtasks.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user is logged into the system.

* A main task exists in the system.

Test Data:

* New subtask due date: "2023-12-25"

* Assignees: "John Doe," (different from the main task assignee)

Test Steps:

1. Navigate to the main task view.

2. Click on the "Add Subtask" button.

3. Enter a title for the subtask.

4. Set the subtask due date to "2023-12-25."

5. Assign the subtask to "John Doe."

6. Click "Save" to create the subtask.

7. Verify that the subtask due date and assignee information are accurately displayed.

Postconditions:

* The subtask due date and assignee are reflected in the system.

Expected Result: The subtask is created with the specified due date and assigned to "John Doe."

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-SUBTASK-03

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

Test Case Description: This test case verifies that the system accurately reflects the progress of individual subtasks, enabling users to monitor the completion status of task components.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user is logged into the system.

* A main task with subtasks exists in the system.

Test Data:

* Subtask statuses: "Not Started," "In Progress," "Completed"

Test Steps:

1. Navigate to the main task view.

2. Select one subtask and change its status to "In Progress."

3. Select another subtask and change its status to "Completed."

4. Verify that the system displays the updated statuses for both subtasks correctly.

Postconditions:

* Subtask statuses are accurately reflected in the system.

Expected Result: The subtasks' progress statuses are correctly displayed as "In Progress" and "Completed."

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-SUBTASK-04

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

Test Case Description: This test case ensures that users can complete subtasks regardless of the main task's status, allowing for granular task management and progress tracking.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user is logged into the system.

* A main task with at least one subtask exists in the system, with the main task status set to "In Progress."

Test Data:

* Subtask marked as completed

Test Steps:

1. Navigate to the main task view.

2. Select a subtask under the main task.

3. Mark the subtask as "Completed."

4. Verify that the subtask's status is updated to "Completed" and remains separate from the main task's status.

Postconditions:

* The subtask remains marked as completed, even if the main task is not.

Expected Result: The subtask is marked as completed without affecting the main task's status.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach:

Test Case ID: TC-SUBTASK-05

Test Case Title: Verify that dependencies between subtasks can be set.

Test Case Description: This test case verifies the functionality of setting dependencies between subtasks. It will ensure that users can define a hierarchy and logical order for subtasks, ensuring tasks are completed in the correct sequence.

Test Suite: Task Management Feature Suite

Test Priority: High

Preconditions:

* A user is logged into the system.
* A main task with at least two subtasks exists in the system.

Test Data:

* Subtask 1: "Design User Interface."
* Subtask 2: "Develop Frontend."
* Dependency: Subtask 2 depends on Subtask 1.

Test Steps:

1. Navigate to the main task view.
2. Select the subtask "Develop Frontend."
3. Set a dependency on "Design User Interface."
4. Verify that the system displays the dependency relationship between the subtasks correctly.
5. Attempt to mark "Develop Frontend" as complete before "Design User Interface" is marked complete.
6. Verify that the system prevents marking "Develop Frontend" as complete until its dependency is met.

Postconditions:

* The dependency relationship remains established.

Expected Result: The system allows setting the dependency, and prevents marking the dependent subtask complete before the dependency is fulfilled.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive, 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:

## Task Management Tool: Test Cases for Task Dependencies

Here are 5 test cases designed to verify the "Task Dependencies" feature of the Task Management Tool:

Test Case 1: Verify that users can set dependencies between tasks

Field	Value
Test Case ID	TC_DEP_01
Test Case Title	Verify that a user can set dependencies between tasks
Test Case Description	This test case verifies the functionality of setting dependencies between tasks.
Test Suite	Feature Testing - Task Dependencies
Test Priority	High
Preconditions	- A user account is logged in.
	- Two tasks have been created.
Test Data	- Task 1: "Write Project Proposal"

	- Task 2: "Secure Funding"
Test Steps	1. Navigate to the task details for "Write Project Proposal".
	2. Locate and access the task dependency section.
	3. Select "Secure Funding" from the available tasks as a dependency.
	4. Save the changes to the task details.
Postconditions	- The dependency between "Write Project Proposal" and "Secure Funding" is displayed correctly.
Expected Result	The system successfully creates the dependency relationship between the two tasks.
Severity	Critical
Priority	High
Type of Testing	Functional Testing
Test Case Approach	Positive

Test Case 2: Verify that the system displays dependencies in the task details

Field	Value
Test Case ID	TC_DEP_02
Test Case Title	Verify that task dependencies are displayed in the task details
Test Case Description	This test case verifies that the system correctly displays pre-set dependencies associated with a particular task
Test Suite	Feature Testing - Task Dependencies
Test Priority	High
Preconditions	- A user account is logged in.
	- Task A is marked as dependent on Task B (created in previous test case).
Test Data	- Task A: "Write Project Proposal"
	- Task B: "Secure Funding"

Test Steps	1. Navigate to the task details for "Write Project Proposal".
	2. Verify that "Secure Funding" is listed as a dependency.
Postconditions	- The dependency relationship is displayed in the task details.
Expected Result	The system correctly displays "Secure Funding" as the prerequisite task for "Write Project Proposal"
Severity	Critical
Priority	High
Type of Testing	Functional Testing
Test Case Approach	Positive

Test Case 3: Verify that tasks cannot be marked as complete if dependent tasks are not completed

Field	Value
Test Case ID	TC_DEP_03
Test Case Title	Verify that a task cannot be marked complete if its dependent tasks are not completed
Test Case Description	This test case verifies the system's logic for preventing tasks from being marked as complete unless their dependent tasks are fulfilled.
Test Suite	Feature Testing - Task Dependencies
Test Priority	Medium
Preconditions	- A user account is logged in.
	- "Write Project Proposal" and "Secure Funding" tasks created and linked as in previous tests.
Test Data	- "Write Project Proposal" is 50% complete.
Test Steps	1. Navigate to the task details for "Write Project Proposal".
	2. Attempt to mark the task as "Complete".
Postconditions	- The user is prevented from marking the task "Complete".
Expected Result	An alert or warning message is displayed to the user, indicating that the dependent task

"Secure Funding" needs to be completed first. |
Severity	Major
Priority	Medium
Type of Testing	Functional Testing
Test Case Approach	Positive

Test Case 4: Verify that users can update or remove dependencies as needed

Field	Value
Test Case ID	TC_DEP_04
Test Case Title	Verify that a user can update or remove dependencies
Test Case Description	This test case verifies the ability to modify existing dependency relationships.
Test Suite	Feature Testing - Task Dependencies
Test Priority	Medium
Preconditions	- A user account is logged in.
	- The dependency between "Write Project Proposal" and "Secure Funding" exists from previous tests.
Test Data	- Another task "Prepare Design" needs to be dependent on "Write Project Proposal".
Test Steps	1. Navigate to the task details for "Write Project Proposal".
	2. Update the dependency to include "Prepare Design" as well.
	3. Remove the existing dependency with "Secure Funding".
Postconditions	- The dependency is updated to reflect the new relationships.
Expected Result	The system successfully updates the task dependency and removes the old link accurately.
Severity	Major
Priority	Medium

Type of Testing	Functional Testing
Test Case Approach	Positive

Test Case 5: Verify that the system provides alerts or warnings if a dependent task is not completed

Field	Value
Test Case ID	TC_DEP_05
Test Case Title	Verify that alerts are given when a dependent task is not completed
Test Case Description	This test case verifies that the system provides proper notification when a dependent task remains unfinished.
Test Suite	Feature Testing - Task Dependencies
Test Priority	Medium
Preconditions	- A user account is logged in.
	- "Write Project Proposal" and "Design Mockups" tasks are linked, with "Design Mockups" dependent on "Write Project Proposal".
Test Data	- "Write Project Proposal" is 100% complete.
Test Steps	1. Navigate to the task details for "Design Mockups".
	2. Observe any notifications or warnings regarding the completion status of "Write Project Proposal".
Postconditions	- The system has indicated the status of the dependent task.
Expected Result	The system displays a notification or warning that "Write Project Proposal" must be completed before "Design Mockups" can be marked complete.
Severity	Minor
Priority	Medium

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

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 five test cases for the "Task Comments and Collaboration" feature of the Task Management Tool, following the provided template:

Test Case 1

* **Test Case ID:** TC_TaskComments_01
* **Test Case Title:** Verify that users can add text comments to tasks.
* **Test Case Description:** This test case verifies the ability for a user to add a basic text comment to a task and ensure it is properly displayed.
* **Test Suite:** Task Management Tool
* **Test Priority:** High
* **Preconditions:**
    * A user has logged into the system.
    * A task exists in the system.
* **Test Data:** Sample text comment (e.g., "Meeting scheduled for tomorrow at 10 AM")
* **Test Steps:**
   1. Navigate to the task page.
   2. Click on the "Add Comment" button.
   3. Enter the sample text comment in the input field.
   4. Click on the "Post Comment" button.
   5. Verify that the added comment is displayed below the task description.
* **Postconditions:** Comment remains visible on the task page.
* **Expected Result:** The comment is successfully added to the task and is visible to other users with access to the task.
* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TC_TaskComments_02

* **Test Case Title:** Verify that users can add attachments to comments.

* **Test Case Description:** This test case verifies the ability for a user to attach a file to a comment and ensure it is properly displayed.

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* **Preconditions:**

    * A user has logged into the system.

    * A task exists in the system.

    * A sample file (e.g., .docx, .pdf, .jpg) is available on the user's computer.

* **Test Data:** Sample file

* **Test Steps:**

   1. Navigate to the task page.

   2. Click on the "Add Comment" button.

   3. Click on the "Attach File" button.

   4. Select the sample file from the user's computer.

   5. Click on "Post Comment".

   6. Verify that the uploaded file is displayed as an attachment alongside the comment.

* **Postconditions:** File remains attached to the comment.

* **Expected Result:** The file is successfully attached to the comment and is visible to other users with access to the task.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TC_TaskComments_03

* **Test Case Title:** Verify that users can reply to comments in a task.

* **Test Case Description:** This test case verifies the ability for a user to create a reply to an existing comment on a task.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* **Preconditions:**

    * A user has logged into the system.

    * A task exists with at least one comment.

* **Test Data:** Sample reply text ("Understood, I will complete this by Friday.")

* **Test Steps:**

   1. Navigate to the task page.

   2. Locate an existing comment.

   3. Click the "Reply" button below the comment.

   4. Enter the sample reply text in the input field.

   5. Click the "Post Reply" button.

   6. Verify that the reply is displayed as a nested comment below the original comment.

* **Postconditions:** Reply remains visible.

* **Expected Result:** The reply is successfully posted as a nested comment under the original comment, clearly indicating the reply relationship.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TC_TaskComments_04

* **Test Case Title:** Verify that users can edit their own comments.

* **Test Case Description:** This test case verifies a user's ability to modify their previously added comment.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* **Preconditions:**

    * A user has logged into the system.

    * A task exists with at least one comment added by the user.

* **Test Data:**   Original comment text, modified comment text

* **Test Steps:**

   1. Navigate to the task page.

   2. Locate a comment added by the current user.

   3. Click the "Edit" button next to the comment.

   4. Modify the comment text.

   5. Click the "Save Changes" button.

   6. Verify that the comment text has been updated.

* **Postconditions:** Edited comment displays.

* **Expected Result:** The user's own comment is successfully edited, reflecting the changes made.

* **Severity:** Minor

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

Test Case 5

* **Test Case ID:** TC_TaskComments_05
* **Test Case Title:** Verify that new comments are displayed chronologically.
* **Test Case Description:** This test case verifies that new comments are added to the task interface in reverse chronological order (most recent first).
* **Test Suite:** Task Management Tool
* **Test Priority:** Medium
* **Preconditions:**
    * A user has logged into the system.
    * A task exists in the system.
* **Test Data:**
    * Three sample comments with sequential timestamps (e.g., Comment 1 at 10:00 AM, Comment 2 at 10:30 AM, Comment 3 at 11:00 AM)
* **Test Steps:**
   1. Navigate to the task page.
   2. Add Comment 1.
   3. Add Comment 2.
   4. Add Comment 3.
   5. Verify that Comment 3 is displayed first, followed by Comment 2, then Comment 1.
* **Postconditions:** Comments remain displayed.
* **Expected Result:** The comments are displayed in the order they were added, with the most recent comment appearing at the top.

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:

## Task Management Tool: File Attachment Test Cases

User Story: As a user, I want to attach files to tasks so that all relevant information is easily accessible.

Test Cases

Test Case ID: TC_ATTACHMENT_001

Test Case Title: Verify that users can attach multiple files to a task.

Test Case Description: This test case aims to verify if the system allows users to attach more than one file to a single task. It will involve creating a task and adding different file types in multiple instances.

Test Suite: File Attachment

Test Priority: High

Preconditions:

* A user account is logged in.

Test Data:

* 3 different files (e.g., document.docx, image.jpg, spreadsheet.xlsx)

Test Steps:

1. Navigate to a new task creation screen.
2. Create a new task with a title and description.
3. Attach the document.docx file to the task.
4. Attach the image.jpg file to the task.
5. Attach the spreadsheet.xlsx file to the task.
6. Verify that all three files are successfully attached to the task.

Postconditions: Verify that attachments can be viewed and downloaded

Expected Result: The system should successfully allow attachment of multiple files to the task, displaying them in the assigned task view.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_ATTACHMENT_002

Test Case Title: Verify that attachments can include various file types.

Test Case Description: This test case will verify the system's ability to handle various file types for attachments, covering common formats like documents, images, and links.

Test Suite: File Attachment

Test Priority: Medium

Preconditions:

* A user account is logged in.

Test Data:

* A document (.doc), an image (.jpg), a compressed file (.zip), a spreadsheet (.xls), and a link to a website.

Test Steps:

1. Navigate to a new task creation screen.

2. Create a new task with a title and description.

3. Attempt to attach each of the provided file types to the task.

Postconditions: Verify that attached files can be viewed and downloaded.

Expected Result: The system should successfully attach all the provided file types to the task.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_ATTACHMENT_003

Test Case Title: Verify that users can view and download attachments

Test Case Description:

This test case aims to verify if users can successfully view and download attached files to a task.

Test Suite: File Attachment

Test Priority: High

Preconditions:

* A user account is logged in

* A task with attached files exists.

Test Data:

* The task with attached files from a previous test case.

Test Steps:

1. Navigate to the task containing the attachments.

2. Click on the view icon for each attached file.

3. Verify that the file content is displayed correctly.

4. Click the download icon for each attached file.

5. Confirm the file is downloaded to the local machine.

Postconditions: Verify that files remain accessible and download successfully.

Expected Result: The system should allow users to view and download all attached files correctly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_ATTACHMENT_004

Test Case Title: Verify that the system limits the size of attachments to prevent data overload.

Test Case Description: This test case will simulate attaching large files to ensure the system enforces its file size limit.

Test Suite: File Attachment

Test Priority: Medium

Preconditions:

* A user account is logged in.

* Identify the maximum allowed file size for attachments from system documentation.

Test Data:

* A file larger than the permitted maximum size (e.g., a high-resolution video file).

Test Steps:

1. Attempt to attach the large file to a task.

2. Observe the system's response. Verify if an error message is displayed.

Postconditions: System should prevent overwhelming data storage

Expected Result: The system should either display an error message indicating that the file size exceeds the limit or prevent the attachment altogether.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case ID: TC_ATTACHMENT_005

Test Case Title: Verify that users can remove attachments if needed.

Test Case Description: This test case will test the functionality of removing attachments from a task.

Test Suite: File Attachment

Test Priority: Medium

Preconditions:

* A user account is logged in.

* A task with attached files exists.

Test Data:

* The task with attached files from a previous test case.

Test Steps:

1. Navigate to the task containing the attachments.

2. Locate the remove icon for one of the attached files.

3. Click on the remove icon.

4. Confirm that the selected file is no longer attached to the task.

Postconditions Verify that removing an attachment successfully deletes it from the task

Expected Result: The system should successfully remove the selected attachment from the task.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

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:

## Test Cases for Recurring Tasks Feature

Test Case ID: TCT-RTC-001

Test Case Title: Verify that tasks can be set to recur daily.

Test Case Description: This test case verifies the functionality of setting a task to recur daily and ensures that the system generates instances accurately.

Test Suite: Recurring Task Functionality

Test Priority: Medium

Preconditions:

* User is logged into the Task Management Tool.
* User has the necessary permissions to create and manage tasks.

Test Data:

* Task title: "Daily Report"

* Task description: "Summarize daily progress on the project."

* Task deadline: Tomorrow

* Recurrence: Daily

Test Steps:

1. Navigate to the task creation page.

2. Fill in the required task details as specified in the Test Data.

3. Select "Daily" as the recurrence frequency.

4. Set the recurrence to start immediately and save the task.

5. Verify that a new instance of the task is automatically generated for each subsequent day.

Postconditions:

* Recurring daily task instances are visible in the task list.

Expected Result: The system successfully creates and schedules recurring daily tasks as specified by the user.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TCT-RTC-002

Test Case Title: Verify that tasks can be set to recur weekly.

Test Case Description: This test case verifies the functionality of setting a task to recur weekly and ensures that the system generates instances accurately.

Test Suite: Recurring Task Functionality

Test Priority: Medium

Preconditions:

* User is logged into the Task Management Tool.

* User has the necessary permissions to create and manage tasks.

Test Data:

* Task title: "Weekly Meeting"

* Task description: "Discuss project progress and assign tasks."

* Task deadline: Next Wednesday

* Recurrence: Weekly on Wednesdays

Test Steps:

1. Navigate to the task creation page.

2. Fill in the required task details as specified in the Test Data.

3. Select "Weekly" as the recurrence frequency.

4. Set the recurrence to start next Wednesday and save the task.

5. Verify that a new instance of the task is automatically generated every Wednesday until the recurrence is canceled.

Postconditions:

* Recurring weekly task instances are visible in the task list.

Expected Result: The system successfully creates and schedules recurring weekly tasks as specified by the user.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TCT-RTC-003

Test Case Title: Verify that tasks can be set to recur monthly.

Test Case Description: This test case verifies the functionality of setting a task to recur monthly and ensures that the system generates instances accurately.

Test Suite: Recurring Task Functionality

Test Priority: Medium

Preconditions:

* User is logged into the Task Management Tool.
* User has the necessary permissions to create and manage tasks.

Test Data:

* Task title: "Monthly Report"
* Task description: "Generate a detailed report on project performance."
* Task deadline: 15th of next month
* Recurrence: Monthly on the 15th

Test Steps:

1. Navigate to the task creation page.
2. Fill in the required task details as specified in the Test Data.
3. Select "Monthly" as the recurrence frequency.
4. Set the recurrence to start on the 15th of next month and save the task.
5. Verify that a new instance of the task is automatically generated on the 15th of each month until the recurrence is canceled.

Postconditions:
* Recurring monthly task instances are visible in the task list.

Expected Result: The system successfully creates and schedules recurring monthly tasks as specified by the user.

Severity: Major
Priority: High
Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TCT-RTC-004
Test Case Title: Verify that users can modify the recurrence pattern of a recurring task.
Test Case Description: This test case verifies the ability of users to change the recurrence setting of an existing recurring task.

Test Suite: Recurring Task Management
Test Priority: Medium

Preconditions:

* A recurring task is already created and scheduled.

  * User is logged into the Task Management Tool.

  * User has the necessary permissions to edit tasks.

Test Data:

* Existing recurring task with "Daily" recurrence.

Test Steps:

1. Navigate to the task details page of the existing recurring task.

2. Modify the recurrence setting to "Weekly" on Mondays.

3. Save the changes to the task.

4. Verify that the system updates the recurrence pattern for the task accordingly and generates new instances based on the modified schedule.

Postconditions:

* Modified recurring task instances are visible in the task list.

Expected Result: The user can successfully modify the recurrence pattern of an existing recurring task without disrupting existing instances.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TCT-RTC-005

Test Case Title: Verify that users can cancel the recurrence of a task.

Test Case Description: This test case verifies the functionality of canceling the recurrence for an existing recurring task.

Test Suite: Recurring Task Management

Test Priority: Medium

Preconditions:

* A recurring task is already created and scheduled.
  * User is logged into the Task Management Tool.
  * User has the necessary permissions to edit tasks.

Test Data:

* Existing recurring task with "Daily" recurrence.

Test Steps:

1. Navigate to the task details page of the recurring task.
2. Cancel the recurrence setting for the task.
3. Save the changes to the task.
4. Verify that the system stops generating future instances of the recurring task.

Postconditions:

* No new instances of the recurring task are generated.

Expected Result: The user can successfully cancel the recurrence of a task, preventing the creation of future instances.

Severity: Major

Priority: High

Type of Testing: Functional Testing

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:

## Test Cases for Task Management Tool - Task Tags and Labels Feature

Test Case ID: TC-TM-001

Test Case Title: Verify that users can add tags and labels to tasks.

Test Case Description: This test case verifies the functionality of adding tags to tasks, ensuring users can input and associate relevant tags with their tasks.

Test Suite: Task Management Feature

Test Priority: High

Test Data: A new task with a predefined description.

Preconditions:

* A logged-in user has access to the task list.

* The system is functioning properly.

Test Steps:

1. Navigate to the task list page.

2. Click on the "New Task" button and input a task description.

3. Locate and click the "Add Tag" or "Add Label" button.

4. Input a new tag (e.g., "ProjectX") and click "Save" or "Add."

5. Verify that the newly added tag is displayed next to the task description.

Postconditions:

* The new tag is saved in the task database.

* The task is updated with the added tag.

Expected Result: The task is successfully tagged with the specified tag.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-TM-002

Test Case Title: Verify that tags can be customized and edited.

Test Case Description: This test case verifies the ability to modify existing tags, ensuring flexibility and accuracy in task categorization.

Test Suite: Task Management Feature

Test Priority: Medium

Test Data: A task with previously added tags.

Preconditions:

* A logged-in user has access to the task list.

* A task with existing tags is present.

* The system is functioning properly.

Test Steps:

1. Navigate to the task list page and select the task with existing tags.

2. Identify an existing tag and click the "Edit" or "Modify" option associated with it.

3. Change the tag name to a new value (e.g., from "ProjectX" to "ProjectY").

4. Click "Save" or "Update" to apply the changes.

5. Verify that the tag name is updated in the task list.

Postconditions:

* The tag is updated in the task database.

* The task is reflected with the new tag name.

Expected Result: The tag is successfully customized and the task displays the edited tag name.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

(Continue creating similar test cases for the remaining acceptance criteria. Refer to the Template provided for structure and ensure each test case is thorough and comprehensive.)

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:

## Test Cases for Task Status Tracking Feature

Here are 5 detailed test cases for the "Task Status Tracking" feature of the Task Management Tool:

Test Case 1:

* **Test Case ID:** TC_TASK_STATUS_001
* **Test Case Title:** Verify that users can update task status from "To Do" to "In Progress"
* **Test Case Description:** This test case verifies the functionality of changing a task's status from "To Do" to "In Progress". It will involve creating a new task, setting its initial status to "To Do", and subsequently updating its status to "In Progress".
* **Test Suite:** Task Management - Status Tracking
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the application
    * User has permission to create and modify tasks
* **Test Data:**
    * Task description: "Write marketing plan"
    * Initial task status: "To Do"
* **Test Steps:**
    1. Navigate to the task list view.
    2. Create a new task titled "Write marketing plan" and set its status to "To Do".
    3. Open the newly created task.
    4. Change the task status from "To Do" to "In Progress".
    5. Save the changes.
* **Postconditions:**
    * The task's status in the task list view reflects "In Progress".
    * The task details page displays "In Progress" as the current status.
* **Expected Result:** The task status is successfully updated from "To Do" to "In Progress".

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TC_TASK_STATUS_002

* **Test Case Title:** Verify that task status is prominently displayed in the task list.

* **Test Case Description:** This test case verifies that the task status is clearly visible and easily identifiable within the task list view. It will involve examining the display of different task statuses (e.g., To Do, In Progress, Completed) in the list.

* **Test Suite:** Task Management - Status Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the application

    * Several tasks with varying statuses are created

* **Test Data:**

    * Tasks with different statuses ("To Do", "In Progress", "Completed")

* **Test Steps:**

    1. Navigate to the task list view.

    2. Observe the display of each task's status.

    3. Manually check if the status is visually distinct (e.g., different color or icon).

* **Postconditions:** None

* **Expected Result:** Task statuses are clearly displayed and distinguishable within the task list.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_TASK_STATUS_003

* **Test Case Title:** Verify that the system calculates and displays the progress percentage of tasks.

* **Test Case Description:** This test case evaluates the accuracy of the system's progress percentage calculation for tasks. It will involve creating tasks, manually updating progress, and verifying that the displayed percentage reflects the inputted progress.

* **Test Suite:** Task Management - Status Tracking

* **Test Priority:** High

* **Preconditions:**

    * User is logged into the application

* **Test Data:**

    * A task with a defined description and estimated completion percentage

* **Test Steps:**

    1. Create a new task with a clear description and 100% estimated completion.

    2. Mark the task as 50% complete.

    3. Verify that the displayed progress percentage is 50%.

    4. Repeat steps 2 and 3 with different progress percentages.

* **Postconditions:** None

* **Expected Result:** The system accurately calculates and displays the progress percentage for the task based on the entered progress.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TC_TASK_STATUS_004
* **Test Case Title:** Verify that notifications are sent for task status changes.
* **Test Case Description:** This test case verifies that the system sends notifications to users when the status of a task is updated. It will involve creating a new task, setting up notification preferences, and simulating status changes to trigger notifications.
* **Test Suite:** Task Management - Notifications
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the application with notification preferences set
* **Test Data:**
    * Task description: "Design website mockup"
    * User email and notification settings
* **Test Steps:**
    1. Create a new task "Design website mockup" and set its initial status to "To Do".
    2. Assign the task to another user.
    3. Simulate a status change from "To Do" to "In Progress" for the task.
    4. Verify that the assigned user receives a notification about the status change.
* **Postconditions:** None
* **Expected Result:** The system successfully sends a notification to the assigned user when the task's status is updated.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing, User Acceptance Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_TASK_STATUS_005
* **Test Case Title:** Verify that task status can be filtered and sorted.
* **Test Case Description:** This test case evaluates the functionality of filtering and sorting tasks by their status. It will involve creating tasks with varying statuses and verifying that the system accurately filters and sorts them accordingly.
* **Test Suite:** Task Management - Status Tracking
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the application
    * Several tasks with different statuses are created
* **Test Data:**
    * Tasks with various statuses (To Do, In Progress, Completed)
* **Test Steps:**
    1. Navigate to the task list view.
    2. Apply a filter to display only completed tasks.
    3. Verify that only completed tasks are displayed.
    4. Apply a filter to display only "in progress" tasks.
    5. Verify that only tasks with the ?In Progress? status are displayed.
    6. Sort the tasks by completion date, ascending order.
    7. Verify that tasks are displayed in chronological order of completion date.
* **Postconditions:** None
* **Expected Result:**
    * The system accurately filters and displays tasks based on selected statuses.

    * The system sorts tasks correctly based on the chosen criteria (e.g., completion date).

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases provide a starting point for comprehensively testing the task status tracking feature. Adapt and expand them as needed based on the specific implementation details and requirements of the Task Management Tool.

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 Management Tool: Task Filtering and Sorting

Test Case ID: TC_001

Test Case Title: Verify that users can filter tasks by due date.

Test Case Description: This test case verifies the functionality of filtering tasks by due date, ensuring that tasks are displayed according to the selected date range.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* At least three tasks with different due dates exist in the system.

* The user is logged in.

Test Data:

* Task 1: Due date - 2023-10-27

* Task 2: Due date - 2023-11-05

* Task 3: Due date - 2023-12-10

* Filter dates: 2023-10-20 to 2023-11-05

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the task list view.

3. Select the "Due Date" filter option.

4. Choose the date range "2023-10-20 to 2023-11-05".

5. Verify that only Task 1 and Task 2 are displayed in the task list.

Postconditions:

* The filter is removed from the task list view.

Expected Result:

The system correctly filters tasks based on the selected due date range.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_002

Test Case Title: Verify that users can sort tasks by priority.

Test Case Description: This test case verifies the functionality of sorting tasks by priority, ensuring that tasks are displayed in ascending or descending order of their assigned priority.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* At least three tasks with different priority levels exist in the system.

* The user is logged in.

Test Data:

* Task 1: Priority - High

* Task 2: Priority - Medium

* Task 3: Priority - Low

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the task list view.

3. Click on the "Priority" column header to sort tasks.

4. Verify that tasks are displayed in ascending order of priority (Low to High).

5. Click on the "Priority" column header again to sort tasks in descending order.

6. Verify that tasks are displayed in descending order of priority (High to Low).

Postconditions:

* The task list reverts to unsorted order.

Expected Result:

The system correctly sorts tasks by priority in both ascending and descending order.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_003

Test Case Title: Verify that users can filter tasks by assignee.

Test Case Description: This test case verifies the functionality of filtering tasks based on the assigned user.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* At least three tasks are assigned to different users.

* The user is logged in.

Test Data:

* Task 1: Assigned to User A

* Task 2: Assigned to User B

* Task 3: Assigned to User C

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the task list view.

3. Select the "Assignee" filter option.

4. Choose "User A" from the dropdown list.

5. Verify that only Task 1 is displayed in the task list.

6. Repeat steps 3-5 with different assigned users to verify filtering functionality.

Postconditions:

* The filter is removed from the task list view.

Expected Result:

The system correctly filters tasks based on the selected assignee.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_004

Test Case Title: Verify that users can filter tasks by project.

Test Case Description: This test case verifies the functionality of filtering tasks based on the associated project.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* At least three tasks belonging to different projects exist in the system.

* The user is logged in.

Test Data:

* Task 1: Project - Project A

* Task 2: Project - Project B

* Task 3: Project - Project C

Test Steps:

1. Log in to Task Management Tool.

2. Navigate to the task list view.

3. Select the "Project" filter option.

4. Choose "Project A" from the dropdown list.

5. Verify that only Task 1 is displayed in the task list.

6. Repeat steps 3-5 with different projects to verify filtering functionality.

Postconditions:

* The filter is removed from the task list view.

Expected Result:

The system correctly filters tasks based on the selected project.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_005

Test Case Title: Verify that users can save and apply custom filter and sort settings.

Test Case Description: This test case verifies the functionality of saving and applying custom filter and sort settings, allowing users to personalize their task view.

Test Suite: User Interface Testing

Test Priority: Medium

Preconditions:

* The user is logged in.

* At least five tasks exist in the system with varying due dates, priorities, and assignments.

Test Data:

* Custom filter settings: Due date range "2023-10-25 to 2023-11-15", priority "High".

* Custom sort settings: Sort by "Due Date" in descending order.

Test Steps:

1. Log in to Task Management Tool.

2. Navigate to the task list view.

3. Apply desired filters and sort settings (e.g., due date range, priority, sort order).

4. Use the "Save Settings" option to save the custom filter and sort configuration.

5. Log out of the system.

6. Log back into the system.

7. Navigate back to the task list view.

8. Verify that the previously saved filter and sort settings are automatically applied.

Postconditions:

* Saved custom settings are applied on subsequent logins.

Expected Result:

The system correctly saves and applies custom filter and sort settings, restoring the personalized task view.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

These test cases provide a comprehensive framework for evaluating the task filtering and sorting feature in the Task Management Tool, ensuring its functionality, usability, and adherence to user expectations.

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:

## Test Cases for Time Tracking Feature

Task Management Tool

### User Story: As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Test Case ID: TC_TM001

Test Case Title: Verify that users can start and stop a timer for each task.

Test Case Description: This test case verifies the functionality of starting and stopping a timer for individual tasks. It will ensure that the timer accurately records the time elapsed and that users can easily pause and resume tracking.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

* A user is logged in to the Task Management Tool.
* A new task is created.

Test Data: No test data needed

Test Steps:

1. Navigate to the task view for the newly created task.
2. Click on the "Start Timer" button.
3. Perform work on the task for a predetermined period.
4. Click on the "Stop Timer" button.

5. Verify that the total time elapsed is accurately reflected in the task details.

6. Repeat steps 2-5 multiple times with varying durations.

Postconditions:

* Timer is stopped for the task.

* Time tracking data is correctly displayed in the task details.

Expected Result: The timer should start and stop correctly, accurately recording the time spent on the task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TM002

Test Case Title: Verify that the system records and displays total time spent on tasks

Test Case Description: This test case ensures that the system accurately captures and presents the total time spent on a task, regardless of the number of start and stop cycles.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

* A user is logged in to the Task Management Tool.

* A task is created with a specific name.

* The user starts and stops the timer for the task multiple times with varying durations.

Test Data: No test data needed

Test Steps:

1. Start the timer for the task and perform work for a set period.

2. Stop the timer and resume work, starting the timer again for another period.

3. Repeat steps 1 and 2 multiple times with different work durations.

4. Stop the timer for the last time.

5. View the task details and verify that the total time spent is accurately displayed.

Postconditions:

* The task timer is stopped.

* The time tracking data is correctly displayed in the task details.

Expected Result: The system should accurately aggregate all time entries for the task, displaying the total time spent without any discrepancies.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TM003

Test Case Title: Verify that users can manually enter or adjust time entries

Test Case Description: This test case verifies the functionality of manually entering or modifying time entries for tasks.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* A user is logged in to the Task Management Tool.
* A task is created with a specific name.

Test Data: No test data needed

Test Steps:

1. Navigate to the task view for the target task.
2. Click on the "Time Tracking" section.
3. Manually enter a specific time duration for the task.
4. Modify the entered time duration and save the changes.
5. Verify that the modified time is reflected in the task details.

Postconditions:

* Time tracking data is updated with the new entry.
*

Expected Result: The system should allow users to accurately input and edit time entries for tasks. Manual adjustments should reflect in the task's total time spent.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive, Negative

Test Case ID: TC_TM004

Test Case Title: Verify that time tracking data is stored and can be exported.

Test Case Description: This test case confirms that time tracking data is persistently stored and can be exported to a file format for further analysis.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* A user is logged in to the Task Management Tool.
* Multiple tasks have been created with recorded time entries.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Time Tracking" section of the application.
2. Select an export format (e.g., CSV, Excel)
3. Initiate the export process.
4. Verify that a file containing the stored time tracking data is generated.
5. Open the exported file and confirm that the data is accurate and organized.

Postconditions:

* A file containing the time tracking data is downloaded.

* The data in the downloaded file is consistent with the system's stored information.

Expected Result: The system should successfully export time tracking data in a selected format, allowing for external analysis and reporting.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TM005

Test Case Title: Verify that the system provides reports on time spent per task or project

Test Case Description: This test case verifies the functionality of generating reports that showcase time spent on individual tasks or projects.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* A user is logged in to the Task Management Tool.

* Multiple tasks have been created and time entries have been recorded.

* Tasks are assigned to different projects.

Test Data: No test data needed

Test Steps:

1. Access the "Reports" section of the application.

2. Select a report type: "Time Spent per Task" or "Time Spent per Project".

3. Choose a date range for the report.

4. Generate the report.

5. Review the report and verify that the data is accurate and presented in a clear and understandable format.

Postconditions:

* A report is generated and downloaded.

* The report data accurately reflects the time spent on tasks or projects within the chosen timeframe.

Expected Result: The system should generate comprehensive reports that provide insights into time spent on tasks or projects, enabling efficient resource allocation and performance analysis.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Non-Functional Testing

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:

## Test Cases for Task Management Tool - Task Templates Feature

User Story Title: As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

**Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority**
TC-1	Verify that users can create task templates	This test case verifies the ability to create new task templates with predefined task structures, descriptions, assignees, and due dates.	Task Management	High
TC-2	Verify that users can apply existing templates to new tasks	This test case ensures that users can successfully apply existing task templates to newly created tasks, inheriting the template's predefined information.	Task Management	High
TC-3	Verify that users can edit existing task templates	This test case confirms that users can modify the parameters of existing task templates, such as task structure, descriptions, assignees, and due dates.	Task Management	High
TC-4	Verify that users can delete existing task templates	This test case validates the functionality to remove unwanted or outdated task templates.	Task Management	High
TC-5	Verify that the system displays a list of available task templates	This test case ensures that the system presents a clear and comprehensive list of available task templates for user selection.	Task Management	Medium

Table Breakdown:

TC-1: Verify that users can create task templates

* **Preconditions:**
    * The user is logged in to the task management tool.

   * The user has permission to create task templates.

* **Test Data:**
   * Descriptive task title (e.g., "Project Planning Template")
   * Task descriptions
   * Assigned user details
   * Defined due date

* **Test Steps:**
   1. Navigate to the task template creation section.
   2. Enter desired title, description, assignee, and due date for the new template.
   3. Define the task structure within the template.
   4. Save the newly created task template.

* **Postconditions:**
   * The system displays a success message confirming template creation.
   * The created template appears in the list of available templates.

* **Expected Result:** The task template is successfully created and saved in the system.
* **Severity**: Major
* **Priority**: High
* **Type of Testing**: Functional Testing

Let me know if you'd like me to elaborate on any of the test cases or create more comprehensive documentation following the provided template.

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:

## Task Management Tool Test Cases - Calendar Integration

Test Case ID: TC-CAL-001

Test Case Title: Verify that tasks can be synced with external calendars.

Test Case Description: This test case verifies the functionality of syncing tasks with popular external calendars like Google Calendar and Outlook.

Test Suite: Calendar Integration Suite

Test Priority: High

Preconditions:

* User has an active account in the Task Management Tool.
* User has an existing account on Google Calendar or Outlook.

Test Data:

* Sample task with a deadline.
* User's Google Calendar or Outlook account credentials.

Test Steps:

1. Log in to the Task Management Tool.
2. Navigate to the task view and select a sample task.
3. Click on the "Sync to Calendar" button and choose Google Calendar or Outlook from the list.
4. Authorize the Task Management Tool to access the user's Google Calendar or Outlook account.
5. Verify that the task and its deadline appear in the selected calendar.

Postconditions:

* Verify synchronization settings are saved for the selected calendar.

Expected Result: The task with its deadline should be successfully synced to the specified external calendar.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC-CAL-002

Test Case Title: Verify that the system displays task deadlines in the calendar view.

Test Case Description: This test case checks if the system correctly displays task deadlines within the calendar interface.

Test Suite: Calendar Integration Suite

Test Priority: High

Preconditions:

* Same as Test Case TC-CAL-001.

Test Data:

Same as Test Case TC-CAL-001 and additional sample tasks with varied deadlines.

Test Steps:

1. Log in to the Task Management Tool.
2. Select the calendar integration view.
3. Observe the displayed calendar events.
4. Verify that each task event includes its corresponding deadline.
5. Verify that deadlines are displayed accurately in the correct time format.

Postconditions:

* No specific postconditions.

Expected Result: Task deadlines should be accurately displayed as events in the calendar view.

Severity: Major

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC-CAL-003

Test Case Title: Verify that users can set reminders for tasks via the calendar.

Test Case Description: This test case evaluates the functionality of setting reminders for tasks directly within the calendar interface.

Test Suite: Calendar Integration Suite

Test Priority: High

Preconditions:

* Same as Test Case TC-CAL-001

* Notification settings are enabled for the user.

Test Data:

* Sample task with a specific deadline.

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the task view and select a sample task.

3. Click on the "Sync to Calendar" button and choose a calendar.

4. Authorize access, if prompted.

5. On the chosen calendar view, locate the task event.

6. Set a reminder for the task task event through the calendar's reminder functionality.

7. Verify that the reminder is displayed in the calendar and notification settings for the task.

Postconditions:

* Verify reminder settings are saved for the task.

Expected Result: The system should successfully set a reminder for the task via the selected calendar.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, System Testing, Functional Testing

Test Case ID: TC-CAL-004

Test Case Title: Verify that calendar events update automatically when tasks are changed.

Test Case Description: This test case checks if changes made to tasks in the Task Management Tool are reflected automatically in the synced calendar.

Test Suite: Calendar Integration Suite

Test Priority: High

Preconditions:

* Sample task synced with a calendar (as in TC-CAL-001)

* User has editing permissions for the task and calendar.

Test Data:

* Sample task synced with a calendar

* Changes to the task (e.g., updated deadline, modified description)

Test Steps:

1. Log in to the Task Management Tool and navigate to the task view.

2. Select the sample task and make a specific change (update the deadline).

3. Wait for the system to process the change.

4. Access the synced calendar and locate the corresponding task event.

5. Verify that the calendar event reflects the updated changes made to the task.

Postconditions:

* Verify that the synchronization is complete, and the changes are reflected.

Expected Result: Changes made to tasks in the Task Management Tool should automatically update the corresponding calendar events.

Severity: High

Priority: High

Type of Testing: Integration Testing, System Testing, Functional Testing

Test Case ID: TC-CAL-005

Test Case Title: Verify that users can manage calendar sync settings within the app.

Test Case Description: This test case verifies the user's ability to manage calendar sync settings, including selecting sync calendars and disabling syncing.

Test Suite: Calendar Integration Suite

Test Priority: Medium

Preconditions:

* User has a calendar synced as per TC-CAL-001.

Test Data:

* Additional calendar accounts (e.g., if supported).

Test Steps:

1. Log in to the Task Management Tool and navigate to the settings or profile section.
2. Locate the calendar sync settings.
3. Select a synced calendar and un-sync it.
4. Verify that the un-synced calendar is no longer listed.
5. If applicable, select an additional calendar from a list of available options and sync it.

Postconditions:

* No specific postconditions.

Expected Result: Users should be able to easily add, remove, and manage calendars that are synced with their task management tool.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface (UI) Testing

---

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:

## Task Management Tool: Test Cases - Task Duplication

Here are 5 detailed test cases for the "Task Duplication" feature of the Task Management Tool:

Test Case 1

* **Test Case ID:** TC_001
* **Test Case Title:** Verify that new tasks can be duplicated.
* **Test Case Description:** This test case verifies the functionality of duplicating newly created tasks. It ensures that a user can duplicate a task and the duplicated task appears in the task list with all original details.
* **Test Suite:** User Interface - Task Management
* **Test Priority:** High
* **Preconditions:**
    * The user is logged into the Task Management Tool.
    * The user has the necessary permissions to create and duplicate tasks.
* **Test Data:**
    * New task details: description, assignee, deadline, category
* **Test Steps:**
    1. Navigate to the task list view.
    2. Create a new task using the provided test data.
    3. Locate the newly created task in the task list.
    4. Click on the "Duplicate" button associated with the task.
    5. Verify that a new task with the same details as the original task appears in the task list.
* **Postconditions:**
    * The original task remains unchanged.
* **Expected Result:** The duplicate task is created successfully with all original details maintained.
* **Severity:** High

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

Test Case 2

* **Test Case ID:** TC_002
* **Test Case Title:** Verify that existing tasks can be duplicated.
* **Test Case Description:** This test case verifies the functionality of duplicating existing tasks that have already been added to the task list.
* **Test Suite:** User Interface - Task Management
* **Test Priority:** High
* **Preconditions:**
    * The user is logged into the Task Management Tool.
    * The user has the necessary permissions to create and duplicate tasks.
    * An existing task with details is present in the task list.
* **Test Data:**
    * An existing task with description, assignee, deadline, category
* **Test Steps:**
    1. Navigate to the task list view.
    2. Locate the existing task with predetermined details.
    3. Click on the "Duplicate" button associated with the task.
    4. Verify that a new task with the same details as the original task appears in the task list.
* **Postconditions:**
    * The original task remains unchanged.
* **Expected Result:** The duplicate task is created successfully, retaining all original details.
* **Severity:** High

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

Test Case 3

* **Test Case ID:** TC_003
* **Test Case Title:** Verify that duplicate tasks retain all original data.
* **Test Case Description:** This test case confirms that all data elements (description, assignees, deadlines, categories) are correctly copied when a task is duplicated.
* **Test Suite:** Data Validation - Task Duplication
* **Test Priority:** High
* **Preconditions:**
    * The user is logged into the Task Management Tool.
    * The user has the necessary permissions to create and duplicate tasks.
    * An existing task with specific details is present in the task list.
* **Test Data:**
    * An existing task with complex details: detailed description, multiple assignees, specific deadline, unique category.
* **Test Steps:**
    1. Navigate to the task list view.
    2. Locate the task with the complex details.
    3. Click on the "Duplicate" button associated with the task.
    4. Review the duplicate task's details, ensuring all information is identical to the original.
        * Compare description, assignees, deadline, and category.
* **Postconditions:**
    * The original task remains unchanged.

* **Expected Result:** The duplicate task reflects all original data elements accurately.
* **Severity:** High
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TC_004
* **Test Case Title:** Verify modifying a duplicated task.
* **Test Case Description:** This test case verifies a user can modify a duplicated task after its creation.
* **Test Suite:** User Interface - Task Management
* **Test Priority:** Medium
* **Preconditions:**
    * The user is logged into the Task Management Tool.
    * The user has the necessary permissions to create, duplicate, and modify tasks.
* **Test Data:**
    * Existing task with specific details
    * Modifications: Change the deadline, category, description
* **Test Steps:**
    1. Navigate to the task list view.
    2. Duplicate the existing task.
    3. Locate the newly created duplicate task.
    4. Modify the deadline, category, and description of the duplicate task.
    5. Verify that the original task's details remain unchanged.
* **Postconditions:**
    * The original task remains unchanged.

* **Expected Result:** Modifications are saved successfully for the duplicated task, leaving the original task unaffected.

* **Severity:** Medium
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC_005
* **Test Case Title:** Verify error handling for task duplication.
* **Test Case Description:** This test case explores error scenarios when attempting to duplicate a task.
* **Test Suite:** Error Handling - Task Duplication
* **Test Priority:** Medium
* **Preconditions:**
    * The user is logged into the Task Management Tool.
    * The user has the necessary permissions to create and duplicate tasks. Existing task is present in the task list.
* **Test Data:**
    * An existing task (can be a sample task with default values)
    * Invalid input: attempting to duplicate a task that doesn't exist.
* **Test Steps:**

1. Navigate to the task list view.
2. Attempt to duplicate a non-existent task (one that?s not in the task list).
3. Observe the error message displayed.

* **Postconditions:**

* **Expected Result:** An appropriate error message is displayed to the user, informing them that the selected task doesn't exist.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

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:

## Test Cases for Task Management Tool - Task History Feature

Here are 5 detailed test cases for the "Task History" feature of the Task Management Tool:

Test Case 1:

* **Test Case ID:** TC_TASK_HISTORY_001
* **Test Case Title:** Verify that users can view a detailed history of changes to tasks.
* **Test Case Description:** This test case verifies if the system displays a comprehensive history of changes made to a task, including modifications to details like description, assignee, due date, etc.
* **Test Suite:** Task Management Feature Suite
* **Test Priority:** High
* **Preconditions:**

   * A valid user account is created.
   * A task is created with at least three editable fields (e.g., description, assignee, due date).
   * The user has the permission to view task history.
* **Test Data:**
   * The task created with initial values.
   * Modified values for at least three task fields.
* **Test Steps:**
   1. Log in as a valid user with access to task history.
   2. Navigate to the created task.
   3. Open the "Task History" section.
   4. Verify that the history displays past modifications made to the task fields.
   5. Ensure timestamps are included for each change.
* **Postconditions:**
   * The task remains in its modified state.
* **Expected Result:** The task history section accurately displays a chronological list of modifications made to the task fields, along with timestamps.

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

Test Case 2:

* **Test Case ID:** TC_TASK_HISTORY_002
* **Test Case Title:** Verify that task history includes comments and other relevant activity.
* **Test Case Description:** This test case verifies if comments and other relevant activities related to a task

are recorded in the history.

* **Test Suite:** Task Management Feature Suite

* **Test Priority:** High

* **Preconditions:**

    * A valid user account is created.

    * A task is created with a description.

    * The user has the permission to view task history.

* **Test Data:**

    * Task with initial description

    * Added comments to the task

    * Task status updates (e.g., from "To Do" to "In Progress")

* **Test Steps:**

    1. Log in as a valid user.

    2. Navigate to the created task.

    3. Add comments to the task.

    4. Update the task status.

    5. Open the "Task History" section.

    6. Verify that comments and status changes are recorded in the history.

* **Postconditions:**

    * The changes to the task remain.

* **Expected Result:** Task history accurately reflects all comments and status updates made to the task.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_TASK_HISTORY_003

* **Test Case Title:** Verify that users can filter the history by type of change or date.

* **Test Case Description:** This test case verifies users can filter the task history by the type of change (e.g., description, assignee, status) or by date range.

* **Test Suite:** Task Management Feature Suite

* **Test Priority:** Medium

* **Preconditions:**

    * A valid user account is created.

    * Tasks have been created with various changes made at different timestamps.

    * The user has the permission to view task history.

* **Test Data:**

    * Tasks with multiple modifications, including changes to description, deadlines, assignee, and status.

* **Test Steps:**

    1. Log in as a valid user.

    2. Navigate to a task with a history of changes.

    3. Access the "Filter" options within the task history section.

    4. Filter by specific change types (e.g., "Description," "Assignee," "Status").

    5. Verify that the filtered history displays only relevant changes.

    6. Filter by date range (e.g., "Last week," "Today").

    7. Verify that the filtered history displays changes within the selected date range.

* **Postconditions:**

    * No data loss or corruption occurs after filtering.

* **Expected Result:** The task history can be accurately filtered by both change type and date range.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TC_TASK_HISTORY_004

* **Test Case Title:** Verify that the system provides timestamps for all changes.

* **Test Case Description:** This test case verifies that timestamps are consistently associated with all recorded changes in the task history.

* **Test Suite:** Task Management Feature Suite

* **Test Priority:** Medium

* **Preconditions:**

    * A valid user account is created.

    * A task is created.

    * The user has the permission to view task history.

* **Test Data:**

    * Tasks with modifications made at different times.

* **Test Steps:**

    1. Log in as a valid user.

    2. Navigate to a task with recent modifications.

    3. Open the "Task History" section.

    4. Observe the timestamp displayed next to each recorded change.

    5. Verify that the timestamps accurately reflect the order and timing of changes.

* **Postconditions:**

    * The task status remains unchanged.

* **Expected Result:** Each entry in the task history is accompanied by a precise timestamp indicating when

the change occurred.

* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_TASK_HISTORY_005
* **Test Case Title:** Verify that task history is accessible from the task details page.
* **Test Case Description:** This test case verifies the task history section is accessible directly from the task view page without requiring additional navigation.
* **Test Suite:** Task Management Feature Suite
* **Test Priority:** Low
* **Preconditions:**
    * A valid user account is created.
    * A task is created.
    * The user has the permission to view task history.
* **Test Data:**
    * A recently created task.
* **Test Steps:**
    1. Log in as a valid user.
    2. Navigate to the task details page.
    3. Locate and click on the "Task History" section.
    4. Verify that the history section displays correctly.

* **Postconditions:**

    * No changes are made to the task itself.

* **Expected Result:** The "Task History" section is easily accessible directly from the task details page, providing a seamless user experience.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **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:

## Test Cases for Task Notifications Feature

Project Context: Task Management Tool

User Story: As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

### Test Case 1

Test Case ID: TCN-001

Test Case Title: Verify that notifications are sent for task updates when assignee is changed.

Test Case Description: This test case verifies that the system sends a notification to both the previous and new assignee when the assignee of a task is changed.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* A user is logged into the system.

* A task exists with an assigned assignee.

Test Data:

* Task ID: T-001

* Original Assignee: User A

* New Assignee: User B

Test Steps:

1. User A logs in and assigns a task (T-001) to User B.

2. Ensure that User A receives a notification about the task assignment being changed.

3. Ensure that User B receives a notification about the task assignment being updated.

Postconditions:

* Task T-001 is now assigned to User B.

* Notifications are recorded in the user activity log.

Expected Result:

Both User A and User B receive notifications about the task assignee change.

Severity: Critical

Priority: High

Type of Testing: Functional testing

Test Case Approach: Positive

### Test Case 2

Test Case ID: TCN-002

Test Case Title: Verify that notifications can be customized for different notification types.

Test Case Description: This test case verifies the user's ability to customize notification preferences for task updates, comments, mentions, and other relevant activities.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* A user is logged into the system.
* Different notification channels (email, push, in-app) are available.

Test Data:

* User preferences for different notification types.

Test Steps:

1. Navigate to the user's notification settings.
2. Set preferences for receiving task update notifications via email for the user.
3. Set preferences for receiving comment notifications via push notifications for the user.
4. Set preferences for disabling mention notifications for email.

Postconditions:

* Updated notification settings are saved.

Expected Result:

The user's notification preferences are correctly configured, and notifications are sent accordingly based on the selected channels.

Severity: Major

Priority: Medium

Type of Testing: Functional testing

Test Case Approach: Positive

### Test Case 3

Test Case ID: TCN-003

Test Case Title: Verify that notifications provide detailed information about the activity.

Test Case Description: This test case verifies that notifications include all relevant information about the notified activity, enabling users to understand the context.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* A user is logged into the system.

* A task exists with a comment and an attachment.

Test Data:

* Task ID: T-002

* Comment content: "Important feedback added to the task"

* Attachment file: report.pdf

Test Steps:

1. User A leaves a comment on task T-002, attaching a document.

2. User B, who is assigned to the task, receives a push notification.

Postconditions:

* Notification is recorded in the user activity log.

Expected Result:

The notification received by User B includes:

* Task title: T-002

* Comment content: "Important feedback added to the task"

* Indication that an attachment (report.pdf) is included.

* Link to the task details page.

Severity: Major

Priority: Medium

Type of Testing: Functional testing

Test Case Approach: Positive

### Test Case 4

Test Case ID: TCN-004

Test Case Title: Verify that the system logs notification history for future reference.

Test Case Description: This test case verifies that the system maintains a log of all sent notifications, allowing users to track past activity.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

* A user is logged into the system.

* Notifications have been generated (task updates, comments, etc.).

Test Data:

* User ID: User A

Test Steps:

1. Navigate to the user's notification history page.

2. Verify that a list of past notifications is displayed.

Postconditions:

* Notification history is displayed and accessible.

Expected Result: A chronological list of past notifications received by User A is displayed, clearly indicating the type of activity, associated task, timestamp, and other relevant details.

Severity: Minor

Priority: Low

Type of Testing: Functional testing

Test Case Approach: Positive

### Test Case 5

Test Case ID: TCN-005

Test Case Title: Verify that no notifications are sent when no relevant activity occurs.

Test Case Description: This test case verifies that the system does not generate unnecessary notifications for tasks lacking updates.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

* A task exists with no recent updates or changes.

Test Data:

* Task ID: T-003

Test Steps:

1. Perform specific actions that *should not* trigger notifications (e.g., viewing the task details, marking it as completed).

2. Verify that the user does not receive any notifications related to these actions.

Postconditions:

* No new notifications are generated.

Expected Result:

No notifications are sent to users for actions that do not constitute relevant activity updates on the task.

Severity: Minor

Priority: Low

Type of Testing: Functional testing

Test Case Approach: Negative

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 Management Tool Test Cases - Task Archiving Feature

Here are 5 detailed test cases for the "Task Archiving" feature of the Task Management Tool, following the provided template:

Test Case ID: TMT-ARCH-001

Test Case Title: Verify that completed tasks can be archived.

Test Case Description: This test case verifies the functionality of archiving completed tasks from the active task list.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* The user has access to create and complete tasks.

* A task exists in the active task list that is marked as completed.

Test Data:

* One completed task with a descriptive title and due date.

Test Steps:

1. Navigate to the active task list view.

2. Identify a completed task in the list.

3. Click on the "Archive" button (or equivalent) associated with the completed task.

4. Verify that the task is successfully removed from the active task list.

5. Navigate to the archived task list view.

6. Verify that the archived task is present in the archived task list.

Postconditions:

* Archived task remains accessible in the archived task list.

Expected Result: The completed task is successfully archived and moved to the designated archived task section.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMT-ARCH-002

Test Case Title: Verify that inactive tasks can be archived.

Test Case Description: This test case verifies the functionality of archiving inactive tasks from the active task list.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* The user has access to create and modify tasks.

* A task exists in the active task list that is past its due date.

Test Data:

* One task with a past due date and no status updates.

Test Steps:

1. Navigate to the active task list view.

2. Identify an inactive task (past due) in the list.

3. Click on the "Archive" button (or equivalent) associated with the inactive task.

4. Verify that the task is successfully removed from the active task list.

5. Navigate to the archived task list view.

6. Verify that the archived task is present in the archived task list.

Postconditions:

* Archived task remains accessible in the archived task list.

Expected Result: The inactive task is successfully archived and moved to the designated archived task section.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMT-ARCH-003

Test Case Title: Verify that archived tasks can be searched.

Test Case Description: This test case verifies the ability to search for archived tasks by specific keywords or criteria.

Test Suite: User Interface Testing

Test Priority: Medium

Preconditions:

 * A user is logged into the Task Management Tool.

 * At least two archived tasks exist with different titles and descriptions.

Test Data:

* Existing archived tasks with distinct

titles and descriptions.

Test Steps:

1. Navigate to the archived task list view.

2. Enter a specific keyword related to a task title or description into the search bar.

3. Click the search button.

4. Verify that the tasks matching the search criteria are displayed in the results.

5. Input a completely unrelated keyword into the search bar.

6. Click the search button and verify that no matching tasks are displayed.

Postconditions:

* Search functionality remains effective after the session.

Expected Result: The system accurately retrieves archived tasks based on the search keywords used.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive/Negative

Test Case ID: TMT-ARCH-004

Test Case Title: Verify that archived tasks can be restored.

Test Case Description: This test case verifies the ability to restore archived tasks back to the active task list.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* At least one task exists in the archived task list.

Test Data:

* One archived task that requires restoration.

Test Steps:

1. Navigate to the archived task list view.

2. Identify the archived task to be restored.

3. Click on the "Restore" button (or equivalent) associated with the task.

4. Verify that the task is successfully removed from the archived task list.
5. Navigate to the active task list view.
6. Verify that the restored task is present in the active task list.

Postconditions:
* Restored task retains all previous details.

Expected Result: The archived task is successfully restored to its original state in the active task list, retaining all original data and history.

Severity: Major
Priority: High
Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMT-ARCH-005
Test Case Title: Verify that archived tasks retain original details.
Test Case Description: This test case verifies that archived tasks retain all original fields (deadline, description, attachments, etc.) upon restoration.

Test Suite: Data Integrity Testing
Test Priority: High

Preconditions:
* The user has created a task with a description, attachments, and a due date.

* The task has been successfully archived.

Test Data:

* A task with detailed description, attachments (images/documents), and a specific due date.

Test Steps:

1. Navigate to the archived task list.
2. Select the task from step 1, and review its details (description, attachments, due date).
3. Restore the task to the active task list.
4. Directly from the active task listview, re-open the restored task.
5. Verify that the description, attachments, and due date from the archived state are unchanged.

Postconditions:

* No data corruption is observed

Expected Result: All original details of the archived task remain intact after restoration, including descriptions, attachments, due dates, and any other relevant information.

Severity: Critical (Data corruption)

Priority: High

Type of Testing: Functional Testing, Data Integrity Testing

Test Case Approach: Positive

These test cases cover the key functionalities of the task archiving feature and aim to ensure a high-quality user experience. Remember that this is not an exhaustive list, and additional test cases may be necessary depending on the specific implementation details of the Task Management Tool.

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:

## Test Cases for Guest Access Feature

Project Name: Task Management Tool

Feature Name: Guest Access

User Story Title: As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

### Test Case 1

Test Case ID: TC_GUEST_001

Test Case Title: Verify that users can invite external collaborators as guests.

Test Case Description: This test case verifies the functionality of inviting external collaborators as guests with limited access to specific tasks.

Test Suite: Guest Access

Test Priority: High

Preconditions:

* A logged-in user is accessing the task management platform.
* There is a task or project available for sharing.

Test Data:

* A valid external email address for the guest collaborator.

Test Steps:

1. Navigate to the task or project to be shared.

2. Click on the "Share" or "Invite Collaborator" option.

3. Enter the external email address in the designated field.

4. Select the "Guest" access role.

5. Click "Invite".

Postconditions:

* The guest user receives an invitation email with a link to accept the invitation.

Expected Result: The system successfully sends the invitation email to the guest collaborator, granting them limited access permissions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 2

Test Case ID: TC_GUEST_002

Test Case Title: Verify that guests can view tasks they are invited to.

Test Case Description: This test case verifies that guest users can successfully view the details of tasks they have been invited to collaborate on.

Test Suite: Guest Access

Test Priority: High

Preconditions:

* A guest user has accepted the invitation to a task.

Test Data:

* Existing task with details (description, attachments, due date)

Test Steps:

1. The guest user logs into the platform using the provided credentials.

2. Navigate to the designated task.

3. Verify that the guest user can view all task details.

Postconditions:

* The guest user can access and view all information about the task.

Expected Result: The guest user can see all the information related to the task, including description, attachments, and due date.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 3

Test Case ID: TC_GUEST_003

Test Case Title: Verify that guests cannot access restricted areas of the platform.

Test Case Description: This test case verifies that guest users do not have access to sensitive areas of the platform, ensuring data security.

Test Suite: Guest Access

Test Priority: High

Preconditions:

* A guest user is logged into the platform.

Test Data:

* Sensitive areas of the platform (e.g., user profile settings, admin dashboard).

Test Steps:

1. Log into the platform as a guest user.
2. Attempt to navigate to restricted areas of the platform (e.g., user profile settings, admin dashboard).
3. Observe the outcome.

Postconditions:

* Access denied message appears.

Expected Result:

The guest user is prevented from accessing restricted areas and sees an appropriate error message indicating they lack permissions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Negative

### Test Case 4

Test Case ID: TC_GUEST_004

Test Case Title: Verify that guests can collaborate on tasks assigned to them.

Test Case Description: This test case verifies the guest user can add comments and updates to a task they are invited to.

Test Suite: Guest Access

Test Priority: Medium

Preconditions:

* A guest user has accepted an invitation to a task.

Test Data:

* A clear task with instructions for guest collaboration

* A sample comment or update

Test Steps:

1. The guest user logs into the platform.

2. Navigate to the task they were invited to.

3. Compose a comment or update related to the task.

4. Submit the comment or update.

Postconditions:

* The comment or update appears on the task page visible to all collaborators.

Expected Result: The guest user's comment or update is successfully added to the task, demonstrating collaborative functionality.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 5

Test Case ID: TC_GUEST_005

Test Case Title: Verify that the system clearly distinguishes between guest and full access roles.

Test Case Description: This test case ensures the visual differentiation of user roles (guest vs. full access) within the platform.

Test Suite: Guest Access

Test Priority: Medium

Preconditions:

* A logged-in user with full access and a guest user are both viewing the same task.

Test Data:

* Existing task with user list and role indicators

Test Steps:

1. The full access user and the guest user both log into the platform.

2. Both users access the same task.

3. Observe the user list for the task.

Postconditions:

* The user list clearly distinguishes between the full access user and the guest user based on visual cues or role indicators.

Expected Result: The platform visually differentiates between full access users and guest users within the task interface, preventing confusion and misunderstanding.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

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:

## Task Management Tool: Task Import/Export Test Cases

Test Case ID: TM-IMP-001

Test Case Title: Verify that tasks can be imported from a CSV file.

Test Case Description:

This test case aims to verify the functionality of importing tasks from a CSV file. It will ensure that the system correctly parses the CSV data and creates corresponding tasks in the system. This test will also verify the mapping of fields during import.

Test Suite: Task Import

Test Priority: High

Preconditions:

* A valid CSV file containing task data is available.

* User account with sufficient permissions to import tasks is logged in.

* The system is configured to support CSV imports.

Test Data:

* CSV file containing task information (Title, Description, Due Date, Assignees, Project)

Test Steps:

1. Navigate to the Task Import section within the application.

2. Upload the prepared CSV file.

3. Review the field mapping provided by the system.

4. Confirm the accuracy of the field mapping and finalize the import process.

5. Verify that the imported tasks are displayed in the task list.

6. Check the details of the imported tasks to ensure data accuracy.

Postconditions:

* The imported CSV file is processed.

* Imported tasks are visible in the application.

Expected Result: All tasks from the CSV file are successfully imported into the system with accurate field values.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TM-IMP-002

Test Case Title: Verify that tasks can be imported from Excel file.

Test Case Description:

This test case verifies the functionality of importing tasks from an Excel file. It ensures the system can appropriately read the Excel data and create corresponding tasks.

Test Suite: Task Import

Test Priority: High

Preconditions:

* A valid Excel file containing task data is available.
* User account with sufficient permissions to import tasks is logged in.
* The system is configured to support Excel imports.

Test Data:

* A well-formatted Excel file containing task data (Title, Description, Due Date, Assignees, Project)

Test Steps:

1. Access the Task Import section within the application.
2. Upload the prepared Excel file.
3. Review the field mapping provided by the system.
4. Confirm the accuracy of the field mapping and proceed with the import.
5. Examine the newly imported tasks in the task list.
6. Verify the details of the imported tasks for data accuracy.

Postconditions:

* The imported Excel file is processed.

* Imported tasks are visible in the application.

Expected Result: All tasks from the Excel file are successfully imported into the system with accurate field values.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TM-IMP-003

Test Case Title: Verify that task import handles empty fields correctly.

Test Case Description: This test case evaluates how the system handles tasks with missing or empty fields during import.

Test Suite: Task Import

Test Priority: Medium

Preconditions:

* A CSV file containing tasks with some empty fields is prepared.

* A user account with sufficient permissions to import tasks is logged in.

* The system is configured to support CSV imports.

Test Data:

* CSV file containing data with empty fields in Title, Description, Due Date, Assignee

Test Steps:

1. Navigate to the Task Import section within the application.

2. Upload the CSV file containing incomplete data.

3. Review the field mapping and ensure empty fields are correctly identified.

4. Finalize the import process.

Postconditions:

* The imported CSV file is processed.

Expected Result: Tasks with empty fields will be imported, and the empty fields will be handled appropriately. This could be by leaving the field blank in the system or using default values.

Severity: Major

Priority: Medium

Type of Testing: System Testing, Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TM-EXP-001

Test Case Title: Verify that tasks can be exported to a CSV file.

Test Case Description:

This test case verifies the user?s ability to export selected or all tasks from the system into a CSV file format.

Test Suite: Task Export

Test Priority: High

Preconditions:

* A user account with sufficient permissions to export tasks is logged in.

* The system has at least one task available.

Test Data:

* Defined selection criteria for exported tasks (e.g., all tasks, tasks in a specific project, tasks assigned to a specific user).

Test Steps:

1. Navigate to the Task Export section within the application.

2. Select the desired tasks to export (all tasks or filtered by criteria).

3. Choose "CSV" as the export format.

4.Initiate the export process.

Postconditions:

* A CSV file containing the exported task data is downloaded.

Expected Result: A CSV file containing the selected tasks? data (Title, Description, Due Date, Assignees, Project) is successfully downloaded.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TM-LOG-001

Test Case Title: Verify that import/export activities are logged.

Test Case Description:

This test case verifies that the system logs details about import and export activities for audit trails and troubleshooting.

Test Suite: Task Management Logging

Test Priority: Medium

Preconditions:

* A user account with permission to import/export tasks is logged in.

* An import or export activity is performed.

Test Data:

* Task data for import/export (CSV or Excel format).

Test Steps:

1. Perform an import or export activity (import a CSV file or export tasks).

2. Navigate to the system log or activity tracking section.

3. Locate the log entry related to the performed import/export action.

4. Examine the log entry for relevant details (date/time, user, file name, number of tasks imported/exported, type of import/export).

Postconditions:

* The log entry related to the import/export activity is visible.

Expected Result: A log entry is created that contains information about the import/export activity, including the timestamp, user who initiated the activity, file name used (if applicable), number of tasks imported or exported, and the type of action performed (import or export).

Severity: Major

Priority: Medium

Type of Testing: System Testing, Integration Testing

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:

## Test Cases for Task Prioritization Feature

User Story: As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

### Test Case 1

Test Case ID: TCT_PR_001

Test Case Title: Verify that users can set task priorities using defined levels.

Test Case Description: This test case verifies the user's ability to assign priority levels (e.g., High, Medium, Low) to tasks upon creation and through task editing.

Test Suite: User Interface - Task Management

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* The system has predefined task priority levels (High, Medium, Low).

Test Data:

* Sample task descriptions

* Data for different priority levels.

Test Steps:

1. Navigate to the Task Creation form.

2. Enter a task description for the new task.

3. Select the desired priority level (High, Medium, or Low) from the dropdown menu.

4. Create the task successfully.

5. Navigate to the task details page.

6. Verify that the selected priority level is displayed correctly for the task.

Postconditions:

* A new task is created with the assigned priority level.

Expected Result: The task is created and displays the selected priority level.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 2

Test Case ID: TCT_PR_002

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

Test Case Description: This test case ensures that tasks can be accurately sorted by their defined priority levels (High, Medium, and Low) using available sorting options within the application.

Test Suite: User Interface - Task Management

Test Priority: High

Preconditions:

* A user is logged into the Task Management Tool.

* At least three tasks with different priority levels exist.

Test Data:

* Existing tasks with different priority levels.

Test Steps:

1. Navigate to the task list view.

2. Select "Priority" from the available sorting options.

3. Verify that tasks are sorted in descending order of priority (High, Medium, Low).

Postconditions:

* The task list is sorted by priority levels.

Expected Result: Tasks are displayed in descending order of priority (High, Medium, Low).

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 3

Test Case ID: TCT_PR_003

Test Case Title: Verify that the system provides visual indicators of task priority.

Test Case Description: This test case checks if the application visually distinguishes tasks based on their assigned priority levels through icons, colors, or other graphical cues.

Test Suite: User Interface - Task Management

Test Priority: Medium

Preconditions:

* A user is logged into the Task Management Tool.
* Tasks with different priority levels exist.

Test Data:

* Existing tasks with different priority levels.

Test Steps:

1. Navigate to the task list view.
2. Observe the visual indicators associated with tasks of different priority levels (e.g., High - red icon, Medium - yellow icon, Low - green icon).
3. Verify that the visual indicators are distinct and clearly distinguish the priority levels.

Postconditions:

* The system displays visual indicators for each task's priority level.

Expected Result:

The interface presents distinct visual cues for each priority level.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 4

Test Case ID: TCT_PR_004

Test Case Title: Verify that users receive notifications for high-priority tasks.

Test Case Description: This test case verifies if users receive timely notifications for high-priority tasks approaching their deadlines or when other relevant events occur. (This may require integrating with a notification system)

Test Suite: User Experience Feedback

Test Priority: High

Preconditions:

* A high-priority task with a near-future deadline exists.
* User notification settings are enabled.

Test Data:

* High-priority task with a set deadline.

Test Steps:

1. Set up a high-priority task with a specific deadline in the near future.

2. Simulate a notification based on the defined deadline (e.g., through a mock notification system).

3. Verify that the user receives a notification alert for the high-priority task.

Postconditions:

* A notification is sent to the user regarding the high-priority task.

Expected Result: A timely notification is delivered to the user regarding the approaching deadline of the high-priority task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 5

Test Case ID: TCT_PR_005

Test Case Title: Verify that priority changes are logged and tracked.

Test Case Description: This test case examines the system's ability to record and display the history of priority changes made to a task.

Test Suite: Audit Trail

Test Priority: Medium

Preconditions:

* A user is logged into the Task Management Tool.

* A task exists.

Test Data:

* Existing task.

Test Steps:

1. Navigate to the task details page.

2. Change the task's priority level.

3. Verify that the task history log shows the change of priority level and the corresponding timestamp.

4. Change the task's priority level again.

5. Verify that the history log displays both changes and their respective timestamps.

Postconditions:

* The task history includes a record of the priority changes.

Expected Result: Both priority changes are logged in the task history with accurate timestamps.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive, Negative

These test cases are structured to cover the defined acceptance criteria for the Task Prioritization feature in the Task Management Tool. Please note that these are just examples, and more comprehensive testing may be required to ensure full functionality and quality.

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:

## Test Cases for Task Management Tool - Task Collaboration Feature

Test Case ID: TCT_001

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 specific team members within the Task Management Tool. It will involve creating a task, selecting a team member for assignment, and confirming successful assignment.

Test Suite: Feature Testing - Task Collaboration

Test Priority: High

Preconditions:

* User is logged in to the Task Management Tool.

* A user has created a project and added additional users to it.

* The "Assign Task" feature is enabled for the project.

* Test accounts need to be set up to ensure functionality across different user roles.

Test Data:

* Task details: Title, Description, Project, Deadline (within a reasonable timeframe)

* Team member list with user login credentials

Test Steps:

1. Navigate to the project containing the task to be assigned.

2. Create a new task with the provided details.

3. Click on the "Assign Task" button.

4. Select the intended team member from the available list.

5. Confirm the assignment.

Postconditions:

* Assigned user receives a notification about the assigned task.

* Assigned task appears in the designated team member's task list.

Expected Result: The task is successfully assigned to the selected team member. The system should provide clear confirmation of the assignment, and the assigned user should receive a notification.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TCT_002

Test Case Title: Verify that users can share updates and communicate through comments or messages.

Test Case Description: This test case verifies the functionality for sharing updates and communicating through comments or messages on assigned tasks within the Task Management Tool. It will involve creating a test task, adding updates and messages, and verifying their visibility to assigned users.

Test Suite: Feature Testing - Task Collaboration

Test Priority: High

Preconditions:

* User is logged in to the Task Management Tool.
* A task exists with assigned team members.

Test Data:

* Test update message with appropriate content.
* Test comments with different message types (informative, requesting clarification, suggestions).

Test Steps:

1. Navigate to the task page.
2. Provide a test update message and post it on the task.
3. Add test comments with varying content and purposes.

Postconditions:

* All updates and comments visible to all assigned users.
* Assigned users can reply to messages

Expected Result: The shared updates and comments are visible to all assigned team members in a chronological order. Team members should be able to reply to messages and contribute to the conversation.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TCT_003

Test Case Title: Verify that task updates and comments are visible to assigned team members only.

Test Case Description: This test case ensures that task updates and comments are restricted to assigned team members and not accessible to unauthorized users. It will involve creating a task, assigning it, and simulating access attempts from different user types.

Test Suite: Feature Testing - Task Collaboration

Test Priority: High

Preconditions:

* User is logged in as an authorized user with access to the project.

* A user is logged in as an unauthorized user with no access to the project.

* A task exists within a project and has been assigned to specific team members.

Test Data:

* A task title and description

* A few test updates and comments

Test Steps:

1. Navigate to the task page as the authorized user.

2. Add test updates and comments to the task.

3. Log out of the authorized user account.

4. Log in as the unauthorized user.

5. Attempt to access the task and view updates/comments.

Postconditions:

* Authorized user can view all updates and comments on the task.

* Unauthorized user cannot view the task updates or comments.

Expected Result: The unauthorized user is not allowed to access task updates or comments.

Severity: Critical

Priority: High

Type of Testing: Security Testing, Access Control Testing

Test Case Approach: Negative

Test Case ID: TCT_004

Test Case Title: Verify that users receive notifications for task updates and comments.

Test Case Description: This test case verifies the functionality of notifications for task updates and comments. It will involve assigning a task, making an update, and confirming that a notification is delivered to the assigned user.

Test Suite: Feature Testing - Task Collaboration

Test Priority: High

Preconditions:

* User is logged in to the Task Management Tool.

* A task exists with assigned team members.

* The notification settings are configured and enabled for receiving updates and comments.

Test Data:

* A task with a test deadline and description

* Test update messages with different message types

Test Steps:

1. As the assigning user, create a new task and assign it to another user.

2. Log out of the assigning user account.

3. Log in as the assigned user.

4. Make a test update on a different task to ensure notifications are triggered.

Postconditions:

* Assigned User receives a notification about the new task assignment.

* Assigned user can view the notification from the assigned task in the notification section.

Expected Result: The assigned user receives a notification about the task assignment. The notification should contain relevant details about the assigned task.

Severity: High

Priority: High

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Test Case ID: TCT_005

Test Case Title: Verify that the system provides a clear view of task collaboration history.

Test Case Description: This test case checks the effectiveness of the Task Management Tool's history view for task collaboration.

It will involve adding updates, comments, and assignments to a task and then verifying the chronological order and clarity of the displayed history.

Test Suite: Feature Testing - Task Collaboration

Test Priority: Medium

Preconditions:

* User is logged in to the Task Management Tool.
* A task exists with assigned team members.

Test Data:

* Varying types of test updates (text, links, files)
* Test comments with different response types (acknowledgement, question, suggestion)
* Multiple task assignments & re-assignments

Test Steps:

1. Access the task in question and add several test updates and comments.

2. Make multiple assignments and re-assignments to different team members.

3. Navigate to the task history section.

4. Review the displayed history for chronological order and clarity.
5. Verify that all updates, comments, and assignments are accurately reflected.

Postconditions:
* Task history is complete and accurate.
* All users with access to the task can view history.

Expected Result: The task history displays a clear and chronological record of all updates, comments, and assignments made to the task. Each entry should be accurately labelled and attributed to the responsible user.

Severity: Major
Priority: Medium
Type of Testing: Functional Testing, Usability Testing, UI Testing
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:

## Task Management Tool: Test Cases for Task Tracking Feature

User Story: As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

### Test Case 1

Test Case ID: TC_TaskTracking_001

Test Case Title: Verify that users can view task statuses accurately.

Test Case Description: This test case verifies that the system correctly displays the status of tasks (e.g., "To Do", "In Progress", "Completed") and that these statuses are updated accordingly as the task progresses.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

* A user is logged in to the system.
* Several tasks with different statuses (To Do, In Progress, Completed) exist in a project.

Test Data:

* A list of tasks with varying statuses.

Test Steps:

1. Navigate to the project where the tasks are located.
2. View the list of tasks.
3. Observe the status displayed for each task and ensure it matches the expected status based on the task details.
4. Update the status of one task to a different status (e.g., change "To Do" to "In Progress").
5. Refresh the task list and verify that the updated status is reflected.

Postconditions:

* No changes to task data besides the status update.

Expected Result: The system accurately displays the status of all tasks, and the status of a modified task is reflected immediately after the update.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive, Exploratory

### Test Case 2

Test Case ID: TC_TaskTracking_002

Test Case Title: Verify that users can track task progress and deadlines.

Test Case Description: This test case checks if the system allows users to monitor the progress of individual tasks and their associated deadlines.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

* A user is logged in to the system.
* A task with a defined deadline exists in a project.

Test Data:

* A task with a specific deadline in the future.

Test Steps:

1. Navigate to the task details page.
2. Verify that the task progress is displayed (e.g., a percentage bar or a graphical representation).
3. Observe the task deadline and ensure it is accurately reflected.
4. Update the task progress manually (e.g., move the progress bar)
5. Observe how the display of task progress and remaining time to deadline changes.

Postconditions:

* No changes to task deadline information.

Expected Result: The system accurately displays task progress and remaining time to deadline, and these values update dynamically as the task progresses.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive, Exploratory

### Test Case 3

Test Case ID: TC_TaskTracking_003

Test Case Title: Verify real-time task updates.

Test Case Description: This test case confirms that task status updates are reflected in real-time across the system.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

* Two users (User A and User B) are logged into the system.
* A task is assigned to both users.

Test Data:

* A task with shared responsibility.

Test Steps:

1. User A completes a portion of the task and updates its status.
2. User B refreshes their view of the task details.
3. Observe if the updated status is immediately visible to User B.

Postconditions:

* No changes to the task data except for the status update initiated by User A.

Expected Result: User B sees the updated status of the task instantly upon refreshing their view, confirming real-time updates.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

### Test Case 4

Test Case ID: TC_TaskTracking_004

Test Case Title: Verify visual indicators for task progress.

Test Case Description: This test case examines the system?s visual cues for representing task progress and ensures they are clear, intuitive, and correctly reflect the task's completion status.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

* A user is logged in to the system.
* A task is assigned with a percentage completion value.

Test Data:

* A task with a specified percentage completion (e.g., 50%).

Test Steps:

1. Navigate to the task details page.
2. Observe the visual indicator representing the task progress (e.g., a progress bar, percentage display).
3. Ensure the visual indicator accurately reflects the set percentage completion of the task.

Postconditions:

* User view of the task status remains unchanged

Expected Result: The system displays a clear visual representation of the task's progress, aligning with the defined percentage completion.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

### Test Case 5

Test Case ID: TC_TaskTracking_005

Test Case Title: Verify handling of invalid task progress updates.

Test Case Description: This test case investigates how the system handles invalid inputs for task progress updates, ensuring data integrity and preventing unintended changes.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

* A user is logged in to the system.
* A task exists in a project.

Test Data:

* Invalid input values for task progress (e.g., a negative percentage, a value exceeding 100%).

Test Steps:

1. Attempt to update the task progress with an invalid input value.
2. Observe the system's response (e.g., an error message, rejection of the update).

Postconditions:

* Task progress remains unchanged.

Expected Result: The system prevents invalid task progress updates, displaying an appropriate error message or rejecting the change to maintain data integrity.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

These are just a few examples, and more test cases can be created depending on the specific functionalities and complexities of the Task Management Tool.

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:

## Test Cases for Task Management Tool - Task Notifications

Test Case ID: TMT-NTF-001

Test Case Title: Verify that users receive notifications for new tasks.

Test Case Description: This test case verifies that a new task assigned to a user triggers a notification.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* A user account is created and logged in.

* Notifications are enabled for the user.

* A new task is created with an assigned user.

Test Data:

* Task details including title, description, due date, and expected notification message.

Test Steps:

1. A user logs in to the Task Management Tool.

2. An administrator creates a new task and assigns it to the logged-in user.

3. The logged-in user should receive a notification about the new task assignment.

4. The notification should include the task title and a brief description.

5. The user clicks on the notification link, which takes them to the corresponding task page.

Postconditions:

* The user is redirected to the task details page.

Expected Result: The user receives a notification for the new task assignment.

Severity: Major

Test Case ID: TMT-NTF-002

Test Case Title: Verify that users receive notifications for task updates.

Test Case Description: This test case verifies that any changes made to an existing task trigger a notification.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* A user account is created and logged in.

* Notifications are enabled for the user.

* A task exists and is assigned to the logged-in user.

Test Data:

* Modified task details including title, description, or due date, and expected notification message.

Test Steps:

1. A user logs in to the Task Management Tool.

2. The user opens an existing task assigned to them.

3. The user makes a change to the task details (e.g., updates the due date).

4. The user should receive a notification about the changes made to the task.

5. The notification should include a brief summary of the changes.

Postconditions:

* The user's task details are updated.

Expected Result: The user receives a notification about the task update.

Test Case ID: TMT-NTF-003

Test Case Title: Verify that users receive notifications for task comments.

Test Case Description: This test case verifies that new comments added to a task trigger a notification.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* A user account is created and logged in.

* Notifications are enabled for the user.

* A task exists and is assigned to the logged-in user.

Test Data:

* New comment text for the task.

Test Steps:

1. A user logs in to the Task Management Tool.

2. The user opens an existing task assigned to them.

3. Another user or the same user adds a new comment to the task.

4. The logged-in user should receive a notification about the new comment.

5. The notification should briefly mention the commenter and the content of the comment.

Postconditions:

* The comment is visible in the task's comment section.

Expected Result: The user receives a notification about the new comment.

Test Case ID: TMT-NTF-004

Test Case Title: Verify that users receive reminders for upcoming deadlines.

Test Case Description: This test case verifies that users receive timely reminders for tasks approaching their deadlines.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* A user account is created and logged in.

* Notifications are enabled for the user.

* A task with a due date is created and assigned to the user.

Test Data:

* Task with a specific due date set in the future.

Test Steps:

1. A user logs in to the Task Management Tool.

2. A task is created with a specified due date, e.g., in 24 hours.

3. Before the due date, the user should receive a reminder notification about the upcoming deadline.

4. The notification should include the task title and the remaining time until the deadline.

5. A timed reminder should be sent again shortly before the actual deadline.

Postconditions:

* The reminder notifications are received timely.

Expected Result: The user receives timely reminder notifications for upcoming task deadlines.

Test Case ID: TMT-NTF-005

Test Case Title: Verify that notification settings can be customized by users.

Test Case Description: This test case verifies that users can adjust their notification preferences, including frequency and types.

Test Suite: User Settings

Test Priority: Medium

Preconditions:

* A user account is created and logged in.

Test Data:

* Different notification settings configurations: push, email, in-app, frequency (daily, weekly).

Test Steps:

1. A user logs in to the Task Management Tool and accesses their profile settings.

2. The user navigates to the "Notification" settings section.

3. The user selects different notification preferences, such as choosing email notifications over push notifications.

4. The user can choose to receive notifications for specific events, like new tasks, updates, or deadlines.

5. The user can adjust the frequency of notifications.

Postconditions:

* The changes to notification settings are saved.

Expected Result: Users can successfully customize their notification settings to their desired preferences.

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:

## Test Cases for Task Categorization Feature

Project Name: Task Management Tool

Feature Name: Task Categorization

User Story: As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

Test Case 1

Test Case ID: TC_TASK_CAT_001

Test Case Title: Verify that users can create new categories or projects.

Test Case Description: This test case verifies the user's ability to create new categories or projects within the task management tool.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the system.

Test Data: "New Project", "Marketing Campaign"

Test Steps:

1. Navigate to the "Projects" or "Categories" section.

2. Click on the "Create New Category/Project" button.

3. Enter the name "New Project" and click "Save".

4. Verify that a new category/project named "New Project" is created and displayed.

5. Repeat steps 2 to 4 with the name "Marketing Campaign".

Postconditions:

* Two new categories/projects ("New Project" and "Marketing Campaign") exist.

* Deletion of the newly created categories/projects isn't necessary for this test.

Expected Result: Two new categories/projects are successfully created and displayed in the system.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_TASK_CAT_002

Test Case Title: Verify that users can assign tasks to different categories or projects.

Test Case Description: This test case verifies that users can assign existing tasks to different categories/projects.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the system.

* At least one task exists.

* At least two categories/projects exist (one newly created and the other existing).

Test Data: Existing task, "New Project" category, Existing category

Test Steps:

1. Open the task details for an existing task.

2. Select the "Assign to" dropdown menu.

3. Choose the newly created "New Project" category.

4. Verify that the task is successfully assigned to the "New Project" category.

5. Repeat steps 1-4, assigning the same task to a different existing category.

Postconditions:

* The task should be effectively reassigned without losing any data.

Expected Result: Tasks can be successfully assigned to different categories/projects.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_TASK_CAT_003

Test Case Title: Verify that users can view tasks grouped by categories or projects.

Test Case Description: This test case ensures the system displays tasks correctly categorized within the respective views.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the system.

* At least two tasks exist, each assigned to a different category/project.

Test Data: Two Tasks

Test Steps:

1. Navigate to the "Task List" view.

2. Verify that tasks are displayed under their corresponding categories/projects.

3. Select a category/project from the filter dropdown.

4. Confirm that only tasks belonging to the selected category/project are displayed.

Postconditions: The filter functionality retains its state after application.

Expected Result: Tasks are accurately displayed grouped by their assigned categories/projects.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC_TASK_CAT_004

Test Case Title: Verify that tasks can be moved between categories or projects.

Test Case Description: This test case checks the ability to move tasks between different categories or projects.

Test Suite: User Interface Testing

Test Priority: Medium

Preconditions:

* User is logged into the system.

* At least two tasks exist, each assigned to a different category/project.

Test Data: Task

Test Steps:

1. Select a task from the task list.

2. Click the "Move to" button or equivalent functionality.

3. Choose a different category/project for the selected task.

4. Verify that the task is successfully moved to the new category/project.

Postconditions: The task's status reflects the change in category/project.

Expected Result: Tasks can be effectively moved and reassigned between existing categories/projects.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC_TASK_CAT_005

Test Case Title: Verify that Category or project information is accurately reflected in task views.

Test Case Description: This test case verifies the accurate display of category/project information alongside task details.

Test Suite: User Interface Testing & Data Validation

Test Priority: Medium

Preconditions:

* User is logged into the system.

* At least one task exists and is assigned to a specific category/project.

Test Data: Task

Test Steps:

1. Open the details view for the assigned task.

2. Verify that the category/project to which the task belongs is clearly displayed.

3. Change the task's category/project through the system interface.

4. Reopen the task details and verify that the updated category/project information is reflected.

Postconditions:

Expected Result: Task details consistently show the accurate and updated category/project assignment.

Severity: Major

Priority: Medium

Type of Testing: Functional & Data Validation

Test Case Approach: Positive & 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:

## Task Management Tool - Task Search Feature Test Cases

Test Case ID: TC_TASK_SEARCH_01

Test Case Title: Verify that users can search for tasks by name.

Test Case Description: This test case will verify that the system correctly retrieves tasks based on a user-entered task name. It will examine both exact matches and partial matches.

Test Suite: Task Functionality

Test Priority: High

Preconditions:

- A user account is logged in.

- At least three tasks with distinct names are present in the user's project or task list.

Test Data:

- Task names: "Meeting with John", "Project Proposal", "Code Review"

- Search term: "Meeting"

Test Steps:

1. Open the Task Management Tool and navigate to the task list.

2. Enter the search term "Meeting" in the search bar.

3. Click the search button.

4. Verify that the task "Meeting with John" is displayed in the search results.

5. Verify that the tasks "Project Proposal" and "Code Review" are not displayed in the search results.

Postconditions:

- The search bar is cleared.

Expected Result:

The search results display the task "Meeting with John" and exclude the other tasks.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_SEARCH_02

Test Case Title: Verify that users can search for tasks by description.

Test Case Description: This test case will verify the system's ability to retrieve tasks based on a description entered by the user.

Test Suite: Task Functionality

Test Priority: High

Preconditions:

- A user account is logged in.
- At least three tasks with unique descriptions are present in the user's project or task list.

Test Data:

- Task descriptions: "Discuss project progress", "Review code for bugs", "Prepare presentation materials"
- Search term: "bugs"

Test Steps:

1. Open the Task Management Tool and navigate to the task list.
2. Enter the search term "bugs" in the search bar.
3. Click the search button.
4. Verify that the task "Review code for bugs" is displayed in the search results.
5. Verify that the tasks "Discuss project progress" and "Prepare presentation materials" are not displayed in the search results.

Postconditions:

- The search bar is cleared.

Expected Result:

The search results display the task "Review code for bugs" and exclude the other tasks.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_SEARCH_03

Test Case Title: Verify that users can filter search results by category.

Test Case Description: This test case verifies the functionality of filtering search results based on task categories.

Test Suite: Task Functionality

Test Priority: High

Preconditions:

- A user account is logged in.
- At least three tasks with different categories are present in the user's project or task list.

Test Data:

- Task categories: "Project A", "Project B", "Personal"
- Filter category: "Project A"

Test Steps:

1. Open the Task Management Tool and navigate to the task list.
2. Enter any search term in the search bar.
3. Click on the "Category" filter dropdown.
4. Select the filter "Project A".
5. Verify that only tasks belonging to the "Project A" category are displayed in the search results.

Postconditions:

- The search bar retains the original search term.

- The "Category" filter dropdown remains open.

Expected Result:

Tasks from categories other than "Project A" are excluded from the search results.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_SEARCH_04

Test Case Title: Verify that users can filter search results by priority.

Test Case Description: This test case verifies the functionality of filtering search results based on task priority.

Test Suite: Task Functionality

Test Priority: High

Preconditions:

- A user account is logged in.

- At least three tasks with different priorities are present in the user's project or task list.

Test Data:

- Task priorities: "High", "Medium", "Low"

- Filter priority: "High"

Test Steps:

1. Open the Task Management Tool and navigate to the task list.

2. Enter any search term in the search bar.

3. Click on the "Priority" filter dropdown.

4. Select the filter "High".

5. Verify that only tasks with a priority of "High" are displayed in the search results.

Postconditions:

- The search bar retains the original search term.

- The "Priority" filter dropdown remains open.

Expected Result:

Tasks with priorities other than "High" are excluded from the search results.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TASK_SEARCH_05

Test Case Title: Verify that users can filter search results by due date.

Test Case Description: This test case verifies the functionality of filtering search results based on task due dates.

Test Suite: Task Functionality

Test Priority: High

Preconditions:

- A user account is logged in.

- At least three tasks with different due dates are present in the user's project or task list (e.g., today, tomorrow, next week).

Test Data:

- Due dates: Today, Tomorrow, Next week.

- Filter due date: "Today"

Test Steps:

1. Open the Task Management Tool and navigate to the task list.

2. Enter any search term in the search bar.

3. Click on the "Due Date" filter dropdown.

4. Select the filter "Today".

5. Verify that only tasks with a due date of today are displayed in the search results.

Postconditions:

- The search bar retains the original search term.

- The "Due Date" filter dropdown remains open.

Expected Result:

Tasks with due dates other than "Today" are excluded from the search results.

Severity: Major

Priority: High

Type of Testing: Functional Testing

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 some test cases for the "Task Collaboration" feature of the Task Management Tool, based on your provided information:

Test Case 1

* **Test Case ID:** TC_TASK_COLLAB_01
* **Test Case Title:** Verify that users can add comments to tasks.
* **Test Case Description:** This test case aims to ensure that users can add text comments to tasks and that these comments are displayed correctly to other collaborators.
* **Test Suite:** Task Management
* **Test Priority:** High
* **Preconditions:**
    * A logged-in user has access to a task.
* **Test Data:** Sample text comment content
* **Test Steps:**
    1. Navigate to the task view.
    2. Click the "Add Comment" button.
    3. Enter the sample text comment in the comment box.
    4. Click the "Post Comment" button.
    5. Verify that the comment is displayed below the task details.
* **Postconditions:**
    * The comment remains visible to other collaborators.
* **Expected Result:** The comment is posted successfully and is visible to other collaborators.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing

Test Case 2

* **Test Case ID:** TC_TASK_COLLAB_02

* **Test Case Title:** Verify that users can attach files to tasks.

* **Test Case Description:** This test case verifies the functionality of attaching files to tasks, ensuring they are correctly displayed and accessible to collaborators.

* **Test Suite:** Task Management

* **Test Priority:** High

* **Preconditions:**

    * A logged-in user has access to a task.

    * A sample file is available on the user's device.

* **Test Data:** Sample file (e.g., document, image, spreadsheet)

* **Test Steps:**

    1. Navigate to the task view.

    2. Click the "Attach File" button.

    3. Select the sample file from the device.

    4. Click the "Upload" button.

    5. Verify that the file is listed under the "Attached Files" section.

* **Postconditions:**

    * The file remains accessible to other collaborators.

* **Expected Result:** The file is successfully uploaded and displayed in the designated area.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case 3

* **Test Case ID:** TC_TASK_COLLAB_03

* **Test Case Title:** Verify that users can share tasks with other users.

* **Test Case Description:** This test case confirms the ability to share tasks with other users and ensures these users receive notifications about the shared task.

* **Test Suite:** Task Management

* **Test Priority:** High

* **Preconditions:**

    * A logged-in user has access to a task.

    * Another user account is available.

* **Test Data:** Email address of the other user

* **Test Steps:**

    1. Navigate to the task view.

    2. Click the "Share Task" button.

    3. Enter the email address of the other user.

    4. Select the desired access level (e.g., View Only, Edit, Collaborate).

    5. Click the "Share" button.

* **Postconditions:**

    * The other user receives a notification about the shared task.

* **Expected Result:** The task is successfully shared with the other user, and they receive appropriate notifications.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case 4

* **Test Case ID:** TC_TASK_COLLAB_04

* **Test Case Title:** Verify that task collaboration features work seamlessly and enhance team communication.
* **Test Case Description:** This test case assesses the overall usability and effectiveness of the collaboration features in facilitating communication within a team.
* **Test Suite:** Task Management
* **Test Priority:** Medium
* **Preconditions:**
    * Two or more user accounts are available.
    * A task is created and accessible to all users.
* **Test Data:** Sample comments, files, and task updates
* **Test Steps:**
    1. Multiple users collaborate on a task by adding comments, attaching files, and completing assigned tasks.
    2. Observe the ease of use and clarity of the collaboration interface.
    3. Evaluate the effectiveness of notifications and reminders.
* **Postconditions:**
    * All users confirm their experience with the collaboration features.
* **Expected Result:** Users report that the collaboration features are intuitive, enhance communication, and contribute to efficient task completion.

Test Case 5

* **Test Case ID:** TC_TASK_COLLAB_05
* **Test Case Title:** Verify handling of invalid user inputs for task sharing.
* **Test Case Description:** This test case checks that the system handles invalid input for task sharing in a user-friendly way.
* **Test Suite:** Task Management
* **Test Priority:** Medium

* **Preconditions:**
    * A logged-in user has access to a task.
    * Mock email addresses are available.
* **Test Data:**
    * Invalid email addresses (e.g., missing "@", incorrect domain)
* **Test Steps:**
    1. Attempt to share the task with an invalid email address.
    2. Observe the feedback provided to the user.
* **Postconditions:** There are no unintended system consequences.
* **Expected Result:**
    * The system displays an error message indicating the invalid email format.

Let me know if you'd like more test cases for a specific scenario or want to explore negative or boundary condition testing further.

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:

Here are 5 detailed test cases for the "Course Creation Tools" feature, following your provided template:

## Test Case 1

Test Case ID: TC-CC-001

Test Case Title: Verify that instructors can create courses using videos.

Test Case Description: This test case aims to validate the functionality of adding video content to a new course. It will involve uploading a sample video file and ensuring it is properly embedded within the course structure.

Test Suite: Course Creation

Test Priority: High

Preconditions:

* An instructor account is logged in.

* A sample video file (MP4 format) is available.

Test Data: Sample video file

Test Steps:

1. Navigate to the course creation page.

2. Click on the "Add Video" button.

3. Select the sample video file from the local machine.

4. Configure video settings (e.g., title, description, duration) if applicable.

5. Save the course as a draft.

Postconditions:

* The video should be listed within the course outline.

Expected Result: The video should be incorporated successfully into the course structure and be viewable within the platform.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TC-CC-002

Test Case Title: Verify that instructors can create courses using presentations.

Test Case Description: This test case aims to validate the functionality of adding presentation materials (e.g., PowerPoint, PDF) to a new course.

Test Suite: Course Creation

Test Priority: High

Preconditions:

* An instructor account is logged in.
* A sample presentation file (PPT or PDF format) is available.

Test Data: Sample presentation file

Test Steps:

1. Navigate to the course creation page.
2. Click on the "Add Presentation" button.
3. Select the sample presentation file from the local machine.
4. Configure presentation settings (e.g., title, description) if applicable.
5. Save the course as a draft.

Postconditions:

* The presentation should be listed within the course outline.

Expected Result: The presentation should be incorporated successfully into the course structure and be viewable within the platform.

Severity: Critical
Priority: High
Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TC-CC-003
Test Case Title: Verify that instructors can create quizzes within a course.

Test Case Description: This test case aims to validate the functionality of adding quizzes and questions to a new course.

Test Suite: Course Creation
Test Priority: High
Preconditions:

* An instructor account is logged in.

Test Data: Sample quiz questions

Test Steps:

1. Navigate to the course creation page.
2. Click on the "Add Quiz" button.
3. Create a new quiz with a title and description.
4. Add multiple-choice questions to the quiz.
5. Save the course as a draft.

Postconditions:

* The quiz should be listed within the course outline.

Expected Result: The quiz should be incorporated successfully into the course structure, and instructors should be able to view and manage the questions added to it.

Severity: Critical
Priority: High
Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TC-CC-004
Test Case Title: Verify that instructors can preview a course before publishing.

Test Case Description: This test case aims to ensure that instructors can review and interact with the course content before making it publicly accessible.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

* An instructor account is logged in.
* A course has been partially created with content and assignments.

Test Data: Course with partially added content

Test Steps:

1. Navigate to the course creation page.
2. Locate the "Preview" button.
3. Click on the "Preview" button to open the course in preview mode.
4. Navigate through the course content, interacting with videos, presentations, and assignments.
5. Close the preview mode.

Postconditions:

* The preview mode should function correctly, allowing instructors to fully experience the course.

Expected Result: The preview should accurately reflect the course structure and content, allowing instructors to identify and rectify any issues before publishing.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: TC-CC-005

Test Case Title: Verify that instructors can edit published courses.

Test Case Description: This test case validates the functionality for instructors to modify published courses, ensuring content can be updated and improvements can be made.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

* An instructor account is logged in.
* A course has been successfully published.

Test Data: Changes to be incorporated into the course (e.g., updated video, revised quiz)

Test Steps:

1. Access the published course from the instructor's dashboard.
2. Locate the "Edit" button or menu option.

3. Click on the "Edit" button to open the course for editing.
4. Make the desired changes to the course content, such as updating a video or revising a quiz question.
5. Save the changes.

Postconditions:

* The published course should reflect the implemented changes.

Expected Result: The platform should allow instructors to edit published courses, ensuring the learning materials remain current and relevant.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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:

## Test Cases for Interactive Lessons Feature

Project Name: Online Learning Platform

Feature Name: Interactive Lessons

User Story Title: As a learner, I want interactive lessons so that I can engage with the content more effectively.

Test Case 1:

Test Case ID: TC_INTERACTIVE_001

Test Case Title: Verify that quizzes are included in lessons.

Test Case Description: This test case verifies the presence and functionality of quizzes within interactive lessons on the platform. It checks if quizzes are integrated seamlessly into the lesson content and if learners can attempt them correctly.

Test Suite: Interactive Lesson Functionality

Test Priority: High

Preconditions:

* User is logged in to the platform.
* User has access to at least one lesson with interactive content.
* The lesson contains a quiz.

Test Data: No test data needed

Test Steps:

1. Navigate to the lesson with the quiz.
2. Locate and click on the "Quiz" section within the lesson.
3. Attempt all questions in the quiz.
4. Submit the quiz.
5. Review the quiz results and ensure feedback is provided for each question.

Postconditions:

* The quiz must be completed successfully.
* Feedback for each question must be displayed.

Expected Result: The learner can access and attempt the quiz within the lesson, receive feedback on their answers, and see their overall quiz score.

Severity: Critical

Priority: High

Test Case 2:

Test Case ID: TC_INTERACTIVE_002

Test Case Title: Verify that polls are integrated seamlessly into lessons.

Test Case Description: This test case aims to ensure that polls are properly implemented within lessons and function as intended. It checks if learners can participate in polls, see real-time results, and that the polls are integrated smoothly into the lesson flow.

Test Suite: Interactive Lesson Functionality

Test Priority: High

Preconditions:

* User is logged in to the platform.

* User has access to a lesson containing a poll.

* The poll is active.

Test Data: No test data needed

Test Steps:

1. Navigate to the lesson containing the poll.

2. Find and click on the "Poll" section within the lesson.

3. Select an option from the poll and submit the vote.

4. Observe the real-time results displayed after voting.

5. Refresh the page to ensure results are consistent.

Postconditions:

* The user's vote is recorded.

* Real-time results are updated accurately.

Expected Result: The learner can participate in the poll, see the results, and the poll seamlessly integrates with the lesson content.

Test Case 3:

Test Case ID: TC_INTERACTIVE_003

Test Case Title: Verify learners can track their progress through interactive activities.

Test Case Description: This test case verifies the functionality of the progress tracking system for interactive elements within lessons. It checks if learners can view completed activities, unfinished activities, and their overall progress through the lesson.

Test Suite: User Progress Tracking

Test Priority: High

Preconditions:

* User is logged in to the platform.

* User has access to a lesson with multiple interactive activities.

* The user has started and completed some of the activities.

Test Data: No test data needed

Test Steps:

1. Navigate to the lesson with multiple interactive activities.

2. Identify and interact with various interactive activities within the lesson (quizzes, polls, simulations).

3. Using the provided progress tracker, review the status of each activity.

4. Verify the system correctly reflects completed and incomplete activities.

5. Observe the overall progress indicator within the lesson.

Postconditions:

* User progress is recorded accurately.

Expected Result: The learner can track their progress through interactive activities within the lesson and see a clear representation of their completion status.

Severity: Major

Priority: High

Test Case 4:

Test Case ID: TC_INTERACTIVE_004

Test Case Title: Verify that interactive elements are accessible on different devices.

Test Case Description: This test case verifies the responsiveness and functionality of interactive lessons across various devices (desktop, mobile, tablet). It checks if the interactive elements function as expected and the user interface adapts appropriately.

Test Suite: Cross-device Compatibility

Test Priority: Medium

Preconditions:

* User has access to multiple devices (desktop, mobile, tablet).

* Internet connectivity is available on all devices.

Test Data: No test data needed

Test Steps:

1. Log in to the platform using the same user account on each device.

2. Navigate to a lesson with interactive elements.

3. Test the functionality of quizzes, polls, simulations, and discussions on each device.

4. Observe the user interface and layout on each device.

5. Attempt to complete an interactive task on each device.

Postconditions:

* All interactive elements function correctly.

Expected Result: The interactive lessons function seamlessly and the user interface adapts to different screen sizes without compromising functionality.

Test Case 5:

Test Case ID: TC_INTERACTIVE_005

Test Case Title: Verify feedback provided after each interactive component is informative and helpful.

Test Case Description: This test case assesses the quality and usefulness of feedback provided to learners after completing interactive elements (quizzes, polls, simulations). It checks if the feedback is constructive, explains incorrect answers, and encourages further learning.

Test Suite: Interactive Lesson Feedback

Test Priority: High

Preconditions:

* User is logged in to the platform.

* User has access to a lesson with detailed feedback mechanisms for interactive elements.

Test Data: No test data needed

Test Steps:

1. Navigate to the lesson with interactive elements.

2. Complete a quiz within the lesson, selecting both correct and incorrect answers.

3. Analyze the feedback provided for each question, both correct and incorrect.

4. Review any additional explanations or resources offered for incorrect answers.

5. Participate in a poll and observe the feedback regarding the results.

Postconditions:

* Learners receive constructive feedback.

Expected Result:

The feedback provided for each interactive component is clear, insightful, and aids the learner in understanding their performance and areas for improvement.

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:

## Test Cases for Personalized Learning Paths Feature

User Story: As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

Test Case ID: PLC-001

Test Case Title: Verify that learners can choose courses and modules based on their interests.

Test Case Description: This test case validates that learners can select courses and modules based on their preferred subjects and interests using the platform's filtering and search functionality.

Test Suite: User Story "Personalized Learning Paths"

Test Priority: High

Preconditions:

* A user account is created on the platform.
* Multiple courses and modules covering diverse subjects are available.

Test Data:

* List of preferred subjects provided by a user (e.g., coding, history, design).

Test Steps:

1. Log in as a user.
2. Navigate to the "Courses" or "Learning Paths" section.
3. Use filter options (e.g., by subject, category) to select courses and modules related to the user's preferred subjects.
4. Verify that only relevant courses and modules appear in the results.
5. Select a course and browse its modules; ensure they align with the user's chosen subject interests.

Postconditions:

* The user remains on the relevant course or module selection page.

Expected Result:

The platform displays only courses and modules relevant to the user's selected subject preferences, allowing for a personalized browsing experience.

Severity: Critical

Type of Testing: Functional Testing, User Acceptance Testing.

Test Case Approach: Positive

(Repeat the template for the remaining test cases, focusing on the remaining acceptance criteria)

Test Case ID: PLC-002

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress.

Test Case ID: PLC-003

Test Case Title: Verify that learners receive recommendations for new courses and modules.

Test Case ID: PLC-004

Test Case Title: Verify that progress tracking is personalized to each learner's path.

Test Case ID: PLC-005

Test Case Title: Verify that learners can adjust their learning paths at any time.

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:

## Test Cases for Online Learning Platform - Progress Tracking Feature

User Story: As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Test Case ID: TCP_PT_001

Test Case Title: Verify that completed activities are displayed in the progress tracker.

Test Case Description: This test case verifies that the progress tracker accurately reflects the learner's completion status for all activities within a course.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

* A learner account is created and logged in.
* A course with multiple activities (lectures, quizzes, assignments) is available.
* The learner has completed various activities within the course.

Test Data: Course with completed activities

Test Steps:

1. Navigate to the course dashboard.
2. Access the progress tracker section.
3. Verify that all completed activities are marked as complete or finished in the progress tracker.
4. Click on individual completed activities to confirm their details are displayed correctly.
5. Compare the progress tracker data with the learner's actual activity completion status.

Postconditions:

* The learner remains logged into the platform.

Expected Result: The progress tracker accurately displays all completed activities with their corresponding completion status.

Severity: Critical
Priority: High
Type of Testing: Functional Testing

Test Case ID: TCP_PT_002
Test Case Title: Verify that the progress tracker displays grades for completed quizzes and assignments.
Test Case Description: This test case verifies that the progress tracker accurately displays the learner's grades for completed quizzes and assignments.

Test Suite: Progress Tracking
Test Priority: High
Preconditions:

* A learner account is created and logged in.
* A course with quizzes and assignments is available.
* The learner has completed quizzes and assignments and received corresponding grades.

Test Data: Course with graded quizzes and assignments

Test Steps:

1. Navigate to the course dashboard.

2. Access the progress tracker section.

3. Verify that completed quizzes and assignments are displayed along with their corresponding grades.

4. Click on individual graded items to verify that the displayed grades match the learner's recorded grades.

5. Cross-reference the progress tracker data with the official grading system.

Postconditions:

* The learner remains logged into the platform.

Expected Result: The progress tracker accurately displays grades for completed quizzes and assignments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case ID: TCP_PT_003

Test Case Title: Verify that achievements are highlighted in the progress tracker.

Test Case Description: This test case verifies that significant milestones or achievements within a course are correctly highlighted in the progress tracker.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

* A learner account is created and logged in.
* A course with clearly defined achievements (completing modules, achieving a certain grade, etc.) is available.
* The learner has achieved at least one milestone in the course.

Test Data: Course with predefined achievements

Test Steps:

1. Navigate to the course dashboard and access the progress tracker section.
2. Verify that completed achievements are visually highlighted (e.g., with a badge, different color).
3. Click on a highlighted achievement to view details about its achievement criteria and progress.
4. Confirm that unlocking a new achievement also generates a corresponding notification.
5. Check for visual cues indicating upcoming achievements (e.g., progress bar).

Postconditions:

* The learner remains logged into the platform.

Expected Result: The progress tracker visually highlights completed achievements with distinctive markers.

Severity: Major
Priority: Medium
Type of Testing: Functional Testing, User Interface Testing

Please note that these are just examples, and you may need to create additional test cases depending on the specific features and functionalities of your online learning platform's progress tracking system.

--

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:

## Test Cases for Online Learning Platform - Assessment and Grading Feature

User Story: As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

Test Case ID: TC-001

Test Case Title: Verify that instructors can create quizzes.

Test Case Description: This test case will verify that instructors can successfully create quizzes with multiple choice questions, true/false questions, and short answer questions.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Instructor account is logged in.

* A course is created.

Test Data:

* Quiz title: Sample Quiz

* Multiple choice questions: 5 questions with 4 options each

* True/false questions: 3 questions

* Short answer questions: 2 questions

Test Steps:

1. Navigate to the "Assessments" section within the course.

2. Click on the "Create Quiz" button.

3. Enter the quiz title, "Sample Quiz."

4. Create multiple choice questions with 4 options each.

5. Create true/false questions.

6. Create short answer questions.

7. Save the quiz.

Postconditions:

* A quiz named "Sample Quiz" is created in the course.

Expected Result: The quiz is successfully created with the defined question types and content.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case ID: TC-002

Test Case Title: Verify that instructors can create tests.

Test Case Description: This test case will verify that instructors can create tests with different question types and various answer formats like multiple choice, fill in the blanks, and essay questions.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Instructor account is logged in.

* A course is created.

Test Data:

* Test title: Sample Test

* Multiple choice questions: 5 questions with 4 options each

* Fill in the blanks: 3 questions

* Essay questions: 2 questions

Test Steps:

1. Navigate to the "Assessments" section within the course.

2. Click on the "Create Test" button.

3. Enter the test title, "Sample Test."

4. Create multiple choice questions with 4 options each.

5. Create fill-in-the-blank questions.

6. Create essay questions.

7. Save the test.

Postconditions:

* A test named "Sample Test" is created in the course.

Expected Result: The test is successfully created with the defined question types and answer formats.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case ID: TC-003

Test Case Title: Verify that instructors can create assignments.

Test Case Description: This test case will verify that instructors can create assignments and specify requirements, deadlines, and grading rubrics.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Instructor account is logged in.

* A course is created.

Test Data:

* Assignment title: Sample Assignment

* Detailed instructions

* Due date: 2024-03-15

* Grading rubric with scoring criteria

Test Steps:

1. Navigate to the "Assessments" section within the course.

2. Click on the "Create Assignment" button.

3. Enter the assignment title, "Sample Assignment"

4. Write detailed instructions for the assignment.

5. Set the due date for the assignment.

6. Create a grading rubric with specific scoring criteria.

7. Save the assignment.

Postconditions:

* An assignment named "Sample Assignment" is created in the course.

Expected Result: The assignment is successfully created with the defined instructions, deadline, and grading rubric.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case ID: TC-004

Test Case Title: Verify automatic grading for objective questions.

Test Case Description: This test case will ensure that objective questions in quizzes and tests are automatically graded and feedback is provided to learners.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Instructor account is logged in.

* A quiz is created with multiple choice questions.

* A test is created with true/false questions

Test Data:

  * Quiz title: Auto-Grading Quiz

  * Test title: Auto Quiz

* Multiple choice answers: Correct options are chosen.

* True/false answers: Correct options are chosen.

Test Steps:

1. Learners attempt the quiz and test.

2. Learners submit their answers.

3. System automatically grades the answers.

Postconditions:

* Grading results are displayed to learners.

Expected Result: The objective questions are automatically graded, and learners receive immediate feedback on their answers.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

 Test Case ID: TC-005

Test Case Title: Verify manual grading for subjective questions.

Test Case Description: This test case will verify that instructors can manually grade subjective questions like essays and provide specific feedback to learners.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Instructor account is logged in.

* A quiz is created with essay questions.

* A test is created with short answer questions

Test Data:

  * Quiz title: Manual Grading Quiz

  * Test title: Manual Grading Test

Test Steps:

1. Learners attempt the quiz and test, completing essay and short answer questions.

2. Learners submit their answers.

3. Instructor views the submitted answers.

4. Instructor manually grades the answers according to the set rubric.

5. Instructor provides specific feedback to learners.

6. Learner views the feedback provided by the instructor.

Postconditions:

* Grading results and feedback are visible to the learner.

Expected Result:

Instructors can access graded submissions, provide individualized feedback for subjective questions, and learners can view the feedback provided.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

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:

## Test Cases for Discussion Forums - Online Learning Platform

Test Case ID: T-DL-001

Test Case Title: Verify that learners can post questions and responses in forums.

Test Case Description: This test case verifies the functionality of a learner posting a question and subsequent response within a discussion forum on the online learning platform.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the platform as a learner.

* A discussion forum within a relevant course is accessible.

Test Data:

* Sample question text: "What is the main concept behind quantum mechanics?"

* Sample response text: "Quantum mechanics deals with the behavior of matter and energy at the atomic and subatomic levels."

Test Steps:

1. Navigate to the specified course and forum.

2. Click the "Post New Thread" button.

3. Enter the sample question into the provided text field.

4. Click the "Post" button to submit the question.

5. Navigate to the same forum and locate the newly posted question.

6. Click the "Reply" button under the question.

7. Enter the sample response into the provided text field.

8. Click the "Post" button to submit the response.

Postconditions:

* Confirmed the post thread is publicly visible.

* The question and response are successfully displayed within the forum thread.

Expected Result:

Both the question and the response will be successfully posted and displayed within the discussion forum thread.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: T-DL-002

Test Case Title: Verify that forums support threaded discussions.

Test Case Description: This test case verifies the functionality of creating nested replies within a discussion forum, ensuring proper threading structure.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the platform as a learner.
* A discussion forum within a relevant course is accessible.
* A discussion thread with existing posts is present.

Test Data:

* Sample reply text 1: "This is a great question! On that note,..."

* Sample reply text 2: "Thanks for the insight! Do you have any further..."

Test Steps:

1. Navigate to the discussion forum thread with existing posts.

2. Reply to one of the existing posts using the "Reply" button and enter Sample reply text 1.

3. Post the reply.

4. Another learner replies to Sample reply text 1 using the "Reply" button and enters Sample reply text 2.

5. Post the second reply.

Postconditions:

* Both replies are displayed correctly within the specified thread.

Expected Result:

Sample reply text 2 appears as a direct reply to Sample reply text 1, clearly indicating the nested reply structure.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: T-DL-003

Test Case Title: Verify that instructors can moderate discussions.

Test Case Description: This test case verifies the functionality of instructors deleting inappropriate posts or locking threads within a discussion forum.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the platform as a learner.

Test Data:

* A new discussion thread with 3 posts (2 from learners and 1 from instructor)

Test Steps:

1. Navigate to the instructor account with administrative access.

2. Access the course with the targeted discussion forum.

3. Identify a learner's post in the existing thread deemed inappropriate.

4. Use the provided moderation tools (e.g., "Delete Post" or "Lock Thread") to take action on the post.

5. Verify the post is deleted or the thread is locked.

Postconditions:

* The targeted post is removed, or the thread heading displays as locked.

Expected Result:

The instructor can successfully delete a learner's post or lock the entire discussion thread.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: T-DL-004

Test Case Title: Verify that learners receive notifications for new posts.

Test Case Description: This test case verifies that learners receive timely notifications about new posts within forums they subscribe to.

Test Suite: User Interface Testing

Test Priority: High

Preconditions:

* User is logged into the platform as a learner.
* Learner has subscribed to a relevant discussion forum.

Test Data: No test data needed

Test Steps:

1. Open the subscribed forum and observe the initial number of posts.

2. Have another user post a new message in the forum thread.

3. Check the learner's notification center or inbox.

4. Verify a notification about the new post is received.

Postconditions:

* The learner's notification center displays a notification indicating a new post in the subscribed forum.

Expected Result:

The learner receives a notification promptly upon a new post being made in the subscribed forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: T-DL-005

Test Case Title: Verify that forums are accessible on all devices.

Test Case Description: This test case verifies that discussion forums function correctly and are visually appealing across different devices (desktop, mobile, tablet).

Test Suite: Cross-Browser and Device Compatibility Testing

Test Priority: Medium

Preconditions:

* User account with access to Learn Platform.

* Access to different devices (desktop, mobile, tablet) with various browsers.

Test Data:

* Sample forum content with varied text formats, images, and links.

Test Steps:

1. Login to the platform using the user account on a desktop computer.

2. Navigate to a discussion forum and interact with its features (posting, replying, viewing attachments, etc.).

3. Repeat the access and interaction steps on a mobile device (smartphone).

4. Repeat the access and interaction steps on a tablet device.

5. Document any differences in functionality or visual presentation across devices.

Postconditions:

* Functionality of the discussion forum is consistent across all devices.

Expected Result:

All features of the forum function seamlessly and the interface adapts appropriately for each device screen size without any noticeable performance issues.

Severity: Major

Priority: Medium

Type of Testing: Cross-Browser and Device Compatibility Testing ,Functional Testing

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:

## Test Cases for "Live Classes and Webinars" Feature

User Story: As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

Test Case 1:

Test Case ID: TC_LIVECLASS_001

Test Case Title: Verify that learners can attend live classes through video conferencing tools.

Test Case Description: This test case verifies if learners can successfully join a scheduled live class using the integrated video conferencing tool, ensuring proper audio and video functionality.

Test Suite: Live Classes Feature

Test Priority: High

Preconditions:

* User account created with correct credentials

* A live class session must be scheduled and accessible

* Stable internet connection

Test Data: No test data needed

Test Steps:

1. Navigate to the platform and log in as a learner.
2. Access the live classes section and locate the scheduled class.
3. Click on "Join Class" / "Attend Webinar" button.
4. Verify that the video conferencing window opens successfully.
5. Ensure audio and video functionalities work correctly.

Postconditions:

* Leave the video conferencing session by using the appropriate exit option.
* Close the web browser or application.

Expected Result: The user should be able to join the live class successfully, with proper audio and video communication capabilities.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_LIVECLASS_002

Test Case Title: Verify that learners can ask questions in real-time during a live session

Test Case Description: This test case ensures that learners can utilize the provided mechanism to ask questions during a live session and that they receive appropriate responses.

Test Suite: Live Classes Feature

Test Priority: High

Preconditions:

* User account created with correct credentials

* A live class session must be running with an instructor present

* Stable internet connection

Test Data: No test data needed

Test Steps:

1. Attend a live session as described in test case TC_LIVECLASS_001.

2. During the session, utilize the designated question interface (chat, Q&A panel).

3. Type a relevant question and submit it.

4. Verify that the question is displayed in the designated area for instructor and other learners to see.

5. Observe if the instructor acknowledges the question and provides a response.

Postconditions:

* Close the video conferencing session using the appropriate method.

* Logout from the platform.

Expected Result:

* The learner's question should appear for the instructor and other participants.

* The instructor should acknowledge the question and deliver a timely and appropriate response.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive, Regression

Test Case 3:

Test Case ID: TC_LIVECLASS_003

Test Case Title: Verify that session recordings are available for later viewing

Test Case Description: This test case confirms that recorded live session files are accessible to learners post-session and that playback functionality works correctly.

Test Suite: Live Classes Feature

Test Priority: Medium

Preconditions:

* A live class session should be conducted and recorded.

* User account created with correct credentials.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.

2. Navigate to the session recordings section and locate the recorded live class.

3. Click on the recording to initiate playback.

4. Verify that the video and audio play smoothly.

5. Check if subtitles or transcripts are available and can be accessed.

Postconditions:

* End the recording playback.

Expected Result: The recorded session should play correctly with video, audio, and optional subtitles/transcripts functioning without errors.

Severity: Major
Priority: medium
Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_LIVECLASS_004
Test Case Title: Verify that instructors can share their screen and resources during a live session

Test Case Description: This test case verifies the functionality of the screen sharing feature for instructors, ensuring they can display relevant content and share resources during the live session.

Test Suite: Live Classes Feature

Test Priority: Medium

Preconditions:

* User account created with correct credentials

* Instructor account with designated permissions

* A live class session must be running and accessible to the instructor

* Stable internet connection

Test Data:

* Sample presentation slides

* Website link

Test Steps:

1. Log in as an instructor.

2. Join the live class session as described in test case TC_LIVECLASS_001.

3. Utilize the screen sharing feature within the video conferencing tool.

4. Share a pre-prepared presentation (slides or document) and navigate through it.

5. Verify that the shared content is displayed clearly for all learners.

Postconditions:

* Stop screen sharing and return to normal view

Expected Result: The instructor can successfully share their screen, displaying the selected content (presentation slides, documents, web pages) to all learners in an accessible and clear manner.

Severity: Major

Priority: medium

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_LIVECLASS_005

Test Case Title: Verify that notifications are sent before live sessions start

Test Case Description: This test case ensures that learners receive timely notifications about upcoming live sessions, allowing them to prepare and join on time.

Test Suite: Live Classes Feature

Test Priority: Medium

Preconditions:

* User account created with correct credentials

* A live class session must be scheduled in the future

* Registered to receive notifications

Test Data: No test data needed

Test Steps:

1. Ensure user is registered to receive notifications about live classes.
2. Log in to the platform and navigate to the "Live Classes" section.
3. Verify that the upcoming live class session is listed with details and an estimated start time.
4. Observe and document the time at which the notification for the live class session is received.
5. Compare the notification time with the actual start time of the session.

Postconditions:

* Check inbox for notification confirmation

Expected Result:

* A notification about the scheduled live class should be sent to the user's registered device (email, push notification, etc.) within a reasonable timeframe before the session is due to start.

Severity: Moderate

Priority: Medium

Type of Testing: Functional Testing, Non-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:

## Online Learning Platform - User Story: Mobile Accessibility

Here are 5 detailed test cases for the "Mobile Accessibility" User Story:

Test Case ID: TC-MOBILE-001

Test Case Title: Verify that the platform is accessible on various smartphones.

Test Case Description: This test case aims to ensure that the platform functions correctly and is visually appealing on different popular smartphone models with varying screen sizes and resolutions.

Test Suite: Mobile Application Accessibility

Test Priority: High

Preconditions:

* Access to a test device (iOS and Android) with different screen sizes and resolutions
* Stable internet connection

Test Data: No test data needed.

Test Steps:

1. Launch the mobile app on the specified test devices.
2. Navigate through different sections of the platform, including the course catalog, course details, video lectures, discussion forums, and progress tracking.
3. Check for responsiveness and smooth scrolling on all pages.
4. Ensure all buttons, links, and interactive elements are easily tappable.
5. Verify that text and images are legible and not distorted on all screens.

Postconditions:

* Record any issues observed with navigation, layout, or functionality on the different devices.
* Capture screenshots of the platform on each device.

Expected Result: The platform should function seamlessly and be visually appealing on all tested smartphone models, with no significant layout issues or broken functionality.

Severity: Critical

Priority: High

Type of Testing: Compatibility Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC-MOBILE-002

Test Case Title: Verify that the mobile UI is optimized for touch navigation.

Test Case Description: This test case evaluates the touch interactions on the mobile app, ensuring smooth and intuitive navigation using touch gestures like swiping, tapping, and pinching.

Test Suite: Mobile Application Usability

Test Priority: High

Preconditions:

* Access to a test device (iOS and Android)
* Stable internet connection

Test Data: No test data needed.

Test Steps:

1. Navigate through various sections of the platform using touch gestures only (swiping, tapping, pinching).
2. Test the responsiveness of buttons and interactive elements to single and double taps.
3. Check the smoothness of scrolling through lists, menus, and long text content.
4. Ensure that pinch-to-zoom functionality works correctly for viewing images and text.
5. Observe the overall feel and ease of navigation using touch interactions.

Postconditions:

* Record any delays, unresponsive elements, or awkward gestures during navigation.
* Gather user feedback on the intuitiveness and ease of touch interaction.

Expected Result: Navigation through the platform should be smooth, responsive, and intuitive using touch gestures, providing a seamless user experience.

Severity: Major

Priority: High

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC-MOBILE-003

Test Case Title: Verify that course materials are accessible and readable on mobile devices.

Test Case Description: This test case ensures that course content displayed on mobile devices is easily readable and visually appealing, with proper formatting and layout for text and multimedia elements.

Test Suite: Content Accessibility

Test Priority: High

Preconditions:

* Access to a test device (iOS and Android)
* Stable internet connection
* An online course with various media types (text, images, videos)

Test Data: No test data needed.

Test Steps:

1. Access a course on the mobile app.
2. Read through text-based course material, verifying font size, line spacing, and readability.
3. View embedded images and videos, ensuring proper display and playback on mobile.
4. Download and view offline course materials, checking for accurate formatting and presentation.
5. Navigate through interactive elements within courses (e.g., quizzes, flashcards) and confirm their functionality.

Postconditions:

* Record any issues with text readability, image display, video playback, or interactive element functionality.
* Capture screenshots of different sections of the course material on mobile.

Expected Result: Course materials should be easily readable and visually appealing on all tested mobile devices, with no issues in displaying or interacting with multimedia content.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Content Testing

Test Case Approach: Positive

Test Case ID: TC-MOBILE-004

Test Case Title: Verify that notifications are sent to mobile devices.

Test Case Description: This test case confirms that the platform can successfully deliver notifications to mobile devices, informing users about updates, course announcements, and other relevant information.

Test Suite: Platform Functionality

Test Priority: Medium

Preconditions:

* Access to a test device (iOS and Android)
* Log in to the platform account on the mobile device
* Enable notifications for the mobile app

Test Data:
* Sample course announcement

Test Steps:

1. Create a sample course announcement on the platform, intended to send a push notification to learners.
2. Verify that the announcement is reflected on the mobile app.
3. Check if the user receives a push notification on their mobile device.
4. Tap on the received notification and verify that it directs them to the correct content (e.g., course announcement).
5. Test different notification types (course completion, deadline reminders, etc.) and ensure they arrive as expected.

Postconditions:

* Confirm if the notification appearance and functionality meet design specifications across different devices.

Expected Result: The platform should successfully send notifications to mobile devices, including accurate content and timely delivery.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Test Case ID: TC-MOBILE-005

Test Case Title: Verify that the platform supports offline access to downloaded materials.

Test Case Description: This test case verifies the functionality of the offline access feature for downloaded course materials, ensuring that learners can access content even without an active internet connection.

Test Suite: Offline Functionality Testing

Test Priority: High

Preconditions:

* Access to a test device (iOS and Android)

* Stable internet connection

Test Data:

* A course with downloadable materials (e.g., PDF documents, audio files)

Test Steps:

1. Download course materials (text and multimedia) to the mobile app while connected to the internet.
2. Disconnect the device from the internet.
3. Attempt to access the downloaded materials within the mobile app.
4. Verify if all downloaded materials are accessible and playable offline.
5. Test switching between online and offline modes, confirming that downloaded materials remain available in offline mode.

Postconditions:

* Record any issues with accessing downloaded content offline.

Expected Result: The platform should allow users to download and access course materials offline, ensuring they can continue learning even without an internet connection.

Severity: Critical

Priority: High

Type of Testing: Functionality Testing, Offline Mode Testing

Test Case Approach: Positive, Negative

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:

## Online Learning Platform - Resource Library Test Cases

Test Case ID: RLB-001

Test Case Title: Verify that learners can access e-books within the resource library

Test Case Description: This test case verifies that learners are able to access and open e-books provided within the resource library section of the online learning platform.

Test Suite: Resource Library Feature

Test Priority: High

Preconditions:

* A user account is created and logged in.
* At least one e-book is available in the resource library.

Test Data:

* A known e-book title and author within the resource library.

Test Steps:

1. Navigate to the "Resource Library" section of the platform.
2. Search for the specific e-book title and author using the search bar.
3. Select the e-book from the search results.
4. Verify that the e-book opens and is properly displayed on the screen.
5. Navigate through the e-book, ensuring proper page transitions.

Postconditions:

* The e-book is closed properly.
* The user returns to the resource library section.

Expected Result: The e-book opens successfully and allows for navigation through its pages.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RLB-002

Test Case Title: Verify that learners can search for specific resources in the library

Test Case Description: This test case ensures that learners can effectively search for desired resources within the resource library using keywords.

Test Suite: Resource Library Feature

Test Priority: High

Preconditions:

* A user account is created and logged in.
* At least three resources (e-books, articles, and reference guides) with relevant keywords are available in the library.

Test Data:

* Relevant keywords related to the available resources (e.g., "programming", "history", "business").

Test Steps:

1. Navigate to the "Resource Library" section of the platform.
2. Enter a specific keyword related to one of the available resources in the search bar.
3. Click the "Search" button.
4. Verify that the search results include the resource containing the searched keyword.
5. Repeat steps 2-4 using different keywords related to different resources.

Postconditions:

* The search results are closed.

* The user returns to the resource library section.

Expected Result: The search function accurately retrieves resources containing the specified keywords.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive,

Test Case ID: RLB-003

Test Case Title: Verify that resources are categorized and tagged for easy navigation

Test Case Description: This test case confirms that resources within the library are categorized and tagged appropriately, allowing learners to easily find relevant content.

Test Suite: Resource Library Feature

Test Priority: High

Preconditions:

* A user account is created and logged in.

* Resources in the library are categorized (e.g., by subject, type, difficulty level) and tagged with relevant keywords.

Test Data:

* A specific resource with known categories and tags.

Test Steps:

1. Navigate to the "Resource Library" section of the platform.
2. Choose a specific category from the available categories using the filter options.
3. Verify that the displayed resources belong to the selected category.
4. Use the tag filter feature to search for resources with a specific tag.
5. Verify that the displayed resources include only those tagged with the selected tag.

Postconditions:

* The category and tag filters are reset.
* The user returns to the full list of resources.

Expected Result: The filtering options accurately display resources based on chosen categories and tags.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RLB-004

Test Case Title: Verify that learners can download resources for offline access

Test Case Description: This test case confirms that learners can download resources like e-books and articles for offline viewing.

Test Suite: Resource Library Feature

Test Priority: Medium

Preconditions:

* A user account is created and logged in.

* At least one resource has a download option available.

Test Data:

* A resource with a download option (e.g., an e-book or PDF article).

Test Steps:

1. Navigate to the "Resource Library" section of the platform.

2. Select the desired resource with a download option.

3. Locate and click the download button for the resource.

4. Verify that the download process starts successfully.

5. Confirm the downloaded file is accessible and opens correctly in a suitable application.

Postconditions:

* The downloaded resource is saved in the designated location.

* The user returns to the resource library section.

Expected Result: The chosen resource downloads successfully and can be opened offline.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RLB-005

Test Case Title: Verify that instructors can recommend resources to learners

Test Case Description: This test case verifies the functionality for instructors to recommend specific resources to learners within a course context.

Test Suite: Resource Library Feature

Test Priority: High

Preconditions:

* A user account acting as an instructor is logged in.
* A course exists with learners enrolled.

Test Data:

* An instructor-approved resource from the library.

Test Steps:

1. Navigate to the designated section within the course where instructors can recommend resources.
2. Select the desired resource from the library to be recommended.
3. Add a short descriptive message for the recommendation (optional).
4. Click the "Recommend" button to add the resource to the course's recommendation list.
5. Log in as a learner enrolled in the same course.
6. Verify that the recommended resource is displayed in the recommended resources section of the course.

Postconditions:

* The resource recommendation is saved within the course context.
* The user returns to the respective course areas.

Expected Result: Instructors successfully recommend resources to learners within the course environment.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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 detailed test cases for the "Peer Review" feature of the Online Learning Platform, following your template:

## Test Case 1: Verify that learners can submit assignments for peer review

Test Case Description: This test case verifies that learners can successfully submit their completed assignments for peer review within the platform.

Test Suite: User Interaction - Assignments
Test Priority: High
Preconditions:
* A learner is logged into the platform.
* The learner is enrolled in a course that offers peer review for assignments.
* A new assignment is available for submission.
Test Data:
* A sample assignment document (in a supported format).
Test Steps:

1. Navigate to the course containing the assignment.
2. Locate the assignment and click on the "Submit for Peer Review" button.
3. Upload the prepared sample assignment document.
4. (Optional) Include any specific instructions for reviewers.

5. Click the "Submit" button to finalize the submission.

Postconditions:

* The assignment appears in the learner's "Submitted for Review" section.
* The assignment is visible to designated reviewers.

Expected Result: The assignment is successfully submitted for peer review and appears in the designated review queue.

Severity: Critical
Priority: High
Type of Testing: Functional, System

## Test Case 2: Verify that peers can provide feedback on submitted assignments

Test Case Description: This test case verifies that peers can access submitted assignments and provide constructive feedback within the designated feedback interface.

Test Suite: User Interaction - Peer Review
Test Priority: High
Preconditions:
* A designated reviewer is logged into the platform.
* The reviewer has access to the peer review assignment queue.

Test Data:

* An already submitted assignment for review.

Test Steps:

1. Access the peer review assignment queue.

2. Select the submitted assignment for review.

3. Read through the assignment content.

4. Utilize the provided feedback tools (e.g., text editor, marking system, rating scale) to add comments and assessments.

5. Click "Submit Feedback" to finalize the review.

Postconditions:

* The feedback is saved against the assigned learner's work.

* The learner is notified (optionally) that feedback is available.

Expected Result: The reviewer successfully provides feedback on the assigned assignment using the platform's interface.

Severity: Critical

Priority: High

Type of Testing: Functional, System, Regression

## Test Case 3: Verify that a structured review process guides the feedback

Test Case Description: This test case verifies that the platform provides a structured process for peer review, ensuring reviewers adhere to guidelines and provide focused feedback.

Test Suite: User Interaction - Peer Review Process

Test Priority: Medium

Preconditions:

* A reviewer is logged in and has an assignment to review.

* The platform offers predefined feedback guidelines or rubrics.

Test Data:

* The platform's feedback guidelines or rubric.

Test Steps:

1. Access the assignment for review.

2. Review the platform's provided feedback guidelines or rubric.

3. Provide feedback taking into account the specified criteria.

4. Submit the feedback.

Postconditions:

* The feedback received aligns with the outlined guidelines or rubric.

Expected Result: The platform effectively guides reviewers to provide structured and meaningful feedback based on predefined criteria.

Severity: Major

Priority: Medium

Type of Testing: Functional, Usability

## Test Case 4: Verify that learners can view feedback on their work

Test Case Description: This test case ensures that learners can access and view the feedback received on their submitted assignments.

Test Suite: User Interaction - Feedback

Test Priority: High

Preconditions:

* A learner is logged into the platform.
* The learner has submitted an assignment for peer review.
* Feedback has been received on the assignment.

Test Data:

* The submitted assignment.

Test Steps:

1. Log in to the platform.
2. Navigate to the submitted assignment.
3. Locate the section displaying feedback.
4. Review the received feedback.

Postconditions:

* The learner can clearly see and read all submitted feedback.

Expected Result: Learners can easily access and view the feedback provided by reviewers on their assignments.

Severity: Critical

Priority: High

Type of Testing: Functional, System

## Test Case 5: Verify that reviewers remain anonymous to ensure unbiased feedback

Test Case Description: This test case confirms that the platform maintains anonymity between reviewers and learners to promote honest and unbiased feedback.

Test Suite: Security - Peer Review Anonymity

Test Priority: Medium

Preconditions:

* A learner has submitted an assignment for review.
* A peer reviewer has provided feedback on the assignment.

Test Data: N/A

Test Steps:

1. The learner reviews the received feedback.
2. The learner tries to identify the reviewer based on the feedback or any associated information.

Postconditions:

* The learner is unable to determine the identity of the reviewer.

Expected Result: The platform successfully implements anonymity for reviewers, protecting their identities and encouraging impartial feedback.

Severity: Major
Priority: Medium
Type of Testing: Functional, Security

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:

Okay, I can help you with creating detailed test cases for the "Certificates and Badges" feature of your Online Learning Platform.

Here are 5 test cases following your template:

Test Case 1

* **Test Case ID:** TC_C01
* **Test Case Title:** Verify that learners earn a certificate upon successful course completion.
* **Test Case Description:** This test case verifies that learners receive a certificate automatically after achieving a passing grade in a designated course.
* **Test Suite:** Certification Feature
* **Test Priority:** High
* **Preconditions:**
    * A learner account is created.

   * A course with a defined passing grade is added to the platform.

   * The learner enrolls in the course and completes all its modules and assessments.

* **Test Data:** Course with activities, assessment questions, and a passing grade threshold. Learner account with progress data.

* **Test Steps:**

   1. Log in as the learner.

   2. Enrol in the designated course.

   3. Complete all course modules and assessments.

   4. Achieve the passing grade for the course.

   5. Navigate to the learner's profile section.

   6. Verify that a certificate for the completed course is displayed.

* **Postconditions:** None

* **Expected Result:** The learner's profile should display a certificate for the completed course.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TC_B01

* **Test Case Title:** Verify that badges are awarded for specific achievements and milestones.

* **Test Case Description:** This test case checks if badges are automatically awarded to learners upon reaching certain milestones within a course or for specific actions.

* **Test Suite:** Certification Feature

* **Test Priority:** High

* **Preconditions:**

  * A course with predefined badges linked to specific achievements (e.g., completing a module, earning a high score on a quiz) is added to the platform.

* **Test Data:** Course outline with milestones and associated badges, learner account.

* **Test Steps:**

  1. Log in as the learner.

  2. Enroll in the course.

  3. Complete the specified milestone or achieve the required action.

  4. Verify that the corresponding badge is added to the learner's profile.

* **Postconditions:** None

* **Expected Result:** The learner's profile should display the earned badge.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TC_A01

* **Test Case Title:** Verify that certificates and badges are accessible through learner profiles.

* **Test Case Description:** This test case ensures that learners can easily view their earned certificates and badges within their profile section.

* **Test Suite:** Certification Feature

* **Test Priority:** Medium

* **Preconditions:**

  * A learner account is created with both certificates and badges awarded.

* **Test Data:** Learner account with awarded certificates and badges.

* **Test Steps:**

1. Log in as the learner.
2. Navigate to the learner's profile section.
3. Verify that a dedicated section for certifications and badges is displayed.
4. Verify that all earned certificates and badges are listed correctly.

* **Postconditions:** None
* **Expected Result:** The learner's profile should clearly display a dedicated section for certificates and badges.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing, Acceptance Testing
* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TC_S01
* **Test Case Title:** Verify that learners can share their certificates and badges on social media.
* **Test Case Description:** This test case confirms the functionality to share certificates and badges on popular social media platforms.
* **Test Suite:** Certification Feature
* **Test Priority:** Medium
* **Preconditions:**
  * The user has connected their social media accounts (e.g., Facebook, Twitter) to their platform profile.
* **Test Data:** Certificates and badges, connected social media accounts.
* **Test Steps:**
  1. Log in as the learner.
  2. Navigate to the 'Certificates and Badges' section.
  3. Select a certificate or badge to share.

    4. Choose a desired social media platform from the share options.

    5. Verify that the certificate or badge is successfully shared on the selected platform.

* **Postconditions:** None

* **Expected Result:** The selected certificate or badge should be shared on the connected social media account.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC_IB01

* **Test Case Title:** Verify that instructors can issue custom badges for special achievements.

* **Test Case Description:** This test case checks the instructor's ability to create and award custom badges for unique learner achievements or contributions.

* **Test Suite:** Certification Feature

* **Test Priority:** Medium

* **Preconditions:**

    * An instructor account is created with the necessary permissions to manage badges.

* **Test Data:** Custom badge definition (criteria, name, image, description), learner account

* **Test Steps:**

    1. Log in as the instructor.

    2. Navigate to the 'Badges' management section.

    3. Create a new custom badge with specific criteria and details.

4. Identify a learner who meets the badge criteria.
5. Award the custom badge to the selected learner.
6. Verify that the learner receives the custom badge in their profile.

* **Postconditions:** None

* **Expected Result:** The instructor successfully creates a custom badge and awards it to a learner, who then sees it in their profile.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

Let me know if you want to explore any of these test cases further, or if you'd like to create test cases for other aspects of the "Certificates and Badges" feature.

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:

## Test Cases for Gamification Elements

Project Name: Online Learning Platform

Feature Name: Gamification Elements

User Story: As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

### Test Case 1: Verify That Learners Earn Points for Completing Activities

Test Case ID: TC_GAM001

Test Case Title: Verify that learners earn points for completing activities.

Test Case Description: This test case verifies that learners receive points for various activities on the platform, such as watching video lectures, downloading resources, participating in forums, and completing quizzes.

Test Suite: Gamification Features

Test Priority: High

Preconditions:

* A learner account is created and logged in.
* A course with various activities is accessible to the learner.

Test Data:

* Expected points values for each activity type (e.g., 10 points for watching a video lecture, 5 points for downloading a resource, etc.).

Test Steps:

1. Access the chosen course and navigate to different activity types.
2. Complete each activity (e.g., watch a video lecture in its entirety, download a resource, post a reply in a forum thread, answer all questions correctly in a quiz).
3. Review the learner's profile page to check the updated points balance.
4. Compare the earned points with the expected values for each activity.

Postconditions:

* Learner's profile page should accurately reflect the earned points.

Expected Result:

Points are awarded to the learner for each completed activity as per the predefined values.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 2: Verify That Levels are Achieved Based on Accumulated Points

Test Case ID: TC_GAM002

Test Case Title: Verify that levels are achieved based on accumulated points.

Test Case Description: This test case verifies the mechanism for level progression based on accumulated points.

Test Suite: Gamification Features

Test Priority: High

Preconditions:

* A learner account is created and logged in.

* A course with various activities is accessible to the learner.

Test Data:

* Point thresholds for each level (e.g., Level 1: 0-50 points, Level 2: 51-100 points, etc.).

* Preliminary points assigned to the learner for testing.

Test Steps:

1. Complete several activities to gain points and increase the learner's total points balance.
2. Monitor the learner's profile page for level updates.
3. Ensure the learner progresses to the next level when they reach the corresponding point threshold.

Postconditions:

* Learner's profile page should accurately reflect the attained level.

Expected Result:

The learner progresses through levels as they accumulate points, reaching the next level upon surpassing the predefined point threshold.

Severity: Critical

Priority: High

Type of Testing: Functional, Regression Testing

Test Case Approach: Positive

### Test Case 3: Verify That Leaderboards Display Top-Performing Learners

Test Case ID: TC_GAM003

Test Case Title: Verify that leaderboards display top-performing learners.

Test Case Description: This test case ensures that leaderboards correctly display learners ranked by their accumulated points.

Test Suite: Gamification Features

Test Priority: High

Preconditions:

* At least three learner accounts are created with varying point balances.

Test Data:

* Points values assigned to each learner.

Test Steps:

1. Navigate to the leaderboards section of the platform.
2. Verify that the leaderboard displays a sorted list of learners based on their point totals.
3. Ensure each learner's corresponding point count is accurately shown next to their name.

Postconditions:

* Ranked list on the leaderboard should reflect the point balances accurately.

Expected Result:

The leaderboard correctly displays learners ranked in descending order based on their accumulated points, with the highest-scoring learner at the top.

Severity: Critical

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 4: Verify That Badges are Awarded for Reaching Milestones

Test Case ID: TC_GAM004
Test Case Title: Verify that badges are awarded for reaching milestones.
Test Case Description: This test case verifies that badges are awarded to learners upon reaching specific milestones, such as completing a certain number of courses or earning a specific number of points.

Test Suite: Gamification Features
Test Priority: Medium
Preconditions:
* A learner account is created and logged in.
* Define specific milestones with corresponding badges (e.g., 5 completed courses = "Active Learner" badge).

Test Data:
* Points thresholds or activities required to unlock each badge.

Test Steps:

1. Guide the learner to complete the required activities to achieve a specific milestone.
2. Check the learner's profile page for the newly awarded badge after reaching the milestone.

Postconditions:

* Learner profile page should display the earned badge.

Expected Result:

The learner receives the designated badge once they successfully achieve the predetermined milestone.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

### Test Case 5: Verify That Learners Can View Their Progress in a Gamification Dashboard

Test Case ID: TC_GAM005

Test Case Title: Verify that learners can view their progress in a gamification dashboard.

Test Case Description: This test case verifies the functionality of a gamification dashboard that allows learners to track their progress, points, levels, and earned badges.

Test Suite: Gamification Features

Test Priority: Medium

Preconditions:

* A learner account is created and logged in.

* A gamification dashboard is accessible to the learner.

Test Data:

* Learner's current points, level, and badges.

Test Steps:

1. Navigate to the gamification dashboard.

2. Verify that the dashboard displays the learner's current points balance, attained level, and a list of earned badges.

3. Ensure that the dashboard provides clear and concise visual representations of the learner's progress.

 Postconditions:

* Dashboard should reflect the learner's latest progress accurately.

Expected Result:

The gamification dashboard presents a comprehensive overview of the learner's progress, including points, level, and badges achieved, allowing for easy monitoring and tracking.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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:

## Test Cases for Multilingual Support Feature

Test Case ID: TC_ML_001

Test Case Title: Verify that the platform interface is available in multiple languages.

Test Case Description: This test case will validate that the platform interface offers multiple language options as selectable by the user.

Test Suite: Feature - Multilingual Support

Test Priority: High

Preconditions:

* Platform is accessible.

* User is logged in (optional)

Test Data: List of supported languages

Test Steps:

1. Navigate to the platform homepage.

2. Locate and click on the language selection dropdown/menu.

3. Select each supported language from the list.

4. Verify that the platform interface successfully changes to the selected language.

5. Verify that all key elements (navigation, buttons, headers, etc.) are correctly translated in the chosen language.

Postconditions:

* Platform interface remains in selected language.

* User can navigate freely within the platform.

Expected Result: Platform interface should seamlessly switch to the selected language, displaying all content accurately and responsively.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_ML_002

Test Case Title: Verify that course materials are available in different languages.

Test Case Description: This test case will ensure that the platform provides course content in the chosen languages, covering various media types (text, videos, downloads).

Test Suite: Feature - Multilingual Support

Test Priority: High

Preconditions:

* Platform is accessible.

* User is logged in.

* At least one course with multi-language content is available.

Test Data:

* List of supported languages

* Course ID containing multi-language materials

Test Steps:

1. Navigate to the selected course page.

2. Access the language selection dropdown/menu.

3. Select a language different from the default.
4. Verify that all text content within the course module (lectures, descriptions, assignments) are accurately translated.
5. Verify that downloadable resources (PDFs, presentations) are available in the chosen language.
6. (Optional) Verify that video lectures have captions or subtitles in the selected language.

Postconditions:

* Course materials remain in selected language.
* User can navigate and interact with the course content.

Expected Result: Course materials, including text, downloadable resources, and videos, should be accurately translated and available in the selected language.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Localization Testing

Test Case Approach: Positive

Test Case ID: TC_ML_003

Test Case Title: Verify that learners can select their preferred language at any time.

Test Case Description: This test case will confirm that users can change their preferred language during their platform interaction.

Test Suite: Feature - Multilingual Support

Test Priority: High

Preconditions:

* Platform is accessible.
* User is logged in.
* Platform is set to a default language.

Test Data: List of supported languages

Test Steps:

1. Login to the platform.

2. Navigate to a specific course or section.

3. Locate and click on the language selection dropdown/menu.

4. Select a different language from the default.

5. Verify that all platform elements reflect the new language selection.

6. Repeat steps 3-5, switching to another supported language.

Postconditions:

* Platform interface remains in the chosen language.

* User settings reflect the latest language preference.

Expected Result: Learners should be able to change their preferred language at any point during their platform interaction without affecting their account or existing progress.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_ML_004

Test Case Title: Verify that multilingual support includes translated help and support resources.

Test Case Description: This test case aims to ensure that the platform's help section and support resources are available in the chosen languages.

Test Suite: Feature - Multilingual Support

Test Priority: Medium

Preconditions:

* Platform is accessible.
* User is logged in (optional)
* User have selected a language other than the default.

Test Data: List of supported languages

Test Steps:

1. Navigate to the platform's help section.
2. Verify that the help articles and FAQs are translated into the chosen language.
3. (Optional) Test the platform's contact support feature and verify that the communication is possible in the chosen language.

Postconditions:

* Platform help section remains accessible in the chosen language.
* User can access required support resources in the chosen language.

Expected Result: All help and support resources should be readily available and accurately translated in the chosen language.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Localization Testing

Test Case Approach: Positive

Test Case ID: TC_ML_005

Test Case Title: Verify that instructors can provide content in multiple languages.

Test Case Description: This test case will ensure instructors can upload and manage course materials in different languages.

Test Suite: Feature - Multilingual Support

Test Priority: Medium

Preconditions:

* Platform is accessible.

* Instructor account is logged in.

* A new course is created.

* Supported languages are accessible for course material.

Test Data:

* Sample course content (text, video, download) in various supported languages.

Test Steps:

1. Access the course creation interface.

2. Upload the sample course content in different languages.

3. Verify that the content is saved and displayed correctly in each chosen language.

4. Manage the content visibility for different languages.

5. (Optional) Verify the option to translate automated content within the platform (e.g., course descriptions).

Postconditions:

* Course content is stored and organized based on different languages.

* Instructors can choose specific languages for course visibility and access.

Expected Result: Instructors should be able to seamlessly create and manage course materials in multiple languages, enabling them to cater to a diverse learner base.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Content Management Testing

Test Case Approach: Positive

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 five detailed test cases for the "Accessibility Features" user story on the Online Learning Platform, structured using the provided template:

Test Case 1: Verify that the platform is compatible with screen readers

* **Test Case Description:** This test case aims to verify that the online learning platform's elements are accessible and understandable to users utilizing screen readers.
* **Test Suite:** Accessibility Testing
* **Test Priority:** High
* **Preconditions:**
    * A screen reader application (e.g., NVDA, JAWS) must be installed and configured
    * A compatible web browser is used.
* **Test Data:** No test data needed
* **Test Steps:**
    1. Launch the platform using the configured web browser and screen reader.
    2. Navigate through the platform's main navigation menu (Home, Courses, About, etc.) using the screen reader's controls.
    3. Access a sample course and verify that the course content (lectures, text, images, and interactive elements) is announced clearly and in a logical sequence.
    4. Attempt to interact with interactive elements like buttons, links, and forms using the screen reader.
    5. Verify that keyboard navigation functions correctly with screen reader feedback.
* **Postconditions:**

    * Exit the platform and close the browser.
* **Expected Result:** All platform elements are announced accurately by the screen reader, enabling users to understand the content and interact effectively.

Test Case 2: Verify that learners can adjust font sizes for better readability

* **Test Case Description:** This test case assesses the functionality of the font size adjustment feature and ensures it improves readability for various user needs.
* **Test Suite:** Accessibility Testing
* **Test Priority:** Medium
* **Preconditions:**
    * Access to user settings or preferences on the platform.
* **Test Data:**
    * Different font size values (e.g., 12px, 14px, 16px, 18px).
* **Test Steps:**
    1. Login to the platform as a learner.
    2. Navigate to user settings or preferences.
    3. Locate the font size adjustment option.
    4. Increase the font size to a larger value.
    5. Verify that all text content on the platform (headings, paragraphs, buttons, links) increases in size proportionally.
    6. Repeat steps 4-5, decreasing the font size to smaller values.
* **Postconditions:**

* **Expected Result:** Learners can seamlessly adjust the font size to a comfortable level that enhances their

reading experience.

Test Case 3: Verify that color contrast options are available for improved visibility

* **Test Case Description:** This test case evaluates the presence and functionality of color contrast options, ensuring improved visibility for users with visual impairments.
* **Test Suite:** Accessibility Testing
* **Test Priority:** Medium
* **Preconditions:**
    * Access to user settings or preferences on the platform.
* **Test Data:**
    * Different color preference options (e.g., light mode, dark mode).
* **Test Steps:**
    1. Login to the platform as a learner.
    2. Navigate to user settings or preferences.
    3. Locate the color contrast or theme selection option.
    4. Switch between different color schemes (e.g., light mode to dark mode, or vice versa).
    5. Verify that the platform's text, backgrounds, and UI elements adjust accordingly, maintaining sufficient color contrast for improved readability.
* **Postconditions:**

* **Expected Result:** Users can select color schemes that optimize visibility and reduce strain on their eyes.

Test Case 4: Verify that accessibility features are easy to activate and customize

* **Test Case Description:** This test case assesses the ease of use and customization of accessibility

features, ensuring users can quickly enable them and tailor settings to their preferences.

* **Test Suite:** Accessibility Testing
* **Test Priority:** Medium
* **Preconditions:**
    * Access to the user settings or preferences section on the platform.
* **Test Data:**
    * No specific test data required.
* **Test Steps:**
    1. Login to the platform.
    2. Navigate to the user settings or preferences area.
    3. Locate the accessibility settings.
    4. Activating each accessibility feature (e.g., screen reader compatibility, font size adjustment, color contrast) one at a time.
    5. Verify that features activate or deactivate without any unexpected behavior.
    6. Navigate through the accessibility settings and modify various options (e.g., font size, contrast level, etc.)
* **Postconditions:**

* **Expected Result:** Users can effortlessly find, enable, and customize accessibility features within the platform's settings menu.

Test Case 5: Verify that support resources are available to assist learners with disabilities

* **Test Case Description:** This test case checks if the platform provides accessible support resources and information to help learners with disabilities navigate and utilize the platform effectively.
* **Test Suite:** Accessibility Testing, User Support Testing

* **Test Priority:** Medium

* **Preconditions:**

    * Access to the platform's "Help" or "Support" section.

* **Test Data:**

    * No specific test data required.

* **Test Steps:**

    1. Navigate to the platform's "Help" or "Support" section.

    2. Look for information specifically addressing accessibility features and their use.

    3. Examine if there are FAQs, tutorials, or contact information dedicated to assisting learners with disabilities.

* **Postconditions:**

* **Expected Result:** The platform offers easily accessible support resources and contact options for learners who require assistance with accessibility-related issues.

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:

## Test Cases for Discussion Moderation Feature

Test Case ID: TC_DISC_001

Test Case Title: Verify that instructors can edit posts in discussions.

Test Case Description: This test case verifies the functionality of the "Edit" option for instructors in discussion forums. It checks if instructors can modify existing posts and if changes are correctly reflected.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- An instructor account is logged in.

- An active discussion forum exists.

- The instructor has authored a post in the forum.

Test Data:

- A sample post content authored by the instructor.

- Updated content for the post.

Test Steps:

1. Navigate to the discussion forum.

2. Locate the post authored by the instructor.

3. Click the "Edit" option associated with the post.

4. Modify the existing post content.

5. Click "Save Changes" to apply the modifications.

Postconditions:

- The modified post content is displayed in the forum.

Expected Result: The post content is successfully edited and updated in the forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_DISC_002

Test Case Title: Verify that instructors can delete posts in discussions.

Test Case Description: This test case verifies the functionality of the "Delete" option for instructors in discussion forums. It checks if instructors can remove posts and if deleted posts are permanently removed.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- An instructor account is logged in.

- An active discussion forum exists.

- The instructor has authored a post in the forum.

Test Data:

- A sample post authored by the instructor.

Test Steps:

1. Navigate to the discussion forum.

2. Locate the post authored by the instructor.

3. Click the "Delete" option associated with the post.

4. Confirm the deletion action in the prompt.

Postconditions:

- The deleted post is no longer visible in the forum.

Expected Result: The selected post is successfully deleted from the forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_DISC_003

Test Case Title: Verify that instructors can highlight posts in discussions.

Test Case Description: This test case verifies the functionality of the "Highlight" option for instructors in discussion forums. It checks if instructors can mark posts for attention and if highlighted posts are visually distinct.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- An instructor account is logged in.

- An active discussion forum exists.

Test Data:

- A regular post in the forum.

Test Steps:

1. Navigate to the discussion forum.

2. Locate a post written by a participant.

3. Click the "Highlight" option associated with the post.

Postconditions:

- The highlighted post is visually distinguished from other posts (e.g., bold text, different background color).

Expected Result: The selected post is successfully highlighted and visually distinguishable from other posts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_DISC_004

Test Case Title: Verify that instructors can mute disruptive users.

Test Case Description: This test case checks if instructors can silence disruptive users in discussion forums and if muted users' posts are no longer visible.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- An instructor account is logged in.

- An active discussion forum exists.

- A disruptive user account is available within the forum.

Test Data:

- Pre-written disruptive content from the disruptive user account.

Test Steps:

1. Initiate a new discussion thread in the forum.

2. Have the disruptive user post a disruptive message.

3. Access the user settings for the disruptive user.

4. Select the "Mute" option.

5. Observe the forum and confirm the disruptive user's posts are no longer visible.

Postconditions:

- The disruptive user's posts are no longer visible in the forum.

Expected Result: The disruptive user is successfully muted, and their posts are hidden from the forum.

Severity: Critica

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_DISC_005

Test Case Title: Verify that moderation log records are accurate.

Test Case Description: This test case verifies whether the moderation log accurately records instructor actions taken in the discussion forum. It includes checking the type of action taken (edit, delete, mute), the date and time of the action, and the username of the user who performed the action.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- An instructor account is logged in.

- An active discussion forum exists.

- A sample post in the forum.

Test Data:

- A sample post for testing edit, delete, and mute actions.

Test Steps:

1. Edit the sample post.

2. Delete the sample post.

3. Mute a participant in the forum.

4. Access the moderation log.

5. Verify that each action (edit, delete, mute) is accurately recorded in the log with the correct date/time and user identifier.

Postconditions:

- The moderation log is updated.

Expected Result: All moderation actions are accurately recorded in the log.

Severity: Major

Priority: High

Type of Testing: Functional Testing

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

User Story: As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

Test Case ID: TC_LANA_001

Test Case Title: Verify that engagement metrics are displayed on the analytics dashboard.

Test Case Description: This test case verifies that the learning analytics dashboard displays the expected engagement metrics for a course, including metrics such as total learners, average time spent on course content, and most viewed resources.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:
* An instructor account is logged in.
* A course with learners enrolled is available.

Test Data: No test data needed

Test Steps:

1. Navigate to the learning analytics dashboard for the chosen course.
2. Verify that the dashboard displays the following engagement metrics:
    * Total Learners enrolled
    * Average Time Spent per Lesson
    * Most Accessed Resources
3. Validate the displayed data against actual learner activities within the course.

Postconditions:

* The instructor remains on the analytics dashboard.

Expected Result: All specified engagement metrics are displayed accurately on the dashboard.

Severity: Critical **| Priority:** High **| Type of Testing:** Functional Testing

Test Case Approach: Positive

Test Case ID: TC_LANA_002

Test Case Title: Verify that performance trends are visualized over time.

Test Case Description: This test case ensures that the learning analytics dashboard effectively visualizes performance trends over time for learners enrolled in the course, including progress on modules, quiz scores, and overall completion rates.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:
* An instructor account is logged in.
* A course with learners enrolled is available.
* Enough learner activity has occurred to generate meaningful trends.

Test Data: No test data needed

Test Steps:

1. Navigate to the learning analytics dashboard for the chosen course.
2. Identify the section showcasing performance trends over time.
3. Verify that the trends are visualized using an appropriate chart type (e.g., line chart, bar chart).
4. Analyze the visualized trends for key performance indicators:
    * Module completion rates
    * Average quiz scores
    * Overall course completion rates
5. Ensure the trends accurately reflect the learner performance data.

Postconditions:

* The instructor remains on the analytics dashboard.

Expected Result: Performance trends are effectively visualized and accurately represent learner performance over time.

Severity: Critical **| Priority:** High **| Type of Testing:** Functional Testing

Test Case Approach: Positive

Test Case ID: TC_LANA_003

Test Case Title: Verify that instructors can access detailed learner progress reports.

Test Case Description: This test case confirms that instructors can access detailed reports on individual learner progress within a course, including module completion status, quiz scores, and overall engagement metrics.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:
* An instructor account is logged in.
* A course with learners enrolled is available.

Test Data: No test data needed

Test Steps:

1. Navigate to the learning analytics dashboard for the chosen course.
2. Locate the section dedicated to learner progress reports.
3. Select a specific learner from the list of enrolled students.
4. Review the detailed progress report for the selected learner, verifying the following information:
    * Module completion status for each module
    * Scores achieved on quizzes within the course
    * Engagement metrics such as time spent on each module, resources accessed, and forum participation.

Postconditions:

* The instructor remains on the learner progress report page.

Expected Result: The detailed progress report provides comprehensive information on the individual learner's performance and engagement within the course.

Severity: Critical **| Priority:** High **| Type of Testing:** Functional Testing

Test Case Approach: Positive

Test Case ID: TC_LANA_004

Test Case Title: Verify real-time updating of analytics data.

Test Case Description: This test case confirms that the learning analytics data on the dashboard is updated in real-time as learners interact with the platform.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

* An instructor account is logged in.

* A course with learners actively engaged is available.

Test Data: No test data needed

Test Steps:

1. Monitor the learning analytics dashboard for the chosen course.

2. Observe learner activity within the course, such as module completion, quiz attempts, or forum posts.

3. Track the corresponding changes in the displayed analytics metrics on the dashboard.

4. Verify the updates reflect the learner activity in a timely manner.

Postconditions:

* The instructor remains on the analytics dashboard.

* Learner activity continues within the course.

Expected Result: Analytics data on the dashboard updates in real-time as learners engage with the platform.

Severity: Major **| Priority:** Medium **| Type of Testing:** Performance Testing, Functional Testing

Test Case Approach: Positive, Monitoring

Test Case ID: TC_LANA_005

Test Case Title: Verify the export functionality of analytics reports.

Test Case Description: This test case ensures that instructors can export analytics reports into a usable format for further analysis or sharing.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

* An instructor account is logged in.
* A course with learners enrolled is available.
* Analytics data is available on the dashboard.

Test Data: No test data needed

Test Steps:

1. Navigate to the desired analytics report on the dashboard.
2. Identify the export functionality.
3. Select an appropriate export format (e.g., CSV, PDF).
4. Initiate the export process.
5. Verify that the downloaded file contains the expected analytics data.
6. Validate the data format and structure against the original report.

Postconditions:

* The export file is downloaded successfully.

* The instructor can access the exported data.

Expected Result: The analytics report is exported successfully in the selected format, preserving all relevant data and structure.

Severity: Major **| Priority:** Medium **| 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:

## Test Cases for Social Learning Features - Group Projects

Project Context: Online Learning Platform

Feature Name: Social Learning Features

User Story: As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Test Case 1

Test Case ID: TC_SLF_GRP_001

Test Case Title: Verify that learners can create a new group for a project.

Test Case Description: This test case verifies the functionality of creating a new group for a project by a learner.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

* A logged-in learner account.

* An active course with a group project option.

Test Data:

* Group name: "TestGroup1"

* Group description: "Project group for testing"

Test Steps:

1. Navigate to the course page displaying the group project option.

2. Click on the "Create New Group" button.

3. Enter the group name "TestGroup1" and description "Project group for testing".

4. Click on the "Create Group" button.

5. Verify that a new group named "TestGroup1" is created successfully and the learner is added as a member.

Postconditions:

* The newly created group remains accessible to the learner.

Expected Result: The group is successfully created and the learner is added as a member.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_SLF_GRP_002

Test Case Title: Verify that learners can join an existing group.

Test Case Description: This test case verifies the functionality of joining an existing group for a project by a learner.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

* An active course with a group project option.

* An existing group named "TestGroup1" created in TC_SLF_GRP_001.

* A logged-in learner account different from the account used in TC_SLF_GRP_001.

Test Data:

* Group name: "TestGroup1"

Test Steps:

1. Navigate to the course page displaying the group project option.

2. Click on the "Join Group" button.

3. Select "TestGroup1" from the list of available groups.

4. Click on the "Join Group" button.

5. Verify that the learner is added as a member of "TestGroup1".

Postconditions:

* The learner can now see and interact with the group and its members.

Expected Result: The learner successfully joins the "TestGroup1" group.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_SLF_GRP_003

Test Case Title: Verify that learners can share resources within a group.

Test Case Description: This test case verifies the functionality of sharing resources like documents, links, or files within a group.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

* "TestGroup1" group created and a learner is successfully joined in TC_SLF_GRP_002

* A local file to be shared (e.g., a document, presentation, or image).

Test Data:

* Resource file: "TestResource.pdf"

Test Steps:

1. Navigate to the "TestGroup1" group page.

2. Click on the "Share Resource" button.

3. Select and upload "TestResource.pdf" file.

4. Add a brief description for the shared resource (optional).

5. Click on the "Share" button.

6. Verify that "TestResource.pdf" is listed as shared resource within the group.

Postconditions:

* The resource can be accessed by other group members.

Expected Result: The resource is successfully shared with all group members.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

(Continue generating 2 more test cases covering remaining acceptance criteria with similar structure and detail)

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:

## Online Learning Platform: Content Authoring Tools Test Cases

Test Case ID: TC_CAT_001

Test Case Title: Verify that instructors can create multimedia presentations

Test Case Description: This test case aims to verify the functionality of creating multimedia presentations within the content authoring tools. The test will involve adding slides, inserting images, videos, and audio, and configuring slide transitions and animations.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:
* The instructor is logged into the platform.
* An active course is available for the instructor to access.

Test Data:
* Images, videos, and audio files to be incorporated into the presentation.

Test Steps:
1. Navigate to the content authoring tools for the selected course.
2. Initiate the creation of a new multimedia presentation.
3. Add slides to the presentation.
4. Insert various media types (images, videos, audio) into the slides.
5. Configure and apply slide transitions and animations.
6. Preview the presentation before publishing.

Postconditions:

* The multimedia presentation is saved in the course content.

Expected Result: The instructor is able to successfully create and preview a multimedia presentation with integrated media elements and transitions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_CAT_002

Test Case Title: Verify that interactive modules can be developed and integrated

Test Case Description: This test case evaluates the functionality of creating and integrating interactive modules within the content authoring tools. It will involve creating modules with various interactive elements like quizzes, drag-and-drop exercises, simulations, and branching scenarios.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

* The instructor is logged into the platform.

* An active course is available for the instructor to access.

* A sample module template or pre-built interactive elements are available (if applicable).

Test Data:

* Content and data required for the interactive module (e.g., quiz questions, drag-and-drop objects, simulation scenarios).

Test Steps:

1. Access the content authoring tools and select the interactive module creation option.

2. Choose and configure an appropriate module template or build the module from scratch.

3. Incorporate interactive elements like quizzes, drag-and-drop exercises, or simulations according to the module design.

4. Set up branching scenarios or alternative pathways within the module (if applicable).

5. Preview the interactive module to ensure proper functionality and user experience.

Postconditions:

* The interactive module is saved and integrated into the course content.

Expected Result: The instructor is able to successfully create and integrate a functional interactive module with desired interactive elements and user engagement.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Continue creating test cases following the outlined template.

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:

## Online Learning Platform: Learning Paths & Curriculum Test Cases

Test Case ID: TC-LCP-001

Test Case Title: Verify that learning paths include sequential courses and modules.

Test Case Description: This test case verifies the platform's ability to present learning paths with a clear and logical sequence of courses and modules.

Test Suite: Learning Paths Feature Test Suite

Test Priority: High

Preconditions:

* Login as a learner.

* Access the learning paths section.

* Select a predefined learning path.

Test Data: A predefined learning path with multiple courses and modules.

Test Steps:

1. Navigate to the selected learning path.

2. Observe the course and module structure presented in the learning path.

3. Verify that the courses and modules are listed in a logical and sequential order.

4. Select the first course and then the first module within that course.

5. Verify that the user is directed to the content for the next item in the sequence.

Postconditions:

* Learner returns to the learning path page.

Expected Result: The learning path displays courses and modules in a sequential order, making it clear to the learner the intended progression.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-LCP-002

Test Case Title: Verify that curricula are aligned with specific learning objectives.

Test Case Description: This test case verifies that each course and module within a learning path clearly states its specific learning objectives.

Test Suite: Learning Paths Feature Test Suite

Test Priority: High

Preconditions:

* Login as a learner.

* Access the learning paths section.

* Select a predefined learning path.

Test Data: A predefined learning path with multiple courses and modules.

Test Steps:

1. Navigate to a course within the selected learning path.

2. Verify that a section outlining the learning objectives is clearly visible.

3. Read the listed learning objectives for the course.

4. Repeat steps 1-3 for several modules within the selected course.

5. Confirm that the learning objectives for each module are directly related and contribute to the overall goals of the course.

Postconditions:

* Learner returns to the learning path page.

Expected Result: Each course and module within a learning path outlines its specific, measurable, achievable, relevant, and time-bound (SMART) learning objectives.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-LCP-003

Test Case Title: Verify that learners can track their progress within a learning path.

Test Case Description: This test case confirms the learner's ability to monitor their completion status and advancement through a chosen learning path.

Test Suite: Learning Paths Feature Test Suite

Test Priority: Medium

Preconditions:

* Login as a learner.

* Access the learning paths section.

* Select a predefined learning path and begin navigating through it.

Test Data: A predefined learning path with multiple courses and modules.

Test Steps:

1. Complete multiple courses and modules within the selected learning path.

2. Return to the learning path overview page.

3. Verify that the completion status of each course and module is accurately reflected.

4. Observe the learner's progress visualization (e.g., a progress bar) and confirm its alignment with the completed courses and modules.

5. Attempt to access a course or module that has not been completed and verify that access is restricted.

Postconditions:

* Learner remains on the learning path overview page.

Expected Result: The platform clearly displays the learner's progress through the learning path, including the completion status of courses and modules, and any visual progress indicators.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-LCP-004

Test Case Title: Verify that instructors can customize learning paths for different needs.

Test Case Description: This test case verifies the instructor's ability to modify existing learning paths or create new ones tailored to specific audiences or learning objectives.

Test Suite: Learning Paths Feature Test Suite

Test Priority: Medium

Preconditions:

* Login as an instructor.

* Access the learning paths management section.

Test Data: A predefined learning path and sample course and module content.

Test Steps:

1. Choose an existing learning path to modify.

2. Adjust the sequence of courses and modules within the path.

3. Add new courses or modules to the existing path.

4. Remove existing courses or modules from the path.

5. Verify that changes to the learning path structure are successfully saved.

Postconditions:

* Instructor returns to the learning path management page.

Expected Result: The platform allows instructors to customize learning paths by rearranging, adding, or removing courses and modules, enabling them to create tailored learning experiences.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-LCP-005

Test Case Title: Verify that notifications are sent for upcoming courses or milestones.

Test Case Description: This test case ensures that learners receive timely notifications about upcoming courses or significant milestones within their chosen learning paths.

Test Suite: Learning Paths Feature Test Suite

Test Priority: Medium

Preconditions:

* Login as a learner and select a learning path.

* Enroll in a course with future start dates or milestones within that learning path.

Test Data: A learning path with upcoming courses or milestones.

Test Steps:

1. Verify that notification settings are enabled for the learner account.

2. Observe inbox (likely within the platform) for notifications regarding upcoming courses or milestones in the chosen learning path.

3. Confirm that the notification provides relevant details about the upcoming event.

Postconditions:

* Learner remains on the learning path page.

Expected Result: The platform delivers timely notifications to learners about upcoming courses or milestones within their learning paths, ensuring they are aware of important dates and deadlines.

Severity: Minor

Priority: Medium

Type of Testing: Functional / Non-functional Testing (Notification Functionality)

Test Case Approach: 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:

## Test Cases for Attendance Tracking Feature

Project Context: Online Learning Platform

Feature: Attendance Tracking

User Story: As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
TC_ATT01	Verify that attendance is automatically recorded during live sessions.	This test case verifies that the system automatically records learner attendance when they join and remain present during a live session.	Attendance Tracking	High	- Instructor has created a live session. - System's clock is synchronized.	- List of learners registered for the session	1. Instructor starts the live session. 2. Learners join the session. 3. Learners remain active (participate in chat, ask questions) during the session. 4. Session ends.	- Attendance report is generated for the session.	Attendance records for all participants are correctly marked as present.	Blocker	High	System Testing, Functional Testing	Positive
TC_ATT02	Verify that instructors can view attendance reports for each session.	This test case ensures instructors can access and view attendance reports for specific sessions to track learner participation.	Attendance Tracking	High	- Instructor has a completed live session. - Attendance tracking is enabled.	- Attendance report data for the session	1. Instructor logs into the platform. 2. Navigate to the "Sessions" section. 3. Select the desired session. 4. Access the "Attendance Report" for the selected session.	- Attendance report is displayed in an organized manner.	The attendance report clearly shows learner names and their attendance status (present/absent) for the session.	Critical	High	System Testing, Functional Testing	Positive
TC_ATT03	Verify that learners can see their attendance records.	This test case verifies learners can access their own attendance history and view their participation record.	Attendance Tracking	Medium	- Learner is registered for a live session. - Attendance tracking is enabled.	- Attendance data for the learner	1. Learner

logs into the platform. 2. Navigate to the "My Profile" or "Account" section. 3. Locate the "Attendance" or "Session History" section. 4. View attendance records for past sessions. | - Attendance records are displayed in chronological order. | The learner's attendance history accurately reflects their presence or absence in each session. | Major | Medium | System Testing, Functional Testing | Positive |
| TC_ATT04 | Verify that attendance data is integrated with participation metrics. | This test case checks whether the system integrates attendance data with other participation metrics like question responses or forum engagement. | Attendance Tracking | Medium | - Learner attends a live session. - Learner actively participates in the session. | - Attendance and participation data during the session | 1. Instructor starts a live session. 2. Learner attends the session and participates actively. 3. Instructor accesses the attendance report for the session. | - Participation data is associated with attendance records | The attendance report should show learner participation metrics alongside their attendance status. | Important | Medium | System Testing, Functional Testing | Positive |
| TC_ATT05 | Verify that notifications are sent for missed sessions. | This test case confirms that the system sends notifications to learners when they miss a live session. | Attendance Tracking | High | - Instructor schedules a live session. - Notification settings are enabled for learners. | - List of learners registered for the session. | 1. Instructor schedules a live session and selects "Send notifications for missed sessions." 2. One or more learners do not attend the live session. 3. Learners should receive notifications about missing the session. | - Learners are notified about the missed session. | Learners should receive timely notifications regarding their missed attendance. | Critical | High | System Testing, Functional Testing | Positive

Note: This is a starting point for test case creation. Based on the platform's specific design and functionalities, more test cases may be required to cover all possible scenarios and edge cases.

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 five detailed test cases for the provided user story:

## Test Case 1

Test Case ID: TC_CLRM_001

Test Case Title: Verify that content creators can assign licenses to resources.

Test Case Description: This test case verifies the functionality allowing content creators to select and assign appropriate licenses to uploaded educational resources.

Test Suite: Content Licensing & Rights Management

Test Priority: High

Preconditions:

* A valid user account with content creator role is logged in.

* An educational resource (e.g., document, video, audio file) is uploaded to the platform.

Test Data:

* Different types of content licenses (e.g., Creative Commons, Attribution-NonCommercial, All Rights Reserved).

Test Steps:

1. Navigate to the dashboard of the logged-in content creator.

2. Select the uploaded resource.

3. Access the resource editing section.

4. Choose a content license from the provided options.

5. Save the changes to the resource.

Postconditions:

* The selected content license is displayed on the resource details page.

Expected Result: The content creator can successfully assign a content license to the uploaded resource.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TC_CLRM_002

Test Case Title: Verify that usage rights and permissions are clearly defined for each license.

Test Case Description: This test case ensures that the platform clearly outlines the usage rights and permissions associated with each selected content license.

Test Suite: Content Licensing & Rights Management

Test Priority: High

Preconditions:

* A valid user account with learner role is logged in.

* A resource with a content license is available.

Test Data:

* Different content licenses with varying usage rights and permissions.

Test Steps:

1. Access the resource page.

2. Click on the "License Information" link (or similar).

3. Read the description of the selected content license.

4. Identify the permitted and restricted uses outlined in the license terms.

Postconditions:

* The user understands the license terms and restrictions.

Expected Result: The platform clearly describes the permitted and restricted uses of each content license, enabling learners to understand their rights and obligations.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TC_CLRM_003

Test Case Title: Verify that content creators can set restrictions on usage and distribution.

Test Case Description: This test case confirms the capability of content creators to customize usage restrictions for their uploaded resources.

Test Suite: Content Licensing & Rights Management

Test Priority: High

Preconditions:

* A valid user account with content creator role is logged in.
* A new resource is prepared for uploading.

Test Data:

* Diverse usage restriction options (e.g., limit downloads, prohibit commercial use, restrict sharing).

Test Steps:

1. Upload the new resource.
2. Navigate to the resource editing section.

3. Select specific usage restrictions from the provided list.

4. Save the changes to the resource.

Postconditions:

* The applied usage restrictions are confirmed on the resource page.

Expected Result: Content creators can successfully set custom usage restrictions for their resources.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TC_CLRM_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 are notified about the content licensing terms associated with the resources they access.

Test Suite: Content Licensing & Rights Management

Test Priority: Medium

Preconditions:

* A valid user account with instructor role is logged in.

* A valid user account with learner role is logged in.

* Resources with different content licenses are accessible.

Test Data:

* Resources using various content licenses.

Test Steps:

1. **Instructor:** Access a resource and review its details.
2. **Learner:** Access a resource and review its details.

Postconditions:

* Both the instructor and learner see clear information about the content license associated with the resource.

Expected Result: The platform effectively communicates content licensing information to both instructors and learners.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: TC_CLRM_005

Test Case Title: Verify that notifications are sent for license expiration or renewal.

Test Case Description: This test case validates the functionality of sending notifications to content creators when their licenses are expiring or require renewal.

Test Suite: Content Licensing & Rights Management

Test Priority: High

Preconditions:

* A valid user account with content creator role is logged in.

* A resource with a content license that has a defined expiration date is uploaded.

Test Data:

 * A future date set as the license expiration date.

Test Steps:

1. Upload the resource with an expiring license.

2. Monitor the content creator's inbox for notifications closer to the expiration date.

Postconditions:

* The content creator receives notifications regarding the license expiration.

Expected Result: The platform sends timely notifications to content creators about expiring licenses, ensuring content compliance.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

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:

## Test Cases for Content Versioning Feature

Project Name: Online Learning Platform

User Story: As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

Test Suite: Content Versioning

Test Case ID: VCTV_001

Test Case Title: Verify that instructors can create and save multiple versions of content.

Test Case Description: This test case verifies the ability of instructors to create and save multiple versions of course materials.

Test Priority: High

Preconditions:

* Instructor account is created and logged in.
* A new course is created.

Test Data: Different versions of course content (e.g., text, video, PDF).

Test Steps:

1. Navigate to the course content editing page.
2. Update the course content with new information.
3. Click the "Save Version" button to create a new version of the content.
4. Repeat steps 2 and 3 with different content changes to create 3 distinct versions.
5. Verify that the content versions are listed in chronological order with identifiable version labels.

Postconditions:

* Updated content is saved as different versions.
* Version history is accessible.

Expected Result: The instructor can successfully create and save multiple versions of the course content, with each version identifiable and accessible.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: VCTV_002

Test Case Title: Verify that version history is maintained and accessible.

Test Case Description: This test case verifies that the platform accurately records and displays the history of content changes.

Test Priority: High

Preconditions:

* VCTV_001 test case is preformed successfully.

Test Data: Reverted to previous versions of content.

Test Steps:

1. Navigate to the version history of the course content.
2. Review the list of saved versions, including their timestamps and labels.
3. Select a previous version and click "Restore" to revert the content to that version.
4. Verify that the content successfully reverts to the selected previous version.
5. Repeat steps 2-4 with different previous versions.

Postconditions:

* Content successfully reverted to a previous version.

Expected Result:
The version history accurately reflects all saved versions of the course content. Instructors can easily access and restore previous versions.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: VCTV_003

Test Case Title: Verify that instructors can compare different versions of content.

Test Case Description: This test case ensures the functionality for instructors to compare different versions of content side-by-side.

Test Priority: Medium

Preconditions:

* VCTV_001 test case is preformed successfully.

Test Data: Updated and previous versions of content.

Test Steps:

1. Navigate to the version history page.
2. Select two different versions of the content for comparison.
3. Initiate the content comparison feature.
4. Verify that the content appears side-by-side, highlighting the differences between the two versions clearly.

5. Verify that changes are categorized by type (e.g., text additions, deletions, formatting changes).

Postconditions:

* Content comparison functionality reveals differences.

Expected Result: Instructors can easily compare different versions of their content and identify specific changes made between versions.

Test Case ID: VCTV_004

Test Case Title: Verify that previous versions can be restored if needed.

Test Case Description: This test case verifies the functionality to restore previous versions of content.

Test Priority: High

Preconditions:

* VCTV_001 test case is preformed successfully.

Test Data: A previous version of content with specific changes.

Test Steps:

1. Intentionally make changes to the current content version.
2. Navigate to the version history.
3. Select a previous version before the intentional changes were made.
4. Click the "Restore" button to revert the content to the selected previous version.
5. Verify that the content is successfully restored to the chosen previous version, undoing the intentional changes.

Postconditions:

* Content is restored to the selected previous version.

Expected Result: Restoring a previous version accurately recovers the content to its state at that point in time.

Severity: Blocker

Priority: High

Type of Testing: Functional / Regression Testing

Test Case Approach: Negative

Test Case ID: VCTV_005

Test Case Title: Verify that notifications are sent for significant content updates.

Test Case Description: This test case ensures that notifications are sent to relevant stakeholders when significant content changes occur.

Test Priority: Medium

Preconditions:

* A course is created with content.
* VCTV_001 test case is preformed successfully.

Test Data: Substantial content changes (e.g., addition of a new module, major revision of existing content).

Test Steps:

1. Make significant changes to a considerable portion of the course content.
2. Save the new version.
3. Monitor the notification system (email, in-app) for announcements regarding the content update.
4. Verify the notification includes a brief summary of the changes made and the date/time of the update.

Postconditions:

* Notification system records the significant content change.

Expected Result: An appropriate notification is sent to designated stakeholders (e.g., course instructors, enrolled students) alerting them to the significant content update.

Severity: Major

Priority: Medium

Type of Testing: Functional / System Testing

Test Case Approach: Positive

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:

## Test Cases for Online Learning Platform API Integration

User Story: As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

Test Suite: API Integration

Here are 5 detailed test cases:

Test Case ID: TC-API-001

Test Case Title: Verify that API endpoints for key functionalities are available.

Test Case Description: This test case verifies the existence and accessibility of API endpoints for essential platform functionalities like user registration, course enrollment, submission of assignments, and fetching course materials.

Test Priority: High

Preconditions:

* API server is running and accessible.

* Developer account with API access is available.

Test Data: API endpoint URLs based on documented functionalities.

Test Steps:

1. Use a REST client (e.g., Postman) to send GET requests to documented API endpoints for user registration, course enrollment, assignment submission, and course content retrieval.

2. Verify that each request returns a valid HTTP status code indicating success (e.g., 200 OK).

3. Inspect the response payload to ensure it contains the expected data structure and format.

4. Repeat for different operations, including POST, PUT, and DELETE requests as applicable.

Postconditions: No specific postconditions.

Expected Result: All key functionalities have accessible API endpoints returning expected responses with valid status codes and data format.

Severity: Critical

Priority: High

Type of Testing: Functional, API Testing

Test Case ID: TC-API-002

Test Case Title: Verify that comprehensive API documentation is provided.

Test Case Description: This test case assesses the completeness and clarity of the API documentation.

Test Priority: Medium

Preconditions: API documentation is accessible.

Test Data: A list of expected API endpoints and their functionalities based on user story requirements.

Test Steps:

1. Review the API documentation for availability and organization of information.

2. Check if the documentation includes detailed descriptions of each API endpoint, including URL, method, parameters, request body, and response structure.

3. Verify that code examples are provided in multiple programming languages.

4. Assess the documentation for completeness, accuracy, and readability.

Postconditions: No specific postconditions.

Expected Result: API documentation is comprehensive, easy to understand, and accurately describes the functionality and usage of all API endpoints.

Severity: Major

Priority: Medium

Type of Testing: Functional, Documentation Testing

Test Case ID: TC-API-003

Test Case Title: Verify secure access protocols for API authentication.

Test Case Description: This test case ensures the platform implements robust security measures for API authentication.

Test Priority: High

Preconditions:

* API server is running.

* Developer account with API access is available.

Test Data: Valid API keys or authentication tokens.

Test Steps:

1. Attempt to access protected API endpoints without providing valid authentication credentials.

2. Verify that access is denied with an appropriate HTTP status code (e.g., 401 Unauthorized).

3. Provide valid API keys or authentication tokens for accessing the same endpoints.

4. Verify successful access with an appropriate HTTP status code (e.g., 200 OK) and the expected data.

5. Test different authentication mechanisms (e.g., OAuth, API keys) as implemented.

Postconditions: No specific postconditions.

Expected Result: API access is restricted to authorized users with valid authentication credentials. Unauthorized access attempts are promptly rejected with appropriate error codes.

Severity: Critical

Priority: High

Type of Testing: Security Testing

Test Case ID: TC-API-004

Test Case Title: Verify rate limiting and monitoring features for API usage.

Test Case Description: This test case evaluates the effectiveness of rate limiting and monitoring for API calls.

Test Priority: Medium

Preconditions:

* API server is running.

* Monitoring tools are configured and accessible.

Test Data:

Scripts to simulate high API call volumes.

Test Steps:

1. Send a predefined number of API requests within a short timeframe (e.g., 100 requests in 5 seconds).

2. Verify that the API server enforces rate limiting by returning appropriate error codes (e.g., 429 Too Many Requests) after exceeding the allowed limit.

3. Monitor API usage logs and metrics for accurate tracking of request volume, frequency, and potential anomalies.

Postconditions: No specific postconditions.

Expected Result: API rate limiting is implemented effectively, preventing abuse and ensuring stable service availability.

Monitoring tools accurately capture and display API usage data.

Severity: Major

Priority: Medium

Type of Testing: Performance Testing, Non-Functional Testing

Test Case ID: TC-API-005

Test Case Title: Verify that third-party applications can successfully authenticate and access APIs.

Test Case Description: This test case confirms the seamless integration of third-party applications with the platform's APIs.

Test Priority: High

Preconditions:

* A test third-party application is developed with integration to the platform's APIs.
* Valid API keys or authentication tokens are generated for the test application.

Test Data: API endpoints and functionalities to be accessed by the test application.

Test Steps:

1. Configure the test application to use the generated API keys or authentication tokens.
2. Execute test scenarios through the application to interact with platform APIs (e.g., registering a user, enrolling them in a course, accessing course materials).

3. Verify that the application successfully authenticates with the platform's APIs.
4. Inspect the responses received from the APIs to ensure data accuracy and consistency.

Postconditions: No specific postconditions.

Expected Result: The test application seamlessly integrates with the platform's APIs by authenticating successfully and retrieving all required data.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, API Testing

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:

## Test Cases for Data Analytics and Insights Feature

Project Name: Online Learning Platform

Feature Name: Data Analytics and Insights

User Story: As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

### Test Case 1

Test Case ID: TC_DATA_001

Test Case Title: Verify that data visualization tools are available for learner performance analysis.

Test Case Description: This test case verifies the availability of data visualization tools within the administrator dashboard to analyze learner performance metrics, such as quiz scores, completion rates, and forum engagement.

Test Suite: Data Analytics Suite

Test Priority: High

Preconditions:

* An administrator account is logged in.

* At least one course is published on the platform with enrolled learners.

Test Data: No test data needed

Test Steps:

1. Navigate to the administrator dashboard.
2. Locate the "Data Analytics" or similar section.
3. Verify that the following data visualization tools are accessible:
    * Charts and graphs for displaying learner performance trends
    * Filtering and sorting options for data analysis
    * Interactive dashboards for drill-down analysis

Postconditions:

* Administrator dashboard remains accessible.
* Data visualization tools are closed/minimized.

Expected Result:
The administrator should be able to access and utilize the specified data visualization tools to analyze learner performance.

Severity: Critical
Priority: High
Type of Testing: Functional Testing, System Testing

### Test Case 2

Test Case ID: TC_DATA_002

Test Case Title: Verify that reports can be generated for learner performance metrics.

Test Case Description: This test case verifies the ability to generate reports for specific learner performance metrics, allowing administrators to download and analyze data in a comprehensive format.

Test Suite: Data Analytics Suite

Test Priority: High

Preconditions:

* An administrator account is logged in.
* At least one course is published on the platform with enrolled learners.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Data Analytics" or similar section within the administrator dashboard.
2. Choose a desired report type, such as "Course Completion Rates" or "Quiz Scores".
3. Select a specific time period for the report.
4. Select specific courses or learners to include in the report (optional).
5. Generate the report and verify that it includes relevant learner performance data.

Postconditions:

* The generated report is saved to the desired location (e.g., local computer).

* The administrator dashboard remains accessible.

Expected Result:

The platform should successfully generate comprehensive reports containing chosen learner performance metrics for the specified time period.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

### Test Case 3

Test Case ID: TC_DATA_003

Test Case Title: Verify that real-time analytics provide insights into learner engagement and outcomes.

Test Case Description: This test case verifies the functionality of real-time analytics to display immediate insights into learner engagement and outcomes.

Test Suite: Data Analytics Suite

Test Priority: High

Preconditions:

* An administrator account is logged in.

* At least one live session or course activity is taking place with enrolled learners.

Test Data: No test data needed

Test Steps:

1. Access the real-time analytics dashboard within the administrator section.
2. Monitor key engagement metrics in real time, such as number of active learners, forum post count, and resource downloads.
3. Observe the display of learner outcome metrics, such as quiz completion rates and average quiz scores, as learners participate in activities.

Postconditions:

* The administrator dashboard remains accessible.

Expected Result:
The real-time analytics dashboard should accurately display current learner engagement and outcome data, providing ongoing insights into platform performance.

Severity: Critical
Priority: High
Type of Testing: Functional Testing, System Testing

(Continue creating similar test cases for the remaining acceptance criteria: Data export and notification system)

Note: This is a starting point for your test cases. You should adapt them based on the specific functionalities and requirements of your Online Learning Platform. Ensure to include detailed descriptions, specific data values, and clear expected results for each test case.

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:

## Online Learning Platform Test Cases - User Roles and Permissions

Here are 5 detailed test cases for the "User Roles and Permissions" feature based on the provided user story and acceptance criteria:

Test Case 1:

* **Test Case ID:** TC_URP_001
* **Test Case Title:** Verify that administrators can create new user roles
* **Test Case Description:** This test case aims to verify that administrators have the functionality to create new user roles with specific permissions.
* **Test Suite:** User Roles and Permissions
* **Test Priority:** High
* **Preconditions:**
    * An administrator account is logged in.
* **Test Data:**
    * New role name: "InstructorAssistant"
* **Test Steps:**
    1. Navigate to the User Roles and Permissions section.
    2. Click on the "Create New Role" button.
    3. Enter the name "InstructorAssistant" for the new role.
    4. Assign appropriate permissions to the role (e.g., access to student grades, create assignments).
    5. Click on the "Save" button to create the role.
* **Postconditions:**
    * The "InstructorAssistant" role is listed in the user roles table.
* **Expected Result:** The new role "InstructorAssistant" is successfully created with the assigned permissions.

Test Case 2:

* **Test Case ID:** TC_URP_002

* **Test Case Title:** Verify that permissions are customizable for each role

* **Test Case Description:** This test case aims to verify that administrators can customize permissions for different user roles.

* **Test Suite:** User Roles and Permissions

* **Test Priority:** High

* **Preconditions:**

    * An administrator account is logged in.

* **Test Data:**

    * Existing role: "Student"

* **Test Steps:**

    1. Navigate to the User Roles and Permissions section.

    2. Select the "Student" role from the list of roles.

    3. Modify the permissions for the "Student" role (e.g., allow access to discussion forums).

    4. Click on the "Save" button to apply the changes.

* **Postconditions:**

    * The permissions for the "Student" role are updated in the system.

* **Expected Result:** The permissions for the "Student" role are successfully modified.

Test Case 3:

* **Test Case ID:** TC_URP_003

* **Test Case Title:** Verify that users are restricted to functionalities based on their assigned role

* **Test Case Description:** This test case focuses on verifying that users can only access functionalities

permitted by their assigned role.

* **Test Suite:** User Roles and Permissions

* **Test Priority:** High

* **Preconditions:**

    * A user account with the "Student" role is logged in.

* **Test Data:**

    * Module requiring administrator permissions.

* **Test Steps:**

    1. Navigate to the module requiring administrator permissions.

    2. Attempt to perform an action restricted to administrators (e.g., create a new course).

* **Postconditions:**

    * The user is presented with an error message or denied access.

* **Expected Result:** The user is restricted from accessing the functionality and receives appropriate feedback.

Test Case 4:

* **Test Case ID:** TC_URP_004

* **Test Case Title:** Verify that changes to roles and permissions are logged

* **Test Case Description:** This test case aims to verify that changes made to user roles and permissions are correctly logged within the system.

* **Test Suite:** User Roles and Permissions

* **Test Priority:** Medium

* **Preconditions:**

    * An administrator account is logged in.

* **Test Data:**

    * An existing role.

* **Test Steps:**

   1. Modify the existing role's permissions.

   2. Log off and back in as the administrator.

   3. Navigate to the role logs section.

* **Postconditions:**

   * The role logs show the recent changes made to the role's permissions.

* **Expected Result:** The log accurately records the user, date, time, and nature of the permission changes.

Test Case 5:

* **Test Case ID:** TC_ UR P_005

* **Test Case Title:** Verify that users are notified of their assigned roles and permissions

* **Test Case Description:** This test case verifies that new users receive notification about their assigned roles and associated permissions.

* **Test Suite:** User Roles and Permissions

* **Test Priority:** Medium

* **Preconditions:**

   * A new user account is created.

* **Test Data:**

   * New user account with assigned "Student" role.

* **Test Steps:**

   1. A new user account is created with a specific role (e.g., "Student").

   2. The new user logs into the platform.

   3. The user interfaces with elements related to their assigned role.

* **Postconditions:**

   * The user receives clear notification about their role and permissions.

* **Expected Result:** The new user receives an onboarding message or notification outlining their assigned

role "Student" and the corresponding permissions within the platform.

Testing these user roles and permission features will ensure a secure and compliant online learning platform that provides controlled access to functionalities based on user roles, ultimately enhancing the user experience.

--

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:

## Online Learning Platform: User Feedback and Rating System Test Cases

Test Case ID: TC_USER_FEEDBACK_001

Test Case Title: Verify that learners can rate courses on a scale.

Test Case Description: This test case verifies the functionality of rating courses on a scale, ensuring learners can choose from the available options and submit their ratings.

Test Suite: User Feedback & Rating System

Test Priority: High

Preconditions:

* User is logged into the platform.

* User has completed a course.

* The course contains a rating system.

Test Data: Numeric rating values (e.g., 1-5 stars)

Test Steps:

1. Navigate to the completed course's page.

2. Click on the "Rate Course" button.

3. Select a desired rating from the available options (e.g., a 4-star rating).

4. Click on the "Submit Rating" button.

5. Verify that the selected rating is displayed next to the course title.

Postconditions:

* Rating is persisted in the database.

* User is navigated back to the course page.

Expected Result: The course rating is successfully updated to the selected value.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_USER_FEEDBACK_002

Test Case Title: Verify that learners can provide detailed feedback on courses.

Test Case Description: This test case verifies the functionality of the feedback form, ensuring users can input text feedback and submit it.

Test Suite: User Feedback & Rating System

Test Priority: High

Preconditions:

* User is logged into the platform.

* User has completed a course.

* The course has a feedback form.

Test Data: Sample feedback comments (positive, negative, neutral)

Test Steps:

1. Navigate to the completed course's page.

2. Click on the "Provide Feedback" button.

3. Type a feedback comment in the provided text box.

4. Click on the "Submit Feedback" button.

5. Verify that the feedback comment is displayed on the course page.

Postconditions:

* Feedback is persisted in the database.

* User is navigated back to the course page.

Expected Result: The feedback comment is successfully submitted and displayed.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_USER_FEEDBACK_003

Test Case Title: Verify that instructors can view feedback submitted by learners.

Test Case Description: This test case verifies the instructor's ability to view all feedback comments submitted by learners for their courses.

Test Suite: User Feedback & Rating System

Test Priority: High

Preconditions:

* Instructor is logged into the platform.

* There are feedback comments submitted for one of the instructor's courses.

Test Data: Relevant course information

Test Steps:

1. Navigate to the "My Courses" section.

2. Select the course for which learner feedback exists.

3. Navigate to the "Feedback" tab.

4. Verify that all submitted feedback comments are displayed.

Postconditions:

* Instructor remains on the feedback tab.

Expected Result: All learner feedback for the selected course is visible to the instructor.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_USER_FEEDBACK_004

Test Case Title: Verify that anonymous feedback options are available.

Test Case Description: This test case verifies the functionality of anonymous feedback, ensuring learners can choose to provide feedback without being identified.

Test Suite: User Feedback & Rating System

Test Priority: High

Preconditions:

* User is not logged into the platform.

* User has completed a course.

* The course has a feedback form.

Test Data: Sample feedback comments

Test Steps:

1. Navigate to the completed course's page.

2. Click on the "Provide Feedback" button.

3. Ensure that an option to submit feedback anonymously is available.

4. Type a feedback comment in the provided text box and select the "Anonymous" option.

5. Click on the "Submit Feedback" button.

6. Verify that the feedback comment is submitted without requiring login information.

Postconditions:

* Feedback is persisted in the database.

* User remains on the course page.

Expected Result: The anonymous feedback is successfully submitted.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_USER_FEEDBACK_005

Test Case Title: Verify that aggregated ratings are displayed publicly.

Test Case Description: This test case verifies the display of aggregated ratings for courses, ensuring the public sees the average rating and the number of ratings.

Test Suite: User Feedback & Rating System

Test Priority: Medium

Preconditions:

* Several learners have rated a course.

Test Data: Course with multiple ratings

Test Steps:

1. Navigate to the course page.

2. Locate the aggregated rating section.

3. Verify that the average rating value is displayed accurately.

4. Verify that the total number of ratings is displayed correctly.

Postconditions:

* User remains on the course page.

Expected Result: Aggregated ratings are displayed publicly on the course page, showing the average rating and the number of ratings.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

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:

## Test Cases for Social Media Integration Feature

Project Name: Online Learning Platform

Feature Name: Social Media Integration

User Story Title: As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

Test Case ID: LPL-SMI-TC01

Test Case Title: Verify that learners can connect their Facebook account.

Test Case Description: This test case aims to verify the functionality of connecting a Facebook account to the learning platform. It will ensure the user interface for connecting Facebook is functional and handles successful and unsuccessful connections.

Test Suite: User Authentication and Social Media Integration

Test Priority: High

Preconditions:

- A registered learner account.
- Internet access and functional Facebook account.

Test Data: Valid Facebook email address and password.

Test Steps:

1. Log in to the learning platform as a registered learner.
2. Navigate to the "Settings" or "Profile" section.
3. Locate the "Social Media" or "Connect Account" section.
4. Click on the Facebook icon or button.

5. Authorize the platform to access user's Facebook data.

Postconditions:

- Learner should be successfully connected to their Facebook account.
- A confirmation message should display indicating successful connection.
- Learner should see their Facebook profile information reflected in the platform's settings.

Expected Result: Successful connection to the Facebook account.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: LPL-SMI-TC02

Test Case Title: Verify that learners can share course completion achievements on Facebook.

Test Case Description: This test case will verify the functionality of sharing course completion achievements on Facebook. It will ensure the user interface for sharing is accessible and shares the correct information.

Test Suite: User Feature - Achievement Sharing

Test Priority: High

Preconditions:

- A registered learner account.

- Successful Facebook connection as per TC01.

- Completion of a course.

Test Data: Completed course with title, ID, and outcome.

Test Steps:

1. Navigate to the learner's profile page after course completion.

2. Locate the "Share Achievement" button or option related to the completed course.

3. Select Facebook as the sharing platform.

4. Review and confirm the shared content, including course title and completion date.

Postconditions:

- Learner's Facebook feed should display a post about the course completion.

- The post should include the correct course title and completion date.

- The learner should receive a confirmation message on the platform.

Expected Result: Successful sharing of the course completion achievement on Facebook.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: LPL-SMI-TC03

Test Case Title: Verify that learners can control privacy settings for shared content.

Test Case Description: This test case will verify the functionality of the privacy settings for shared content on social media. It will ensure learners can choose who sees their shared achievements.

Test Suite: User Privacy and Control

Test Priority: Medium

Preconditions:

- A registered learner account.
- Successful Facebook connection as per TC01.
- Completion of a course.

Test Data: Different privacy settings on Facebook (Public, Friends Only, Specific People).

Test Steps:

1. Navigate to learner's profile page after course completion.
2. Locate the "Share Achievement" button or option related to the completed course.
3. Select Facebook as the sharing platform.
4. Adjust the privacy settings for the shared content using available options.
5. Review and confirm the selected privacy settings.

Postconditions:

- The shared content on Facebook reflects the chosen privacy settings.

Expected Result:

Learners can successfully change the privacy settings for shared content and it is reflected on Facebook.

Severity: Major

Priority: Medium

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: LPL-SMI-TC04

Test Case Title: Verify that learners cannot share achievements without connecting a social media account.

Test Case Description: This test case will verify that learners are prevented from sharing achievements on social media until they have connected an account.

Test Suite: User Feature - Social Media Access Control

Test Priority: Medium

Preconditions:

- A registered learner account.

Test Data: None

Test Steps:

1. Navigate to the learner's profile page after course completion.
2. Attempt to share the course completion achievement on Facebook without connecting an account.

Postconditions:

- A warning message or user prompt should appear, stating that a social media account connection is required for sharing.

Expected Result: Learners are barred from sharing achievements until a social media account is connected.

Severity: Minor

Priority: Medium

Type of Testing: System Testing, Functional Testing

Test Case Approach: Negative

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: Activity Tracking Test Cases

Here are 5 test cases for the "Activity Tracking" feature, following the provided template:

Test Case 1:

* **Test Case ID:** TC_ACTVTY_001

* **Test Case Title:** Verify that the app records distance for a running activity.

* **Test Case Description:** This test case verifies that the app accurately records the distance covered during a simulated running activity.

* **Test Suite:** Activity Tracking

* **Test Priority:** High

* **Preconditions:**

    * User has created a profile in the app.

    * User has enabled location services for the app

* **Test Data:**

    * Distance entered during activity: 5 kilometers

* **Test Steps:**

    * Start a running activity in the app.

    * Simulate running for 5 kilometers (e.g., using a treadmill).

    * Stop the running activity.

    * View the activity summary.

* **Postconditions:**

    * The recorded distance in the activity summary should be 5 kilometers.

* **Expected Result:** The app accurately records the distance covered (5 kilometers).

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TC_ACTVTY_002

* **Test Case Title:** Verify that the app allows users to view activity summaries.

* **Test Case Description:** This test case ensures that users can access and view historical records of their activities, including distance, duration, pace, and calories burned.

* **Test Suite:** Activity Tracking

* **Test Priority:** High

* **Preconditions:**

    * User has recorded at least one activity in the app.

* **Test Data:**

* **Test Steps:**

    * Navigate to the "Activity" section of the app.

    * View a previously recorded activity.

  * Verify that the activity summary displays relevant information (distance, duration, pace, calories burned).
* **Postconditions:**
  * User can navigate back to the main screen.
* **Expected Result:** The app accurately displays the activity summary for the selected activity.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 3:
* **Test Case ID:** TC_ACTVTY_003
* **Test Case Title:** Verify that the app stores activity data for historical analysis.
* **Test Case Description:** This test case confirms that the app retains activity data for future retrieval and analysis.
* **Test Suite:** Activity Tracking
* **Test Priority:** Medium
* **Preconditions:**
  * User has recorded several activities over multiple days.
* **Test Data:**
* **Test Steps:**
  * Navigate to the "Activity" section of the app and view activity history.
  * Verify that activities from different days are displayed.
  * Select an activity from a previous day.

* **Postconditions:**
  * User can navigate back to the main screen.
* **Expected Result:** The app correctly stores and displays activity data from past dates.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TC_ACTVTY_004

* **Test Case Title:** Verify that the app allows users to pause and resume activity tracking.

* **Test Case Description:** This test case ensures that users can temporarily stop and later restart tracking their activity without data loss.

* **Test Suite:** Activity Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User has started a running activity.

* **Test Data:**

    * Duration of initial activity segment: 10 minutes

    * Duration of paused activity segment: 5 minutes

    * After resumption, duration of complete activity segment is 17 minutes

* **Test Steps:**

    * Start a running activity.

    * Pause the activity after 10 minutes.

    * Resume the activity after 5 minutes.

    * Stop the activity after a total of 17 minutes.

    * Review activity summary for total duration of 17 minutes.

* **Postconditions:**

* **Expected Result:** The app accurately records the total duration of the activity (17 minutes) despite the pause and resume actions.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_ACTVTY_005
* **Test Case Title:** Verify that the app provides notifications for activity milestones.
* **Test Case Description:** This test case confirms that users receive notifications when they achieve predefined activity milestones.
* **Test Suite:** Activity Tracking
* **Test Priority:** High
* **Preconditions:**
    * User sets a personalized milestone (e.g., walking 10,000 steps).
    * Push notifications are enabled in the app settings.
* **Test Data:**
    * Milestone: 10,000 steps
* **Test Steps:**
    * Set a milestone for 10,000 steps.
    * Simulate walking actions in the app until reaching 10,000 steps.
    * Verify a notification is received on reaching the milestone.
* **Postconditions:**
* **Expected Result:** The app delivers a notification to the user when they complete the 10,000-step milestone.
* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases cover the key acceptance criteria for the "Activity Tracking" feature.

--

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:

## Fitness Tracker App - GPS Tracking Test Cases

Test Case ID: GPS-001

Test Case Title: Verify that GPS tracking accurately records route data.

Test Case Description: This test case verifies that the app accurately records the user's route data while walking or running outdoors.

Test Suite: Fitness Tracker - GPS

Test Priority: High

Preconditions:

* User has enabled location services for the app.
* User is outdoors with clear GPS signal.

Test Data: No test data needed

Test Steps:

1. Start a new workout recording session with the GPS tracking feature enabled.

2. Walk or run along a predetermined route for a specified distance (e.g., 1 kilometer).

3. Stop the workout recording session.

4. View the recorded route on the app's map.

5. Compare the recorded route on the map with the actual route traveled.

Postconditions:

* The workout recording is saved.

Expected Result: The recorded route on the app should closely match the actual route traveled, with minimal deviations.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case ID: GPS-002
Test Case Title: Verify that users can view detailed maps and elevation profiles.
Test Case Description: This test case verifies that the app displays detailed maps and elevation profiles for recorded routes.

Test Suite: Fitness Tracker - GPS

Test Priority: High

Preconditions:

* Simulated GPS data with varying elevation changes.

* User has enabled the detail view for route data.

Test Data: No test data needed

Test Steps:

1. Perform a workout recording while simulating GPS data with significant elevation changes.

2. Stop the workout recording session and view the recorded route.

3. Check if the app displays a detailed map of the route.

4. Analyze the elevation profile displayed alongside the map.

5. Verify the accuracy and visual clarity of the elevation profile.

Postconditions:

* Route data is saved with elevation profile data.

Expected Result: The app should display a clear and accurate map and elevation profile for the recorded route, showcasing the changes in elevation during the activity.

Severity: Major

Priority: Medium

Type of Testing: System Testing, Functional Testing,

Test Case ID: GPS-003

Test Case Title: Verify that performance data includes location-based metrics.

Test Case Description: This test case checks if the GPS tracking feature includes relevant location-based data in performance metrics.

Test Suite: Fitness Tracker - GPS

Test Priority: High

Preconditions:

* User is outdoors with clear GPS signal
* User has started a GPS-enabled workout session

Test Data: No test data needed

Test Steps:

1. Start a workout recording session with GPS tracking enabled.
2. Complete the workout, ensuring the app records route data.
3. Stop the workout session and view the performance details.
4. Analyze the provided location-based metrics, such as distance traveled, average speed, maximum speed, and elevation gain/loss.

Postconditions:

* Workout data is saved.

Expected Result:
The app should accurately display location-based metrics related to the user's recorded workout. Dec

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case ID: GPS-004
Test Case Title: Verify that routes can be saved and revisited.
Test Case Description: This test case confirms that users can save their recorded routes and subsequently access them for review or planning future workouts.

Test Suite: Fitness Tracker - GPS

Test Priority: Medium

Preconditions:

* User has completed a workout with GPS tracking enabled.

Test Data: No test data needed

Test Steps:

1. Complete a workout recording with GPS tracking enabled. 2. Ensure the workout is saved successfully with route data.
3. Navigate to the saved routes section within the app.
4. Select the previously recorded route from the list.
5. Verify that the chosen route displays its detailed map and associated data.

Postconditions:

* The route is successfully loaded for viewing.

Expected Result: The app should enable users to save and revisit previously recorded routes, allowing them to analyze past performance and plan future workouts based on existing routes.

Severity: Minor

Priority: Low

Type of Testing: System Testing, Functional Testing,

Test Case ID: GPS-005

Test Case Title: Verify that notifications are provided for deviations from planned routes.

Test Case Description: This test case verifies that the app provides notifications when the user deviates from a pre-planned route.

Test Suite: Fitness Tracker - GPS

Test Priority: Medium

Preconditions:

* User has created a planned route within the app.
* User is outdoors with clear GPS signal.

Test Data: No test data needed

Test Steps:

1. Create a planned route within the app.
2. Start tracking a workout using GPS, following the predefined planned route.
3. Intentionally deviate from the planned route.
4. Observe the app's response to the deviation.

Postconditions:

* Notification is displayed indicating deviation from the planned route.

Expected Result: The app should provide a visual notification or alert when the user deviates from the planned route, allowing them to adjust their course or stay informed about changes in their intended path.

Severity: Minor

Priority: Low

Type of Testing: System Testing, Functional Testing

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:

## Fitness Tracker App - Heart Rate Monitoring Test Cases

Test Case ID: HRT-001

Test Case Title: Verify that heart rate data is recorded in real-time.

Test Case Description: This test case aims to confirm that the app accurately captures and displays heart rate data in real-time while using a compatible wearable device or the built-in sensor.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

* User has a compatible wearable device or the app utilizes built-in sensor capabilities.
* User has granted permission for the app to access heart rate data.

Test Data: N/A

Test Steps:

1. Launch the Fitness Tracker app and ensure the heart rate monitoring feature is enabled.

2. Begin physical activity that will elevate heart rate.

3. Observe the real-time heart rate display within the app.

4. Verify that the displayed heart rate value fluctuates in a realistic manner corresponding to the user's activity level.

5. Monitor for a consistent update rate of the heart rate display.

6. Stop the physical activity and observe the heart rate gradually return to a resting state.

Postconditions:

* Heart rate monitoring feature is disabled.

Expected Result: The app accurately and consistently records and displays the user's real-time heart rate data throughout the physical activity.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: HRT-002

Test Case Title: Verify that users can set and monitor target heart rate zones.

Test Case Description: This test case verifies the functionality of setting custom target heart rate zones and the app's ability to monitor whether the user's heart rate falls within those zones.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

* User is logged into their account.
* Access to settings menu is available.

Test Data:

* Two different target heart rate ranges (e.g., Zone 1: 120-140 bpm, Zone 2: 140-160 bpm).

Test Steps:

1. Navigate to the settings menu within the app.
2. Locate the "Heart Rate Zones" setting and select it.
3. Enter the desired heart rate range for Zone 1 (e.g., 120-140 bpm).
4. Save the changes and confirm the target zone is displayed correctly.
5. Repeat steps 3-4 for Zone 2 (e.g., 140-160 bpm).
6. Start physical activity that will elevate heart rate into one of the defined zones.
7. Observe the app's display of which zone the user's heart rate falls into.

Postconditions:

* The target heart rate zone settings are saved and accessible.

Expected Result: The app accurately displays the user's current heart rate zone based on the set target heart rate ranges.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: HRT-003

Test Case Title: Verify that historical heart rate data is stored and viewable.

Test Case Description: This test case confirms that the app records and allows users to view historical heart rate data over a specified period.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

* User has used the heart rate monitoring feature at different times.

Test Data: N/A

Test Steps:

1. Launch the app and navigate to the heart rate data view.

2. Verify that historical heart rate data is displayed, showing the trend over time.

3. Select a specific date range within the historical data.

4. Observe the chart and data points for that date range.

5. Identify individual heart rate readings for specific timestamps.

Postconditions:

* App remains in the historical heart rate data view.

Expected Result: The app accurately displays historically recorded heart rate data for the selected date range, allowing the user to track their heart rate trends and progress.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Data Persistence Testing

Test Case ID: HRT-004

Test Case Title: Verify that alerts are provided for abnormal heart rate readings.

Test Case Description: This test case examines the app's functionality in detecting and alerting users about abnormally high or low heart rate readings, promoting user safety and awareness.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

* User has set heart rate alert thresholds in the app.

* User has enabled alert notifications.

* User's heart rate can be artificially elevated or lowered to trigger alerts. (May require a simulation or external input)

Test Data: N/A

Test Steps:

1. Customize heart rate alert thresholds within the app (e.g., high heart rate alert at 180 bpm, low heart rate alert at 40 bpm).
2. Ensure that the app is permitted to send notifications.
3. Simulate a heart rate reading that exceeds the high alert threshold.
4. Observe if the app delivers the corresponding high heart rate alert notification.
5. Manually lower heart rate below the low alert threshold.
6. Verify the app sends a low heart rate alert notification.

Postconditions:

* Alerts are disabled.

Expected Result: The app correctly identifies and promptly notifies the user when heart rate readings surpass the defined high and low alert thresholds.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Performance Testing

Test Case Approach: Positive, Negative

Test Case ID: HRT-005

Test Case Title: Verify compatibility with various wearable devices.

Test Case Description: This test case ensures that the app seamlessly integrates with different popular wearable devices and accurately retrieves heart rate data.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

* Have access to at least two different compatible wearable devices (e.g., smartwatches, fitness trackers).
* User accounts on the app and wearable devices are set up and synchronized.

Test Data: N/A

Test Steps:

1. Pair the first wearable device with the Fitness Tracker app.

2. Verify successful connection and heart rate data synchronization.
3. Disconnect the first device and connect the second wearable device.
4. Observe the app updating to display heart rate data from the new device.
5. Repeate for any additional supported wearable devices.
6. Compare heart rate readings obtained from different devices during identical activities, ensuring consistency.

Postconditions:

* The app displays heart rate data from the last connected device.

Expected Result: The app establishes stable connections with supported wearable devices, accurately retrieves heart rate data, and displays it seamlessly throughout the user experience.

Severity: Major

Priority: Medium

Type of Testing: Integration Testing, Compatibility Testing

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:

## Test Cases for Calorie Tracking Feature in Fitness Tracker App

Test Case ID: CTT_01

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 daily food intake and calorie consumption in the app. It checks if users can input food items and their corresponding calories accurately.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

* User has successfully logged into the app.

* The user is on the "Food Log" screen.

Test Data:
* List of common food items with their calorie information
* User-specific calorie goal (can be set arbitrarily for testing)

Test Steps:
1. Tap on the "Add Food" button on the "Food Log" screen.
2. Search for a food item in the database (e.g., "apple").
3. Select the correct food item from the search results.
4. Enter the quantity consumed of the food item (e.g., 1 apple).
5. Tap on "Save" to log the food item.

Postconditions:
* The food item is displayed in the user's food log with the correct calorie information.
* The total calorie count for the day is updated accordingly.

Expected Result: The app successfully logs the food item and calculates the total calorie consumption for the day.

Severity: Critical
Priority: High
Type of Testing: Functional Testing, User Acceptance Testing

Test Case Approach: Positive

Test Case ID: CTT_02

Test Case Title: Verify that the app calculates and displays total calories consumed and burned.

Test Case Description: This test case verifies whether the app accurately calculates and displays the total calorie consumption and calorie expenditure based on user input and activity data.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

* User has logged into the app.

Test Data:

* User's weight and height (for calculating BMR)
* Estimated daily calorie expenditure (from physical activities logged by the user)

Test Steps:

1. Navigate to the "Calorie Summary" screen in the app.
2. Observe the displayed values for total calories consumed and burned for the current day.
3. Compare the displayed calorie values against the user's logged food intake and estimated physical activity.

Postconditions:

* No errors or unexpected calculations are displayed.

Expected Result: The app accurately calculates and displays the total daily calorie consumption and expenditure based on the provided information.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: CTT_03

Test Case Title: Verify that users can set daily calorie goals and track progress.

Test Case Description: This test case verifies the user's ability to set daily calorie goals and track their progress against those goals within the app.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

* User has logged into the app.

Test Data:

* User-specific calorie goal (for demonstration purposes)

Test Steps:

1. Navigate to the "Settings" or "Profile" section of the app.

2. Access the "Calorie Goal" settings.

3. Input the desired daily calorie goal.

Postconditions:

* The newly set calorie goal is saved and displayed.

Expected Result: The app allows users to set personalized daily calorie goals and displays a visual representation of their progress towards achieving these goals.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: CTT_04

Test Case Title: Verify that food items and calorie information are searchable in the database.

Test Case Description: This test case tests the search functionality for finding food items and their corresponding calorie information within the app's database.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

* User has logged into the app.

Test Data:

* A variety of food items with different names and ingredients.

Test Steps:

1. Go to the "Food Log" or "Food Library" screen.

2. Enter various search terms related to specific food items (e.g., "chicken breast," "pizza," "brown rice").

3. Verify that relevant food items appear in the search results.

Postconditions:

* Search is accurate and efficient.

Expected Result: The app retrieves accurate and relevant food items based on the user's search queries, including their calorie information.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Let me know if you need more test cases or want to explore negative test scenarios for this feature.

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:

## Fitness Tracker App: Sleep Tracking Test Cases

User Story Title: As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Project Name: Fitness Tracker App

Feature Name: Sleep Tracking

Test Suite: User Authentication & Data Tracking

Test Case ID: TC_Sleep_001

Test Case Title: Verify that sleep duration is recorded accurately

Test Case Description: This test case aims to verify that the app accurately records sleep duration based on accelerometer and heart rate sensor data.

Test Priority: High

Preconditions:
* User account is created and logged in.
* Sleep tracking is enabled in app settings.
* The user falls asleep within the app's defined sleep window.

Test Data: No test data needed

Test Steps:
1. Start sleep tracking manually through the app.

2. Remain asleep within the app?s defined sleep window for at least 3 hours.

3. Wake up and manually stop sleep tracking.

4. View the sleep summary within the app.

5. Check the recorded sleep duration against the actual time slept.

Postconditions: Sleep data should be recorded and visible in the app's sleep history.

Expected Result: The recorded sleep duration should match the actual time slept within a reasonable margin of error (e.g., ± 15 minutes).

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_Sleep_002

Test Case Title: Verify that sleep quality is rated accurately

Test Case Description: This test case aims to ensure that the app accurately assigns a sleep quality rating based on sensor data and established sleep stage parameters.

Test Priority: High

Preconditions:

* User account is created and logged in.

* Sleep tracking is enabled in app settings.

* The user falls asleep within the app's defined sleep window.

Test Data: No test data needed

Test Steps:

1. Start sleep tracking manually through the app.

2. Simulate different sleep stage patterns (using adjustable weights in simulated sensor data) for at least 3 hours.

3. Wake up and manually stop sleep tracking.

4. View the sleep summary within the app.

5. Check the assigned sleep quality rating against the simulated sleep stage patterns.

Postconditions: Sleep stage data should be recorded and visible in the app's sleep history.

Expected Result: The assigned sleep quality rating should accurately reflect the simulated sleep stage patterns, demonstrating the app's ability to interpret sensor data correctly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive, Negative

This is just a start. You will need to create additional test cases to cover the remaining acceptance criteria, including negative scenarios and edge cases.

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:

## Fitness Tracker App: Workout Logging Test Cases

Here are 5 detailed test cases for the "Workout Logging" feature of the Fitness Tracker App:

Test Case 1: Verify that users can log various workout types.

* **Test Case Description:** This test case verifies the ability to log different types of workouts, ensuring the app handles a variety of exercise categories.
* **Test Suite:** Workout Logging
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the app.
    * App is connected to a fitness tracker (optional, to explore integration scenarios).
* **Test Data:**
    * List of various workout types: Running, Cycling, Swimming, Strength Training, Yoga, HIIT, Walking, etc.
* **Test Steps:**
    1. Navigate to the "Workout Logging" section of the app.
    2. Select a workout type from the provided list.
    3. Input desired duration, intensity, and notes for the workout.
    4. Save the workout entry.
    5. Verify that the logged workout appears in the user's workout history.
* **Postconditions:**
    * Workout entry persists in the app's database.
* **Expected Result:** The app successfully logs the selected workout type with the specified details.
* **Severity:** Critical

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

Test Case 2: Verify that users can edit and delete workout entries.

* **Test Case Description:** This test case ensures the functionality to modify and remove existing workout entries, allowing for data correction and management.
* **Test Suite:** Workout Logging
* **Test Priority:** High
* **Preconditions:**
    * A workout entry has been previously logged successfully (see Test Case 1).
* **Test Data:**
    * Details of the previously logged workout entry (type, duration, intensity).
* **Test Steps:**
    1. Navigate to the user's workout history.
    2. Select the previously logged workout entry.
    3. Edit the workout details (e.g., change duration or intensity).
    4. Save the edited workout entry.
    5. Verify that the changes are reflected in the workout history.
    6. Delete the edited workout entry.
    7. Verify that the deleted entry is no longer visible in the history.
* **Postconditions:**
    * Modified and deleted workout entry is removed from the app's database.
* **Expected Result:** The user can successfully edit and delete workout entries.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that data is stored and viewable for tracking progress.

* **Test Case Description:** This test case confirms the app's ability to persist workout data and allow users to view their progress over time.
* **Test Suite:** Workout Logging
* **Test Priority:** High
* **Preconditions:**
    * Multiple workout entries have been logged.
* **Test Data:**
    * Multiple workout entries with different types, durations, and intensities.
* **Test Steps:**
    1. Navigate to the "Progress Tracking" or "Workout History" section of the app.
    2. Verify that the logged workout entries are displayed chronologically.
    3. Filter the workout history by date range.
    4. Verify that the displayed entries reflect the selected date range.
    5. Analyze the displayed data to confirm progress trends across different workout types and durations.
* **Postconditions:**
    * No modifications to the existing workout data.
* **Expected Result:** The app accurately stores and presents workout data for progress tracking.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 4: Verify users can filter logs by date and workout type.

* **Test Case Description:** This test case ensures the functionality to filter workout logs based on specific

dates and exercise types, allowing for focused data analysis.

* **Test Suite:** Workout Logging

* **Test Priority:** Medium

* **Preconditions:**

    * Several workout entries have been logged with different dates and types.

* **Test Data:**

    * Specific dates and workout types for filtering.

* **Test Steps:**

    1. Navigate to the "Workout History" section of the app.

    2. Using the date filter, select a specific date range.

    3. Verify that only workout entries within the selected date range are displayed.

    4. Using the workout type filter, select a specific workout type (e.g., "Running").

    5. Verify that only workout entries of the selected type are displayed.

    6. Combine both filters (date and workout type) to refine the displayed entries.

* **Postconditions:**

    * No modifications to existing workout data.

* **Expected Result:** The app successfully filters workout logs based on selected criteria.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that users can correctly input and manage workout duration.

* **Test Case Description:** This test case ensures the app handles various workout durations accurately, including negative durations, exceeding maximum limits, and fractional values.

* **Test Suite:** Workout Logging

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the app.

* **Test Data:**

    * Positive, negative, maximum (app-defined) and fractional duration values.

* **Test Steps:**

    1. Navigate to the "Workout Logging" section.

    2. Select a workout type.

    3. Input a positive duration value (e.g., 30 minutes).

    4. Save the workout entry.

    5. Verify that the duration is correctly displayed in the history.

    6. Repeat steps 3-5 with a negative duration value.

    7. Verify that an error message appears, preventing invalid input.

    8. Try exceeding the app's maximum duration limit (if defined).

    9. Verify that the app handles this scenario appropriately (potentially truncating, displaying an error, or allowing input beyond the limit).

    10. Input a fractional duration (e.g., 25.5 minutes).

    11. Verify that the app correctly interprets and saves the fractional duration.

* **Postconditions:**

    * No modifications to existing or future workout data.

Beyond these five examples, you can create additional test cases to further ensure the robustness and reliability of the "Workout Logging" feature, covering scenarios like:

* **Edge cases:** Zero duration workouts, very long durations, unusual characters in notes.

* **Integration with wearable devices:** Verify data sync and accuracy when logging workouts from wearables.

* **Error handling:** Test the app's response to incomplete or invalid workout entries.

* **User interface considerations:** Evaluate the ease of use and clarity of workout logging inputs.

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 ID: FIT-GOAL-001

Test Case Title: Verify that users can set goals for steps.

Test Case Description: This test case verifies the ability of a user to set a specific number of steps as a daily goal within the fitness tracker app.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

* User has registered and logged into the app.

* User's profile is complete (including age, gender, weight).

* The app is connected to a wearable device (optional).

Test Data:

* Step goal value: 10,000

Test Steps:

1. Navigate to the "Goals" section of the app.

2. Select "Steps" from the list of available goals.

3. Enter the desired step goal of 10,000 steps.

4. Save the goal.

5. Verify that the set goal of 10,000 steps is displayed correctly.

Postconditions:

* The user's step goal is saved and displayed in the "Goals" section.

* The app should track the user's steps towards the goal.

Expected Result: The user successfully sets a step goal of 10,000 and the app displays the goal along with progress tracking.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case ID: FIT-GOAL-002

Test Case Title: Verify that users can set goals for weight loss.

Test Case Description: This test case validates the functionality of setting a weight loss goal within the app.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

* User has registered and logged into the app.

* User's profile is complete (including weight).

Test Data:

* Weight loss goal: 5 pounds

Test Steps:

1. Navigate to the "Goals" section of the app.

2. Select "Weight Loss" from the list of available goals.

3. Enter the target weight loss goal of 5 pounds.

4. Save the goal.

5. Verify that the entered weight loss goal of 5 pounds is displayed.

Postconditions:

* The user's weight loss goal is recorded and tracked.

Expected Result: The user successfully sets a weight loss goal of 5 pounds. The app displays the goal and should track the user's progress.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case ID: FIT-GOAL-003

Test Case Title: Verify that users receive motivational feedback for achieving goals.

Test Case Description: This test case confirms that the app provides encouraging messages to the user when they achieve their set goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

* User has set a goal for steps, weight loss, or another metric.

* The user has achieved the set goal.

Test Data:

* User's past goal achievement data (based on test setup).

Test Steps:

1. Set a goal for steps or weight loss within the app (test plan).

2. Simulate the user achieving the set goal (e.g., by manually updating step count or implementing weight loss within the app's logic).

3. Observe the feedback received upon achieving the goal.

Postconditions:

* The goal is marked as achieved within the app.

Expected Result: The app displays a motivational message or acknowledgement congratulating the user on achieving their goal.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, User Experience Testing

Test Case ID: FIT-GOAL-004

Test Case Title: Verify that users can adjust or update existing goals.

Test Case Description: This test case ensures that users have the flexibility to modify or change their previously set fitness goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

* User has successfully set a goal (steps, weight loss, etc.) within the app.

Test Data:

* Original goal value: 5,000 steps

Test Steps:

1. Navigate to the "Goals" section of the app.
2. Locate the previously set goal for steps.
3. Select the option to edit or update the goal.
4. Modify the goal value to a different target.
5. Save the updated goal.
6. Verify that the goal value has been successfully changed.

Postconditions:

* The updated goal is reflected in the app's interface.

Expected Result: The app allows the user to easily adjust their previous goal value and saves the new target.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case ID: FIT-GOAL-005

Test Case Title: Verify that the app provides a summary of goal attainment.

Test Case Description: This test case validates the display of a comprehensive summary of user's goal achievements.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

* User has set multiple goals (steps, weight loss, etc.)

* User has made progress towards some goals and achieved some goals.

Test Data:

* N/A (User data will be used for progress)

Test Steps:

1. Navigate to the "Goals" section of the app.

2. Locate the "Goal Summary" or similar section within the goals view.

3. Review the displayed information, including the following:

    * Completed goals and their attainment status.

    * Progress towards ongoing goals, including percentage achieved.

    * Any relevant timeline visualization or charts.

Postconditions:

* User can access the goal summary overview.

Expected Result:

The app provides a clear and comprehensive summary of all the user's set goals, highlighting progress made, achievements attained, and goals that are still in progress.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

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:

## Test Cases for Fitness Tracker App - Workout Plans Feature

Test Case ID: FIT-WP-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 successfully browse and select workout plans categorized by different fitness levels (Beginner, Intermediate, Advanced).

Test Suite: Workout Plans Feature

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker App.

* Workout Plan section is loaded and accessible.

* Workout plans are categorized based on fitness levels.

Test Data: Workout plans of varying difficulties (Beginner, Intermediate, Advanced)

Test Steps:

1. Navigate to the Workout Plans section.

2. Filter workout plans by different fitness levels (Beginner, Intermediate, Advanced).

3. Verify that the filtered plans display correctly and are categorized accordingly.

4. Select a workout plan from a specific fitness level.

5. Verify that the selected plan belongs to the chosen fitness level.

Postconditions:

* User remains on the selected workout plan page.

* Workout plan details are displayed.

Expected Result: The app successfully filters and displays workout plans based on the selected fitness level.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: FIT-WP-002

Test Case Title: Verify that plans include detailed exercise routines and schedules.

Test Case Description: This test case verifies that chosen workout plans provide detailed information on exercises, sets, repetitions, rest periods, and daily/weekly schedules.

Test Suite: Workout Plans Feature

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker App.
* A specific workout plan is selected.

Test Data: A selected workout plan with detailed exercise information

Test Steps:

1. Select a specific workout plan from the app.
2. Review the planned workout schedule for the week.
3. Examine the details of each workout day, including specific exercises, sets, repetitions, rest periods, and total duration.
4. Verify that the workout instructions are clear and easy to understand.
5. Check for any visual aids or multimedia elements that enhance exercise demonstration.

Postconditions:

* User remains on the chosen workout plan page.

Expected Result: The selected workout plan provides comprehensive and structured exercise information including routines, schedules, and clear instructions.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Content Verification

Test Case Approach: Positive

Test Case ID: FIT-WP-003

Test Case Title: Verify that users can track adherence and progress within the plan.

Test Case Description: This test case verifies the functionality of tracking user adherence to the workout plan and display progress metrics.

Test Suite: Workout Plans Feature

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker App.

* A specific workout plan is selected and actively followed.

* The user has completed some workouts within the plan.

Test Data: Example workout history data

Test Steps:

1. Access the workout tracking section for the selected workout plan.

2. Verify that the app displays completed workout sessions within the plan timeframe.

3. Examine the progress metrics, such as workout frequency, duration, and adherence rate.

4. Check for visual indicators of progress, such as a progress bar or completion percentage.

5. Identify any tools for recording notes or feedback on individual workouts.

Postconditions:

* User remains on the workout plan progress tracking section.

Expected Result: The app accurately reflects user adherence to the workout plan and provides comprehensive progress tracking metrics.

Severity: Critical
Priority: High
Type of Testing: Functional Testing, Data Validation

Test Case Approach: Positive

Test Case ID: FIT-WP-004
Test Case Title: Verify that the app provides notifications and reminders for workouts.

Test Case Description: This test case verifies the functionality of notifications and reminders for upcoming workouts within the selected plan.

Test Suite: Workout Plans Feature
Test Priority: High
Preconditions:
* User is logged into the Fitness Tracker App.
* A specific workout plan is selected.
* Notifications are enabled within the app settings.

Test Data: A future workout scheduled within the selected plan.

Test Steps:

1. Configure the app's notification system to receive reminders for workouts.

2. Set a specific workout schedule within the chosen plan.

3. Verify that the app sends timely notifications or reminders for the scheduled workout session.

4. Check for customizable notification options (frequency, time, type).

5. Ensure that notifications are displayed correctly across different device platforms.

Postconditions:

* User receives notifications for scheduled workouts.

Expected Result: The app reliably delivers timely notifications and reminders for upcoming workouts within the selected plan, customizable according to user preferences.

Severity: High

Priority: High

Type of Testing: Functional Testing, Notification Integration

Test Case Approach: Positive

Test Case ID: FIT-WP-005

Test Case Title: Verify that users can adjust plans based on progress and preferences.

Test Case Description: This test case verifies the user's ability to modify workout plans based on their progress and evolving fitness goals.

Test Suite: Workout Plans Feature

Test Priority: Medium

Preconditions:

* User is logged into the Fitness Tracker App.

* A specific workout plan is selected and actively followed.

* User has progressed significantly within the current plan.

Test Data: A progress track demonstrating substantial user progress.

Test Steps:

1. Assess user progress within the chosen workout plan.

2. Navigate to the plan's settings or modification options.

3. Explore available adjustment options (e.g., increase intensity, modify schedule, swap exercises).

4. Apply changes to the workout plan based on progress and preferences.

5. Verify that the adjusted plan reflects the updated settings.

Postconditions:

* User's workout plan is modified according to their choices.

Expected Result: The app allows for seamless adjustments to workout plans, enabling users to customize their training based on individual progress and evolving needs.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

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 ID: TC_COM_001

Test Case Title: Verify that users can join community groups.

Test Case Description: This test case verifies the functionality for users to join existing community groups within the fitness tracker app. It covers the user journey from selecting a group to successfully joining it.

Test Suite: Community Support

Test Priority: High

Preconditions:

* A user account is created and logged in.

* At least one community group is available within the app.

Test Data: Sample community group names and descriptions.

Test Steps:

1. Navigate to the "Community" section of the app.

2. Browse through the list of available community groups.

3. Select a community group that interests the user.

4. Click on the "Join Group" button.

5. Verify that the user is successfully added to the selected group.

Postconditions:

* The user should see a confirmation message indicating successful group join.

* The user's profile should reflect membership in the selected group.

Expected Result: The user should be able to successfully join the desired community group.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_COM_002

Test Case Title: Verify that users can share progress updates and achievements.

Test Case Description: This test case verifies the functionality for users to post their fitness progress updates and achievements within chosen community groups.

Test Suite: Community Support

Test Priority: High

Preconditions:

* A user account is created and logged in.
* The user has successfully joined at least one community group.

Test Data: Sample fitness progress updates (e.g., number of steps, workout completed, weight loss milestone)

Test Steps:

1. Navigate to the chosen community group within the app.

2. Click on the "Share Update" or "Post" button.

3. Write a detailed progress update including relevant metrics or achievements.

4. Optionally, attach a photo or video related to the update.

5. Click on the "Post" button to share the update with the community.

Postconditions:

* The shared update should be visible on the community group feed.

* Other community members should be able to view, like, and comment on the update.

Expected Result: The user should be able to successfully share their progress updates with the community group.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Repeat the above template for the remaining acceptance criteria, ensuring each test case provides a clear and distinct scenario that comprehensively explores the functionality of community support within the fitness tracker app.

Additional Test Cases to Consider:

* **TC_COM_003**: Verify that users can participate in community challenges.

* **TC_COM_004**: Verify that users can offer and receive support and encouragement.

* **TC_COM_005**: Verify that community activity and engagement are tracked.

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 - Wearable Integration Test Cases

Test Case ID: FIT-WT-001

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: This test case verifies that the app allows users to connect and sync data from various popular wearable devices, ensuring broad compatibility.

Test Suite: Wearable Integration

Test Priority: High

Preconditions:

* User has an account on the Fitness Tracker App.

* User has Bluetooth enabled on their mobile device.

* User has compatible wearable devices (e.g., Fitbit, Apple Watch, Garmin) within range.

Test Data: List of supported wearable devices (provided in app documentation).

Test Steps:

1. Open the Fitness Tracker app and navigate to the "Settings" or "Devices" section.

2. Select the "Add Device" or "Connect Device" option.

3. Choose the desired wearable device from the list of supported devices.

4. Follow the on-screen prompts to initiate pairing and connect the wearable device.

5. Verify that the app successfully connects to the wearable device.

Postconditions:

* Wearable device successfully added to the list of connected devices.

* App displays relevant information from the connected wearable device.

Expected Result: The app successfully connects to the selected wearable device and displays its information.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive, Compatibility Testing

** ------------- **

Test Case ID: FIT-WT-002

Test Case Title: Verify that users can sync activity data from their wearables.

Test Case Description: This test case checks the functionality of syncing activity data, such as steps, heart rate, and calories burned, from a connected wearable device to the Fitness Tracker app.

Test Suite: Wearable Integration

Test Priority: High

Preconditions:

* User has an account on the Fitness Tracker App.
* User has a compatible wearable device paired and connected to the app (see Test Case FIT-WT-001).
* The wearable device has recorded some activity data.

Test Data: Sample activity data recorded by the wearable device (steps, heart rate, calories burned).

Test Steps:

1. Ensure the wearable device has recorded a significant amount of activity data.
2. Open the Fitness Tracker app and navigate to the "Dashboard" or "Activity" section.
3. Initiate a sync process for the connected wearable device (e.g., tap a "Sync" button).
4. Monitor the app's progress and confirm successful data synchronization.

Postconditions:

* App logs and displays activity data received from the wearable device on the corresponding screens.
* The app's activity logs show updated information reflecting the synced data.

Expected Result: The Fitness Tracker app successfully syncs activity data from the connected wearable device, displaying this data within the app.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive, Data Validation

** ------------- **

Test Case ID: FIT-WT-003
Test Case Title: Verify that the app provides real-time data updates from wearables.

Test Case Description: This test case evaluates the real-time data updating functionality from the connected wearable device, ensuring seamless and timely data reflection.

Test Suite: Wearable Integration

Test Priority: High

Preconditions:

* User has an account on the Fitness Tracker App.
* User has a compatible wearable device paired and connected to the app (see Test Case FIT-WT-001).

Test Data:

* Activity data generated in real-time by the wearable device (e.g., steps taken, heart rate fluctuations).

Test Steps:

1. Initiate a physical activity on the user's side while wearing the connected wearable device.
2. Monitor both the wearable device display and the Fitness Tracker app simultaneously.
3. Observe the real-time data updates from the wearable device reflected in the app, such as step count increment, heart rate changes, etc.

Postconditions:

* The Fitness Tracker app correctly and consistently displays real-time data updates as activities occur on the user's end.
* No significant delays or discrepancies are observed between real-time data on the wearable and the app.

Expected Result: The Fitness Tracker app receives and accurately displays real-time data updates from the connected wearable device without noticeable lag.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive, Performance Testing

Test Case ID: FIT-WT-004

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned.

Test Case Description: This test case confirms that the selected wearable data fields are correctly transferred and displayed within the Fitness Tracker app, covering essential metrics for users.

Test Suite: Wearable Integration

Test Priority: Medium

Preconditions:

* User has an account on the Fitness Tracker App.
* User has a compatible wearable device paired and connected to the app (see Test Case FIT-WT-001).
* Wearable device has recorded activity data including steps, heart rate, and calories burned.

Test Data:

* Sample activity data from the wearable device with recorded steps, heart rate, and calories burned.

Test Steps:

1. Open the Fitness Tracker app and navigate to the "Dashboard" or "Activity" section.

2. Verify that the app displays the steps, heart rate, and calories burned data synced from the wearable device.

Postconditions:

* App accurately displays the latest synced steps, heart rate, and calories burned data.

Expected Result: The Fitness Tracker app correctly displays the step count, heart rate, and calorie burn data derived from the connected wearable device.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Data Validation

Test Case Approach: Positive

** ------------- **

Test Case ID: FIT-WT-005
Test Case Title: Verify the app's troubleshooting capabilities for wearable connectivity issues.

Test Case Description: This test case investigates the app's ability to handle common wearable connectivity problems and guide users towards resolution.

Test Suite: Wearable Integration

Test Priority: Medium

Preconditions:

* User has an account on the Fitness Tracker App.
* The user has a compatible wearable device initially connected to the app.

Test Data:

* Network connectivity issues (e.g., disabling Wi-Fi/Bluetooth).
* Barriers between the device and the wearable (e.g., placing the phone in a different room).

Test Steps:

1. Intentionally introduce a common connectivity issue (e.g., disable Bluetooth temporarily) that disrupts the connection between the phone and wearable device.
2. Observe how the app reacts to the lost connection, looking for error messages, notifications, or prompts.
3. Follow the app's suggested troubleshooting steps for re-establishing the wearable connection.
4. Verify successful reconnection and data synchronization.

Postconditions:

* The app provides clear and actionable error messages or guidance for troubleshooting the connectivity issue.
* User successfully resolves the connection issue using the app's provided instructions.

Expected Result: The app effectively identifies, communicates, and guides users through resolution for common wearable connectivity issues, ensuring a seamless user experience.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Negative

Important Note:

These test cases demonstrate a starting point. Further test cases for different scenarios, edge cases, and known vulnerabilities in wearable device integrations should be developed for a comprehensive QA process.

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:

## Fitness Tracker App: Water Intake Tracking Test Cases

Test Case ID: FIT-WT-001

Test Case Title: Verify that users can log their daily water intake.

Test Case Description: This test case verifies the functionality of logging water consumption throughout the day.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged into the app.

- App is connected to internet (if needed for data sync).

Test Data: Predefined water intake values (e.g., 250ml, 500ml, 1L)

Test Steps:

1. Navigate to the Water Intake section of the app.

2. Enter a water intake value (e.g., 250ml) and select "Log Intake".

3. Repeat steps 2 & 3 with different water intake values (e.g., 500ml, 1L).

4. Verify that the logged water intake values are displayed correctly.

5. Confirm that the total daily water consumption is accurately calculated and displayed.

Postconditions:

- Logged water intake persists within the app.

- App displays a record of logged water intakes.

Expected Result: Users can successfully log their daily water intake. Expected total water intake sum reflects the sum of logged values.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case ID: FIT-WT-002

Test Case Title: Verify that the app calculates total water consumption for the day.

Test Case Description: This test case verifies the accuracy of the total daily water consumption calculation.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

 - User is logged into the app.

 - App is connected to internet (if needed for data sync).

Test Data: Multiple water intake values entered throughout the day.

Test Steps:

1. Log various water intake values at different times throughout the day (e.g., morning, noon, afternoon, evening).

2. After completing a predetermined period (e.g., 24 hours), navigate to the Water Intake section.

3. Verify that the total daily water consumption displayed is the accurate sum of all logged intake values.

Postconditions:

- Logged water intake persists within the app.

- App displays a record of logged water intakes.

Expected Result: The app accurately calculates the total water consumption for the day.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case ID: FIT-WT-003

Test Case Title: Verify that users can set daily hydration goals.

Test Case Description: This test case ensures users can define their desired daily water intake targets.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged into the app.

Test Data: Valid hydration goals (e.g., 2L, 3L)

Test Steps:

1. Navigate to the Water Intake section and locate the "Set Daily Goal" option.
2. Enter a valid hydration goal (e.g., 2L) and save the setting.
3. Verify that the set goal is displayed prominently within the Water Intake section.
4. Attempt to set an invalid hydration goal (e.g., 0L, negative value) and confirm an error message is displayed.

Postconditions:

- User-defined hydration goal is stored.
- App displays the set hydration goal.

Expected Result: Users can successfully set a daily hydration goal within a valid range.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case ID: FIT-WT-004

Test Case Title: Verify that reminders encourage regular water intake.

Test Case Description: This test case validates the functionality of reminders prompting users to drink water.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged into the app

- Location Services enabled (if location-based reminders are implemented)

- App is connected to internet (if needed for reminder schedules)

Test Data: Predefined hour/time for reminder

Test Steps:

1. Set a reminder for regular water intake (e.g., every 2 hours) within the app.

2. Allow the app to send reminder notifications

3. Simulate the time passing until the first reminder is due.

4. Receive and dismiss the reminder notification.

Postconditions:

- Reminder appears at set time intervals.

Expected Result: Users receive timely reminders to drink water at the defined intervals.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case ID: FIT-WT-005

Test Case Title: Verify that users can view hydration progress and trends.

Test Case Description: This test case confirms the ability to visualize hydration progress over time.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged into the app.
- App is connected to internet (if needed for historical data)

Test Data: Logged water intake entries over several days.

Test Steps:

1. Log water intake regularly for at least three consecutive days.
2. Navigate to the Water Intake section and locate "Progress" or "History" view.
3. Verify that the graph/chart accurately displays daily water intake, highlighting trends.
4. Check if the progress view shows cumulative water intake for the chosen period.

Postconditions:

- Hydration progress data persists within the app

Expected Result: The app provides a clear visual representation of hydration progress and trends over time.

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:

Here are 5 test cases for the "Nutrition Tracking" feature of the Fitness Tracker App, following the provided template:

Test Case 1

* **Test Case ID:** NUT-001

* **Test Case Title:** Verify that users can log their daily meals and snacks

* **Test Case Description:** This test case aims to validate the functionality of logging meals and snacks in the nutritional tracker.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** High

* **Preconditions:**

    * User is logged into the Fitness Tracker App

   * App is connected to internet

* **Test Data:** List of common meals and snacks (e.g., breakfast, lunch, dinner, fruit, yogurt, salad)

* **Test Steps:**

   1. Navigate to the "Nutrition" section of the app.

   2. Click on the "Log Meal" button.

   3. Select a meal type (e.g., Breakfast).

   4. Search for and select a specific item from the food database (e.g., "Banana").

   5. Enter the serving size.

   6. Click on the "Save" button.

* **Postconditions:**

   * Logged meal appears in the food diary.

   * Macronutrient breakdown for the logged meal should be displayed.

* **Expected Result:** Successful logging of meals and snacks with proper display of nutritional information.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, User Acceptance Testing

Test Case 2

* **Test Case ID:** NUT-002

* **Test Case Title:** Verify that the app calculates macronutrient intake (carbs, fats, proteins)

* **Test Case Description:**   This test case ensures the app accurately calculates macronutrients for logged meals.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** High

* **Preconditions:**

    * User has logged at least three meals with varying macronutrient compositions.

* **Test Data:** Meals with predefined macronutrient values (e.g., high-protein meal, high-carb meal)

* **Test Steps:**

    1. Review the logged meals in the food diary.

    2. Check the macronutrient breakdown (carbs, fats, proteins) for each meal.

    3. Compare calculated macronutrients to expected values.

* **Postconditions:** None

* **Expected Result:** The app accurately calculates and displays the macronutrient content of logged meals within a predefined tolerance.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, Regression Testing

Test Case 3

* **Test Case ID:** NUT-003

* **Test Case Title:** Verify that users can set nutritional goals and track progress

* **Test Case Description:** This test case validates the functionality of setting and monitoring nutritional goals.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the Fitness Tracker App

* **Test Data:** Example nutritional goals (e.g., daily calorie intake, protein target)

* **Test Steps:**

    1. Navigate to the "Goals" section within Nutrition Tracking.

    2. Set a goal for daily protein intake (e.g., 80 grams).

3. Log meals throughout the day, ensuring they contribute to meeting the protein goal.

4. Review the progress tracker for the day.

5. Verify the app displays the percentage of the protein goal achieved.

* **Postconditions:**

    * Nutritional goal is saved in the user's profile.

* **Expected Result:** Users can set nutrition goals and track their progress effectively.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, User Acceptance Testing

Test Case 4

* **Test Case ID:** NUT-004

* **Test Case Title:** Verify that the app provides nutritional information and suggestions

* **Test Case Description:** This test case ensures the app offers relevant nutritional insights.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User has logged a meal with specific macronutrient composition

* **Test Data:** Meal rich in simple carbohydrates (e.g., sugary cereal)

* **Test Steps:**

1. Log a meal with a high proportion of simple carbohydrates.

2. Review the nutritional information provided for the logged meal.

3. Analyze the suggestions offered by the app for that meal.

* **Postconditions:** None

* **Expected Result:** The app displays relevant nutritional information (e.g., high sugar content) and provides actionable suggestions for healthier choices (e.g., replacing with whole-grain cereal).

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functionality Testing, User Acceptance Testing

 Test Case 5

* **Test Case ID:** NUT-005

* **Test Case Title:** Verify that users can view their nutritional history and trends

* **Test Case Description:** This test case tests the app's ability to track and present historical nutritional data.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User has logged meals for at least a week

* **Test Data:** Logged meals spanning a week or more

* **Test Steps:**

    1. Access the "History" section within Nutrition Tracking.

    2. View the logged meals for the past week.

    3. Explore the graphs or charts that display nutritional trends over time (e.g., daily calorie intake, macronutrient distribution).

* **Postconditions:** None

* **Expected Result:** Users can easily review past meal logs and visualize historical trends in their nutritional intake.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing, Regression Testing

---

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:

## Test Cases for Fitness Tracker App - Progress Reports Feature

Test Case ID: FIT-PR-001

Test Case Title: Verify that users receive weekly progress reports.

Test Case Description: This test case verifies the functionality of the app's automatic generation and delivery of weekly progress reports to users.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

* User has an active account in the Fitness Tracker App.

* User has logged at least one day's worth of activity data (steps, workout, heart rate etc.) within the app.

* The user has opted in to receive progress reports.

Test Data: Baseline user activity data for a week

Test Steps:

1. Log in to the Fitness Tracker App as a user.

2. Navigate to the designated section for reporting or notifications.

3. Observe that a weekly progress report is available.

4. Open the weekly progress report.

5. Verify the report includes information on activity levels, workouts, and goal attainment for the past week.

Postconditions:

* The user can access and view their weekly progress report.

Expected Result:

A comprehensive weekly progress report is successfully generated and presented to the user, showcasing a week's worth of activity data.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: FIT-PR-002

Test Case Title: Verify that users receive monthly progress reports.

Test Case Description: This test case verifies the functionality of the app's automatic generation and delivery of monthly progress reports to users.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

* User has an active account in the Fitness Tracker App.

* User has logged activity data for at least a month.

* The user has opted in to receive monthly progress reports.

Test Data: Baseline user activity data for a month

Test Steps:

1. Log in to the Fitness Tracker App as a user.

2. Navigate to the designated section for reporting or notifications.

3. Observe that a monthly progress report is available.

4. Open the monthly progress report.

5. Verify the report includes information on activity levels, workouts, and goal attainment for the past month.

Postconditions:

* The user can access and view their monthly progress report.

Expected Result:

A comprehensive monthly progress report is successfully generated and presented to the user, showcasing a month's worth of activity data.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: FIT-PR-003

Test Case Title: Verify that users can view detailed breakdowns of achievements and trends.

Test Case Description: This test case verifies that users can access detailed breakdowns of their progress, including achievement summaries and trend analysis.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* User has an active account in the Fitness Tracker App.

* User has logged activity data for at least a week.

Test Data: User activity data with distinct achievements and identifiable trends.

Test Steps:

1. Log in to the Fitness Tracker App as a user.

2. Navigate to the weekly or monthly progress report.

3. Select an option to view a detailed breakdown of achievements.

4. Verify the breakdown lists specific achievements (e.g., "completed 5 workouts this week").

5. Select an option to view trend analysis for key metrics (e.g., steps taken, calories burned).

Postconditions:

* The user can access and view detailed breakdown of achievements and trends.

Expected Result:

Users are presented with detailed visualizations or summaries of their achievements and trends in their activity data.

Severity: High

Priority: Medium

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: FIT-PR-004

Test Case Title: Verify that the app provides personalized recommendations based on report findings.

Test Case Description: This test case tests the app's ability to generate personalized recommendations for users based on their progress report findings.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* User has an active account in the Fitness Tracker App.

* User has logged activity data for at least a week.

* User has set at least one fitness goal in the app.

* The user has opted in to receive personalized recommendations.

Test Data: User activity data that indicates areas where improvements could be made based on their goals.

Test Steps:

1. Log in to the Fitness Tracker App as a user.

2. Navigate to the weekly or monthly progress report.

3. Observe that personalized recommendations are presented alongside the report.

4. Verify the recommendations are relevant to the user's activity data and fitness goals.

5. Example: If steps are below target, a recommendation might be to "increase daily step count"

Postconditions:

* The user can access and view personalized recommendations.

Expected Result:

The app accurately analyzes user progress and provides tailored recommendations to help achieve their fitness goals.

Severity: Major

Priority: Medium

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: FIT-PR-005

Test Case Title: Verify that users can customize the frequency and content of reports.

Test Case Description: This test case verifies the ability for users to personalize their progress report experience.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* User has an active account in the Fitness Tracker App.

* User has logged at least one day's worth of activity data.

Test Data:

* User-selected preferences for report frequency and content.

Test Steps:

1. Log in to the Fitness Tracker App as a user.

2. Navigate to the settings or preferences section.

3. Locate the customization options for progress reports.

4. Adjust the frequency of reports (e.g., weekly, bi-weekly, monthly).

5. Select which data points to include (e.g., steps, workouts, calories burned, sleep data).

6. Save the chosen preferences.

Postconditions:

* User settings for report frequency and content are saved.

Expected Result:

Users can successfully modify the frequency and data content included in their progress reports according to their preferences.

Severity: Major

Priority: Medium

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

General Notes:

* These test cases are a starting point and can be expanded upon to cover a wider range of scenarios.

* For each test case, consider creating variations to test negative scenarios (e.g., incorrect data input, missing data) and edge cases (e.g., report for a user with no activity data).

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, adhering to the provided template and guidelines:

## Test Case 1

Test Case ID: TC_IC01

Test Case Title: Verify that users can join various in-app challenges.

Test Case Description: This test case verifies that users can access and successfully join different types of in-app challenges within the fitness tracker app.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

* A registered user is logged into the app.

* The app is connected to the user's fitness data source (optional, depends on challenge type).

* At least one in-app challenge is available.

Test Data: Challenge names, challenge types (e.g., step count, distance, workout duration)

Test Steps:

1. Navigate to the "Challenges" section of the app.

2. Browse through the available challenges and select a challenge to join.

3. Review the challenge details (objectives, rules, duration, rewards).

4. Tap on the "Join Challenge" button.

5. Confirm participation in the challenge.

Postconditions:

* The user should be added to the selected challenge.

* The user's dashboard should reflect their participation in the challenge.

Expected Result: The user successfully joins the chosen in-app challenge.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TC_IC02

Test Case Title: Verify that challenges have clear objectives and rules.

Test Case Description: This test case ensures that in-app challenges have well-defined objectives and rules that are easily understood by users.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

* A registered user is logged into the app.

* At least one in-app challenge is available.

Test Data: Challenge objectives, challenge rules

Test Steps:

1. Navigate to the "Challenges" section of the app.

2. Select a challenge to view its details.

3. Read the challenge objectives carefully.

4. Carefully read and understand the challenge rules.

Postconditions: None

Expected Result: The challenge objectives and rules are clearly stated, easy to comprehend, and provide a comprehensive understanding of the challenge requirements.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Eligibility Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TC_IC03

Test Case Title: Verify that users can track their progress within the challenge.

Test Case Description: This test case verifies the functionality allowing users to monitor their progress toward completing the challenge objectives.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

* A registered user has joined an in-app challenge.

* The app has access to the user's fitness data.

Test Data: Challenge progress bar or metrics

Test Steps:

1. Within the app, navigate to the "Challenges" section.

2. Select the challenge the user is participating in.

3. View the challenge progress tracker.

4. Observe the progress metrics (e.g., steps completed, distance covered, workouts logged, etc.).

5. Compare progress tracker information with actual logged fitness data.

Postconditions:

* The challenge progress tracker should remain updated as the user progresses.

Expected Result: The progress tracker accurately reflects the user's accomplishments within the challenge.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Performance Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TC_IC04

Test Case Title: Verify that rewards are provided for achieving challenge milestones.

Test Case Description: This test case confirms that users receive promised rewards upon reaching specified milestones within the challenge.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

* A registered user has joined an in-app challenge with defined reward milestones.

* The user has completed a challenge milestone.

Test Data: Reward types, reward triggers

Test Steps:

1. Participate in the challenge and complete a pre-defined milestone.

2. Look for notification within the app about achieving the milestone.

3. Navigate to the challenge details section.

4. Verify the receipt of the associated reward.

Postconditions:

* The user's account should reflect the reward received.

Expected Result: The user receives the intended reward upon achieving the specified milestone within the challenge.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Regression Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: TC_IC05

Test Case Title: Verify that users can compete with friends and view leaderboards for challenges.

Test Case Description: This test case verifies the functionality to compete with friends and view leaderboards based on challenge performance.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

* The user is logged into the app.

* The user has friends connected within the app.

* At least one challenge is available with friend competition enabled.

Test Data: User friends list, challenge leaderboard data

Test Steps:

1. Navigate to the "Challenges" section of the app.

2.   Select a challenge that supports friend competition.

3. View the leaderboard to see the ranking of participating friends.

4. Engage in the challenge and track progress, aiming for a higher ranking.

Postconditions:

* The leaderboard should update in real-time reflecting progress of users.

Expected Result: The user can view a leaderboard displaying the current standing of friend competitors within the chosen challenge.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Performance Testing

Test Case Approach: Positive

Note: These test cases provide a starting point and can be further expanded or modified based on the specific implementation details of the "In-App Challenges" feature within the Fitness Tracker App.

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:

## Test Cases for Customizable Dashboards Feature

Project Context: Fitness Tracker App

User Story: As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Test Suite: User Interface - Dashboard Customization

Note:
* These test cases assume the user is logged in.

## Test Case 1

Test Case ID: TC_DASH_001

Test Case Title: Verify that users can select desired metrics to display on their dashboard.

Test Case Description: This test case verifies the functionality of selecting and deselecting fitness metrics for display on the dashboard. It will cover the ability to choose a combination of metric types and ensure that the selected metrics are reflected on the dashboard in real-time.

Preconditions:

* User is logged in to the Fitness Tracker App.
* A default dashboard layout is loaded.

Test Data:

* Required metrics for selection: Steps, Calories Burned, Distance Traveled, Heart Rate.

Test Steps:

1. Navigate to the dashboard page.
2. Locate the "Customize Dashboard" section.
3. Toggle the "Steps" metric switch to "On".
4. Toggle the "Calories Burned" metric switch to "On".
5. Toggle the "Distance Traveled" metric switch to "Off".
6. Toggle the "Heart Rate" metric switch to "On".
7. Observe the dashboard; the selected metrics should be displayed: Steps, Calories Burned, and Heart Rate.

Postconditions:

* No changes made to user profile settings
* No changes made to pre-existing fitness data

Expected Result:

The dashboard displays the selected metrics: Steps, Calories Burned, and Heart Rate, while Distance Traveled is not displayed.

Severity: Minor

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TC_DASH_002

Test Case Title: Verify that users can customize the dashboard layout.

Test Case Description: This test case verifies the functionality of rearranging dashboard widgets and their positioning. It will cover the ability to drag and drop widgets to different locations on the dashboard and ensure the layout is saved and persisted.

Preconditions:

* User is logged in to the Fitness Tracker App.
* A default dashboard layout is loaded.

Test Data:

 * No test data needed

Test Steps:

1. Navigate to the dashboard page.

2. Locate the dashboard widget for ?Steps?.

3. Click and drag the ?Steps? widget to a different position on the dashboard.

4. Release the drag action.

5. Observe the dashboard; the ?Steps? widget should now be positioned in the new location.

Postconditions:

* The user's dashboard layout is saved.

Expected Result:

The ?Steps? widget is successfully moved to the new location on the dashboard, updating the layout as intended.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TC_DASH_003

Test Case Title: Verify that users can update their saved dashboard settings.

Test Case Description: This test case verifies the functionality of modifying existing saved dashboard settings. It will involve selecting different metrics, changing widget positions, and verifying that the updated settings are applicable to future dashboard displays.

Preconditions:

* User is logged in to the Fitness Tracker App.
* A default dashboard layout with preset metrics is loaded.

Test Data:

* A list of different metrics and widget positions to be tested.

Test Steps:

1. Navigate to the dashboard page.
2. Access the dashboard settings or customization menu.
3. Modify the selected metrics, potentially adding or removing some.
4. Change the arrangement of existing widgets on the dashboard.
5. Save the updated dashboard settings.

Postconditions:

* The user's dashboard settings are saved and applied.

Expected Result:

The dashboard reflects the updated metrics and widget arrangement upon revisiting.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TC_DASH_004

Test Case Title: Verify that dashboard displays update in real-time.

Test Case Description: This test case verifies the real-time updating functionality of the dashboard metrics. It will involve performing a physical activity during the test and observing the corresponding metric changes on the dashboard.

Preconditions:

* User is logged in to the Fitness Tracker App.

* A custom dashboard displaying metrics related to the activity.
* The user has a wearable device or fitness equipment capable of tracking the activity linked to the app.

Test Data:

* Data related to the chosen activity (e.g., distance traveled, steps taken, heart rate)

Test Steps:

1. Begin performing a chosen activity (e.g., walking) that impacts the metrics being displayed on the dashboard.
2. Observe the changes in the relevant metrics on the dashboard in real-time as the activity progresses.
3. Stop the activity and observe the updated final metrics on the dashboard.
4. Repeat the activity multiple times, observing the real-time and final metric updates each time.

Postconditions:

* The user's activity data is synchronized with the app.
* The dashboard updates accurately and consistently with the recorded activity data.

Expected Result:

The dashboard metrics reflecting the chosen activity accurately update in real-time as the activity progresses, and the final metric values reflect the total data recorded during the activity.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Performance Testing (Integration with wearable devices)

Test Case Approach: Positive, Load

## Test Case 5

Test Case ID: TC_DASH_005

Test Case Title: Verify that users can reset the dashboard to default settings.

Test Case Description: This test case verifies that the user can restore the default dashboard layout and settings. It will confirm that all personalized settings are removed and replaced with the pre-defined defaults.

Preconditions:

* User is logged in to the Fitness Tracker App.
* A custom dashboard with unique metrics and layout is set.

Test Data:
* No test data needed

Test Steps:

1. Navigate to the dashboard settings or customization menu.
2. Locate the option to reset the dashboard to default settings.

3. Click the "Reset" button.

4. Confirm the reset action if prompted by the app interface.

5. Observe the dashboard; it should now display the default layout and metrics.

Postconditions:

* No changes made to user profile settings other than dashboard reset.

Expected Result: The dashboard reverts to the default layout and displays only the default set of fitness metrics, effectively removing all previously applied customizations.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

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:

## Fitness Tracker App - Test Cases: Social Sharing Feature

User Story: As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

Test Case 1

* **Test Case ID:** TC_SocialSharing_001
* **Test Case Title:** Verify that users can share workout summaries on social media.
* **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.
  * A workout has been logged with relevant data (duration, distance, calories burned, etc.).
* **Test Data:**
  * Sample workout data with various fields filled in.
* **Test Steps:**
  1. Navigate to the workout details screen of the logged workout.
  2. Click on the "Share" button.
  3. Select a target social media platform (e.g., Facebook, Twitter).
  4. Review the pre-populated share content based on workout data.
  5. Publish the share to the selected platform.
* **Postconditions:**
  * The workout summary is successfully shared on the chosen social media platform.
  * The app displays a confirmation message upon successful sharing.
  * The user can view their shared post on their social media profile.
* **Expected Result:** The workout summary is shared on the selected social media platform with relevant information and a user-friendly format.

Severity: High

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

* **Test Case ID:** TC_SocialSharing_002
* **Test Case Title:** Verify that the app supports sharing on multiple social media platforms.
* **Test Case Description:** This test case verifies the app's capability to share workout data with various social media platforms.
* **Test Suite:** Social Sharing
* **Test Priority:** High
* **Preconditions:**
    * User is logged into the app.
    * A workout has been logged with relevant data.
* **Test Data:**
    * Test accounts created on different platforms (e.g., Facebook, Twitter, Instagram).
* **Test Steps:**
    1. Navigate to the workout details screen of the logged workout.
    2. Click on the "Share" button.
    3. Select different social media platforms (e.g., Facebook, Twitter, Instagram).
    4. Confirm each share action individually.
* **Postconditions:**
    * The workout summary is successfully shared on each chosen social media platform.
    * The app displays a confirmation message after each successful share.
    * Each platform's unique sharing interface and post formatting are utilized accordingly.
* **Expected Result:** The app efficiently shares workout data across multiple supported social media

platforms, adapting to each platform's specific sharing mechanism.

Severity: Medium

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

* **Test Case ID:** TC_SocialSharing_003
* **Test Case Title:** Verify that users can customize the content and format of shared posts.
* **Test Case Description:** This test case ensures that users have options to personalize the shared content before posting.
* **Test Suite:** Social Sharing
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the app.
    * A workout has been logged.
    * The "Share" button is accessible on the workout details screen.
* **Test Data:**
    * Various predefined templates and customizable fields within the sharing interface.
* **Test Steps:**
    1. Navigate to the workout details screen.
    2. Click on the "Share" button.
    3. Choose a social media platform.
    4. Modify the pre-populated content (e.g., add a personal message, change the workout focus).
    5. Adjust the formatting options (e.g., choose a different image, add hashtags).

6. Share the customized post.

* **Postconditions:**
    * The shared post on the chosen platform reflects the customized content and formatting selected by the user.
    * No unexpected content alterations or formatting issues occur upon sharing.
* **Expected Result:** Users can customize the content and format of shared posts effectively, enabling them to tailor the message and presentation according to their preferences and audience.

Severity: Medium

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

* **Test Case ID:** TC_SocialSharing_004
* **Test Case Title:** Verify that sharing features are integrated seamlessly with social media.

* **Test Case Description:** This test case ensures smooth integration with social media platforms, preventing errors or disconnections during the sharing process.
* **Test Suite:** Social Sharing
* **Test Priority:** High
* **Preconditions:**
    * User is logged into both the app and their social media account.
    * The app has access permissions to the chosen social media platform.
* **Test Data:**
    * Sample workout data.

* **Test Steps:**

    1. Navigate to the workout details screen.

    2. Click on the "Share" button.

    3. Select a desired social media platform.

    4. Authorize the app to access your social media account (if necessary).

    5. Choose a sharing option and publish the share.

* **Postconditions:**

    * The share process completes successfully without any interruptions or connection errors.

    * The app displays appropriate confirmation messages throughout the sharing process.

* **Expected Result:** Seamless integration with social media platforms is ensured, allowing users to share workout summaries effortlessly without technical complications.

Severity: High

Priority: High

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case 5

* **Test Case ID:** TC_SocialSharing_005

* **Test Case Title:** Verify that users can track the engagement of their shared posts.

* **Test Case Description:** This test case evaluates the app's ability to display engagement metrics for shared posts, such as likes, comments, and shares.

* **Test Suite:** Social Sharing

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the app.

    * A workout has been shared on a chosen social media platform.

* **Test Data:**

* **Test Steps:**

1. Navigate to the workout details screen.

2. Locate the shared post section.

3. Verify the displayed engagement metrics (e.g., likes, comments, shares).

4. Return to the social media platform and manually update the engagement metrics.

5. Revisit the app's workout details screen and confirm the updated metrics reflect in the app.

* **Postconditions:**

    * The app accurately displays the initial engagement metrics for the shared post.

    * Subsequent changes in engagement on the social media platform are reflected in the app.

* **Expected Result:** Users can effectively track the engagement of their shared posts within the app, providing insight into their social reach and content resonation.

Severity: Medium

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Please note: This is just a starting point. Further test cases can be generated to cover specific social media platforms, edge cases (e.g., network connectivity issues, limited characters in share descriptions), and different user scenarios.

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:

## Fitness Tracker App: Rest and Recovery Tracking Test Cases

Test Case ID: FTR-RC-001

Test Case Title: Verify that users can log rest days.

Test Case Description: This test case verifies the functionality of logging rest days through the app. It ensures users can correctly add a rest day to their activity log and the app acknowledges their input.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:
* A user has logged into the app.
* The user's profile is set up with basic information.

Test Data: "2023-10-27" (Date for the rest day)

Test Steps:
1. Navigate to the "Rest & Recovery" section of the app.
2. Tap on the calendar view for the desired day(2023-10-27).
3. Select the "Log Rest Day" option.
4. Confirm the rest day log entry.

5. Verify that the selected date is marked as a rest day in the calendar view.

Postconditions:

* Rest day data is saved in the user's activity log.

* No notifications or prompts are displayed for incorrect data input.

Expected Result: The rest day is successfully logged in the app.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Acceptance Testing

Test Case Approach: Positive

Test Case ID: FTR-RC-002

Test Case Title: Verify that users can log recovery activities.

Test Case Description: This test case verifies the functionality of logging diverse recovery activities through the app. It ensures users can add activities like stretching, yoga, meditation, and the app correctly categorizes them.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

* A user has logged into the app.

Test Data: "Stretching", "Yoga", "Meditation" (Types of recovery activities)

Test Steps:

1. Navigate to the "Rest & Recovery" section of the app.

2. Tap on the "Add Recovery Activity" button.

3. Select one of the predefined recovery activity categories (e.g., "Stretching").

4. Enter the duration of the activity (e.g., 30 minutes).

5. Verify that the selected recovery activity is correctly displayed in the app's activity log.

Postconditions:

* Recovery activity data is saved in the user's activity log alongside its duration.

Expected Result: The recovery activity is successfully logged in the app, and its type and duration are correctly stored.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Acceptance Testing

Test Case Approach: Positive

Please note these are two example test cases, you would generate at least 3 more comprehensive test cases to cover all the acceptance criteria and various scenarios.

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:

## Fitness Tracker App - Audio Coaching Test Cases

Test Case 1:

Test Case ID: TC_AUDIO_001

Test Case Title: Verify that users can access a variety of audio-guided workouts.

Test Case Description: This test case verifies the availability and diversity of audio-guided workouts within the Fitness Tracker app. It will check for different workout types, durations, and intensity levels.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker app.

* Internet connection is available.

* Audio playback functionality in the device is working.

Test Data:

* No test data needed.

Test Steps:

1. Navigate to the "Workouts" section of the app.

2. Select the "Audio Guided" filter.

3. Verify that a variety of workout options are displayed, including different types (e.g., cardio, strength

training, yoga), durations (e.g., 10 minutes, 30 minutes), and intensity levels (e.g., beginner, intermediate, advanced).

Postconditions:

* User remains within the "Workouts" section.

Expected Result: A diverse selection of audio-guided workouts is displayed, covering various types, durations, and intensities.
Severity: Critical
Priority: High
Type of Testing: Functional Testing

Test Case 2:

Test Case ID: TC_AUDIO_002
Test Case Title: Verify that audio coaching includes motivational messages and coaching cues.
Test Case Description: This test case verifies the incorporation of motivational messages and coaching cues within the audio-guided workouts.
Test Suite: Audio Coaching
Test Priority: High
Preconditions:

* User is logged into the Fitness Tracker app.
* Internet connection is available.
* Audio playback functionality in the device is working.

Test Data:

* No test data needed.

Test Steps:

1. Select an audio-guided workout from the available options.
2. Begin the workout and listen to the audio coaching throughout.
3. Observe and note the presence of motivational messages and coaching cues at appropriate intervals during the workout.

Postconditions:

* User completes the selected audio-guided workout.

Expected Result: Motivational messages and coaching cues are consistently delivered throughout the audio workout, providing encouragement and guidance to the user.
Severity: Medium
Priority: High
Type of Testing: Functional Testing , Usability Testing

Test Case 3:

Test Case ID: TC_AUDIO_003
Test Case Title: Verify that users can choose coaching based on workout type and intensity.
Test Case Description: This test case verifies the ability to filter and select audio coaching based on specific workout types and intensity levels.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

* User is logged into the Fitness Tracker app.

* Internet connection is available.

* Audio playback functionality in the device is working.

Test Data:

* No test data needed.

Test Steps:

1. Navigate to the "Workouts" section of the app.

2. Select the "Audio Guided" filter.

3. Use the available filters to select a specific workout type (e.g., cardio) and intensity level (e.g., beginner).

4. Verify that only audio coaching options relevant to the selected criteria are displayed.

Postconditions:

* User remains within the "Workouts" section.

Expected Result: Filtering options allow users to narrow down available audio coaching options to those relevant to their chosen workout type and intensity preference.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case 4:

Test Case ID: TC_AUDIO_004

Test Case Title: Verify that audio coaching seamlessly integrates with other app features.

Test Case Description: This test case assesses the integration of audio coaching with other core app functionalities, such as workout tracking and progress monitoring.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker app.

* Internet connection is available.

Test Data:

* Example workout data (steps taken, calories burned, heart rate)

Test Steps:

1. Start an audio-guided workout within the app.
2. While the workout is active, manually log steps taken or heart rate data.
3. Observe how the audio coaching updates or responds to these manual data entries.
4. Check if workout progress is correctly tracked and visualized alongside the audio coaching.

Postconditions:

* User completes the selected audio-guided workout.

Expected Result:

Audio coaching adapts accordingly to workout data inputs from other app features, providing a seamless and synchronized experience.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, Functional Testing

 Test Case 5:

Test Case ID: TC_AUDIO_005

Test Case Title: Verify that users can control audio settings and preferences.

Test Case Description: This test case examines user control over audio coaching settings, including volume, playback speed, and language selection.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

* User is logged into the Fitness Tracker app.

* Internet connection is available.

* Audio playback functionality in the device is working.

Test Data:

* No test data needed

Test Steps:

1. Navigate to the "Settings" section of the app.
2. Access the audio coaching settings.
3. Adjust the volume level, playback speed, and language options.
4. Observe and confirm the changes take effect during an ongoing audio-guided workout.

Postconditions:

* User remains within the "Settings" section.

Expected Result: Users can confidently adjust audio settings to their preferences, ensuring an optimal and personalized listening experience.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

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:

## Test Cases for Fitness Tracker App - Health App Integration

Test Case 1:

* **Test Case ID:** TC_HEALTH_APP_INTEGRATION_001
* **Test Case Title:** Verify that the app integrates with popular health apps like MyFitnessPal and Headspace
* **Test Case Description:** This test case verifies the app's ability to connect with popular health and wellness apps, focusing on MyFitnessPal for calorie tracking and Headspace for meditation.
* **Test Suite:** Health App Integration
* **Test Priority:** High
* **Preconditions:**
    * User account is created in the Fitness Tracker App.
    * User accounts are created in MyFitnessPal and Headspace.
* **Test Data:** Credentials for MyFitnessPal and Headspace accounts.
* **Test Steps:**
    1. Login to the Fitness Tracker App.
    2. Navigate to the app integrations settings.
    3. Search for and select MyFitnessPal and Headspace integrations.
    4. Authorize app access through the respective platforms.
    5. Verify successful connection by viewing synced data from both apps.
* **Postconditions:**
    * Data from MyFitnessPal and Headspace are successfully integrated.
* **Expected Result:** App seamlessly connects with MyFitnessPal and Headspace, allows data synchronization, and displays combined data in the Fitness Tracker App.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Integration Testing, Functional Testing

Test Case 2:

* **Test Case ID:** TC_HEALTH_APP_DATA_SYNC_001
* **Test Case Title:** Verify that user data successfully syncs from connected health apps
* **Test Case Description:** This test case validates the data synchronization process between the Fitness Tracker App and connected health apps.
* **Test Suite:** Health App Integration
* **Test Priority:** High
* **Preconditions:**
    * App is successfully connected to MyFitnessPal and Headspace (Refer to TC_HEALTH_APP_INTEGRATION_001).
    * Calorie data is updated in MyFitnessPal and meditation sessions are logged in Headspace.
* **Test Data:**
    * Calorie data recorded in MyFitnessPal
    * Meditation session details in Headspace
* **Test Steps:**
    1. Open the Fitness Tracker App.
    2. Navigate to the data dashboard.
    3. Verify recent calorie data from MyFitnessPal is reflected.
    4. Verify the meditation session details from Headspace are accurately displayed.
* **Postconditions:**
    * Data from both MyFitnessPal and Headspace is updated in Fitness Tracker App.
* **Expected Result:** The Fitness Tracker App successfully synchronizes calorie data from MyFitnessPal and meditation session data from Headspace, presenting a consolidated view of the user's activities.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Integration Testing, Functional Testing

Test Case 3:

* **Test Case ID:** TC_HEALTH_APP_INSIGHTS_001

* **Test Case Title:** Verify that the app provides insights based on combined data from multiple apps

* **Test Case Description:** This test case examines the app's ability to generate meaningful insights by analyzing data from connected health apps.

* **Test Suite:** Health App Insights

* **Test Priority:** High

* **Preconditions:**

    * Integration with MyFitnessPal and Headspace is established (Refer to TC_HEALTH_APP_INTEGRATION_001).

    * Sufficient activity data (calories burned, steps taken, meditation sessions) is entered in both apps.

* **Test Data:**

    * Consistent exercise routine data in MyFitnessPal

    * Regular meditation sessions in Headspace.

* **Test Steps:**

    1. Open the Fitness Tracker App.

    2. Navigate to the insights section.

    3. Examine the provided insights generated based on combined data.

    4. Verify insights reflect trends in calorie consumption, exercise levels, and meditation practices.

* **Postconditions:**

    * Insights generated reflect data trends from both MyFitnessPal and Headspace.

* **Expected Result:** The app provides personalized insights that highlight correlations and patterns in the user's activity, calorie intake, and mindfulness practices, fostering a holistic view of their well-being.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

Test Case 4:

* **Test Case ID:** TC_HEALTH_APP_SETTINGS_001
* **Test Case Title:** Verify that integration settings are easy to manage and configure
* **Test Case Description:** This test case assesses the user-friendliness of the app's integration settings, ensuring ease of connecting, disconnecting, and managing health app connections.
* **Test Suite:** Health App Settings
* **Test Priority:** Medium
* **Preconditions:**
    * The app is launched and logged in.
* **Test Data:**
* **Test Steps:**
    1. Navigate to the app integrations settings.
    2. Verify the layout is intuitive and easily navigable.
    3. Select an existing integration to disconnect (e.g., disconnect from Headspace).
    4. Confirm the disconnection process.
    5. Attempt to connect to another health app (e.g., Strava).
    6. Verify successful connection process without excessive steps.
* **Postconditions:**
    * Disconnecting and reconnecting with health apps is completed without issues.
* **Expected Result:** The user interface for managing health app integrations is clear, user-friendly, and allows for smooth connection/disconnection without complications.
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Usability Testing, Functional Testing

Test Case 5:

* **Test Case ID:** TC_HEALTH_APP_ERROR_HANDLING_001

* **Test Case Title:** Verify that app handles errors during integration with health apps gracefully

* **Test Case Description:** This test case verifies the app's ability to handle potential errors during the health app integration process, ensuring a smooth user experience.

* **Test Suite:** Health App Error Handling

* **Test Priority:** Medium

* **Preconditions:**

    * User account is created in the Fitness Tracker App.

* **Test Data:** Invalid credentials for a health app.

* **Test Steps:**

    1. Go to app integration settings.

    2. Attempt to connect to a health app using invalid credentials (e.g., wrong API key).

    3. Observe the displayed error message.

    4. Verify the error message is clear, concise, and informative.

* **Postconditions:**

    * User is informed about the unsuccessful connection.

* **Expected Result:** The app displays an appropriate error message explaining the cause of the issue and guides the user towards resolving it, preventing frustration or ambiguity.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Error handling Testing, Functional Testing

---

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: TC-BC-001

Test Case Title: Verify that users can log weight, body fat percentage, and muscle mass

Test Case Description: This test case will verify the functionality for users to input their body composition data, including weight, body fat percentage, and muscle mass.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker App.
* Body composition tracking feature is enabled in user settings.

Test Data:

* Valid weight values (e.g., 70kg, 150lbs)
* Valid body fat percentage values (e.g., 15%, 25%)

* Valid muscle mass values (e.g., 40kg, 88lbs)

Test Steps:

1. Navigate to the body composition tracking section of the app.
2. Enter a valid weight value in the designated field.
3. Enter a valid body fat percentage value in the designated field.
4. Enter a valid muscle mass value in the designated field.
5. Click the "Save" or "Submit" button to record the data.

Postconditions:

* The entered data is displayed in the body composition history section.
* The app prompts for confirmation of data entry.

Expected Result: The app successfully records the provided body composition data without errors.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-BC-002

Test Case Title: Verify that BMI is calculated and displayed correctly

Test Case Description: This test case verifies that the app accurately calculates and displays BMI based on the logged weight and height (either directly entered or implied by the user data).

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker App.

* Body composition tracking feature is enabled in user settings.
* User has entered a valid weight and height in the app settings or during body composition log.

Test Data:

* Various weight and height combinations (including metric and imperial units).

Test Steps:

1. Navigate to the body composition tracking section of the app.
2. Verify that the BMI value is displayed prominently.
3. Manually calculate BMI based on the user's entered weight and height.
4. Compare the app's calculated BMI with the manually calculated value.

Postconditions:

* The app refreshes the BMI value upon changes in weight or height.

Expected Result: The app accurately displays the BMI based on the provided weight and height data, matching the manual calculation.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive, Verification

Test Case ID: TC-BC-003

Test Case Title: Verify that users can view trends and history of body composition changes

Test Case Description: This test case verifies the app's ability to display historical body composition data in a clear and organized manner, allowing users to track trends and progress.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

* User is logged into the Fitness Tracker App.

* Body composition tracking feature is enabled in user settings.

* User has logged at least two body composition entries on different dates.

Test Data:

* Historical body composition data entries with varying weight, body fat, and muscle mass values.

Test Steps:

1. Navigate to the body composition tracking section of the app.
2. Review the displayed body composition history, ensuring data is chronologically ordered and easily identifiable.
3. Verify the presence of clear visualizations (e.g., charts or graphs) depicting trends in weight, body fat, and muscle mass over time.
4. Identify and navigate to specific data points within the history section.

Postconditions:

* The app allows users to filter and sort the historical data by date, metric, or other relevant criteria.

Expected Result: The app accurately displays the historical body composition data, enabling users to visualize trends and progress effectively.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive, Exploratory

Test Case ID: TC-BC-004

Test Case Title: Verify that the app provides insights and recommendations based on body composition data

Test Case Description: This test case verifies that the app delivers actionable insights and recommendations to users based on their body composition data and goals.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

* User is logged into the Fitness Tracker App.

* Body composition tracking feature is enabled in user settings.
* User has logged sufficient body composition data for the app to analyze.
* User has set at least one goal related to body composition.

Test Data:

* Realistic body composition data sets with diverse scenarios (e.g., weight loss, muscle gain, maintenance).

Test Steps:

1. Navigate to the body composition tracking section of the app.
2. Review any insights or recommendations provided by the app based on the logged data.
3. Analyze the recommendations for relevance and actionability, ensuring they align with the user's goals and progress.

Postconditions:

* The app allows users to customize their insights and recommendations based on their preferences.

Expected Result: The app provides intelligent insights and actionable recommendations tailored to the user's body composition data and goals.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC-BC-005

Test Case Title: Verify that users can set goals for body composition metrics

Test Case Description: This test case verifies the functionality for users to set specific goals for different body composition metrics.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

* User is logged into the Fitness Tracker App.

* Body composition tracking feature is enabled in user settings.

Test Data:

* Smart goals for each metric: weight, body fat percentage, and muscle mass.

Test Steps:

1. Navigate to the body composition tracking section of the app.
2. Locate the section for setting body composition goals.
3. Input a specific, measurable, achievable, relevant, and time-bound (SMART) goal for weight, body fat percentage, and muscle mass.
4. Verify that the app saves the set goals and displays them clearly.

Postconditions:

* The app sends notifications or reminders to the user regarding their body composition goals.

Expected Result: The app allows users to set realistic and achievable goals for each body composition metric and provides ongoing support for tracking progress.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Acceptance Testing

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:

## Fitness Tracker App: Workout Analysis Test Cases

Test Case ID: FT-WA-001

Test Case Title: Verify that users can view detailed performance metrics for each workout.

Test Case Description: This test case aims to verify that the app displays detailed performance metrics for each completed workout, including distance, duration, average heart rate, calories burned, and pace.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

* The user has created an account.
* The user has completed at least one workout.

Test Data: A completed workout record with varying metrics.

Test Steps:

1. Log in to the fitness tracker app.
2. Navigate to the "Workout History" section.
3. Select a previously completed workout.
4. Verify that the app displays accurate and detailed performance metrics for the selected workout.
5. Compare the displayed metrics with the actual workout data.

Postconditions:

* The user remains in the workout history view.

Expected Result: The app displays all relevant performance metrics for the selected workout accurately.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: FT-WA-002

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration.

Test Case Description: This test case aims to verify that the app effectively visualizes and analyzes workout trends, providing insights into workout intensity and duration over time.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

* The user has completed multiple workouts.

Test Data: A sequence of workouts with varying intensity and duration.

Test Steps:

1. Log in to the fitness tracker app.
2. Navigate to the "Workout Analysis" section.
3. Select the desired timeframe for trend analysis (e.g., last 7 days, last month).
4. Verify that the app displays clear and visually appealing charts or graphs representing workout intensity trends.
5. Verify that the app provides insights into workout duration patterns and potential areas for improvement.

Postconditions:

* The user remains in the workout analysis view.

Expected Result: The app accurately visualizes workout intensity and duration trends with meaningful insights.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: FT-WA-003

Test Case Title: Verify that users can track calorie burn and other health metrics.

Test Case Description: This test case verifies the app's ability to track and display calorie burn, alongside other relevant health metrics like heart rate zones and steps taken, for each workout.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

* The user has enabled health data tracking in the app settings.
* The user has a compatible wearable device or fitness tracker.

Test Data: A completed workout recording with activity data from a wearable device.

Test Steps:

1. Log in to the fitness tracker app and ensure wearable device data syncs.
2. Navigate to the "Workout History" section and select a workout.
3. Verify that the app displays accurate calorie burn data for the selected workout.

4. Check for other relevant health metrics like heart rate zones and steps taken.
5. Compare the displayed calorie burn and other health metrics with the data from the wearable device.

Postconditions:

* The user remains in the workout history view.

Expected Result: The app accurately tracks and displays calorie burn and other relevant health metrics during workouts, based on wearable data.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: FT-WA-004

Test Case Title: Verify that the app offers recommendations based on workout analysis.

Test Case Description: This test case focuses on evaluating the app's ability to provide personalized recommendations based on user workout data analysis.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

* The user has completed multiple workouts.
* The fitness level setting in the app is appropriately configured.

Test Data: A user profile with diverse workout history and fitness goals.

Test Steps:

1. Log in to the fitness tracker app.
2. Navigate to the "Workout Analysis" section.
3. Review the analysis report and observed recommendations.
4. Compare the recommendations with the user's training history and goals.
5. Test out a recommended workout plan or training strategy.

Postconditions:

* The user remains in the workout analysis view or the new recommended workout plan.

Expected Result: The app offers relevant and personalized recommendations based on the user's workout analysis, assisting them in achieving their fitness objectives.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive, Explore different scenarios based on user data and goal settings.

Test Case ID: FT-WA-005

Test Case Title: Verify that users can compare current workouts with past performance.

Test Case Description: This test case assesses the app's functionality for comparing current workouts to previous performances, showcasing performance improvements or regressions.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

* The user has completed multiple workouts.

Test Data: Two workout sessions ? a recent one and an older one with different intensities.

Test Steps:

1. Log in to the fitness tracker app.
2. Navigate to the "Workout History" section.
3. Select the recent workout.
4. Verify that the app provides a comparison feature to select a past workout for contrast.
5. Select an older workout from a different timeframe.
6. Compare the metrics and performance of both workouts side-by-side.

Postconditions:

* The user remains in the workout history view.

Expected Result: The app accurately displays a comparative overview of the selected workouts, highlighting differences in performance metrics.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Note: This is not an exhaustive list but provides a starting point for testing the "Workout Analysis" feature.

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:

## Fitness Tracker App - Weather Integration Test Cases

Test Case ID: FIT-WT-001

Test Case Title: Verify that the app displays current weather conditions for the user's location.

Test Case Description: This test case verifies that the app accurately retrieves and displays the current weather conditions, such as temperature, humidity, wind speed, and precipitation, for the user's location.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User account is logged in.
- Location services are enabled on the device.
- Internet connection is available.

Test Data: No test data needed.

Test Steps:

1. Launch the Fitness Tracker app and log in.
2. Navigate to the home screen or weather section.
3. Verify that the app displays the current temperature, humidity, wind speed, and precipitation for the user's location.

Postconditions:

- App is still running.

Expected Result: The app accurately displays the current weather conditions for the user's location.

Severity: Blocker

Priority: High

Type of Testing: System Testing, Functional Testing.

Test Case Approach: Positive

Test Case ID: FIT-WT-002

Test Case Title: Verify that the app allows users to view weather forecasts for different locations.

Test Case Description: This test case verifies that users can input different locations and view weather forecasts for those specific areas.

Test Suite: Weather Integration

Test Priority: Medium

Preconditions:

- User account is logged in.
- Internet connection is available.

Test Data: List of different cities or locations.

Test Steps:

1. Launch the Fitness Tracker app and log in.

2. Navigate to the weather section.

3. Enter a different location (e.g. NYC, London, Tokyo) in the location search field.

4. Verify that the app displays a weather forecast for the entered location.

5. Repeat steps 3-4 with several different locations.

Postconditions:

- App is still running.

Expected Result: The app accurately retrieves and displays weather forecasts for the entered locations.

Severity: Critical

Priority: Medium

Type of Testing: System Testing, Functional Testing, User Interface Testing.

Test Case Approach: Positive

Test Case ID: FIT-WT-003

Test Case Title: Verify that weather data in the app is updated in real-time.

Test Case Description: This test case verifies the real-time updates of weather data in the app by comparing the displayed conditions before and after a specified time interval.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User account is logged in.

- Internet connection is available.

- Weather information is displayed in the app.

Test Data: No test data needed.

Test Steps:

1. Launch the Fitness Tracker app and log in.
2. Navigate to the weather section.
3. Note down the current weather conditions displayed (temperature, humidity, wind speed, etc.).
4. Wait for 10-15 minutes.
5. Compare the retrieved weather information at the end of the waiting period with the initial values.

Postconditions:

- App is still running.

Expected Result: The displayed weather data has been updated to reflect the current conditions after the specified time interval.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing, Performance Testing.

Test Case Approach: Positive, Negative (if real-time updates fail)

Test Case ID: FIT-WT-004

Test Case Title: Verify that the app offers personalized workout recommendations based on weather conditions.

Test Case Description: This test case verifies that the app provides fitness recommendations customized to the current weather conditions, offering appropriate activities for various scenarios like hot or cold weather, rain, or sunny days.

Test Suite: Weather Integration

Test Priority: Medium

Preconditions:

- User account is logged in.
- Location services are enabled on the device.
- Internet connection is available.
- User has set fitness goals in the app.

Test Data:

- Sample weather conditions (e.g., hot and sunny, rainy, cold and windy).

Test Steps:

1. Launch the Fitness Tracker app and log in.
2. Navigate to the workout recommendations section.
3. Simulate different weather conditions (adjust location or manually input data).
4. Verify that the app provides different workout recommendations based on the simulated weather conditions.
5. Example: "Hot and Sunny" prompt - Suggestions for outdoor water sports, light cardio, or early morning workouts.

Postconditions:

- App is still running.

Expected Result: The app offers personalized workout activities suitable for the prevailing weather conditions.

Severity: Major

Priority: Medium

Type of Testing: System Testing, Functional Testing, User Experience Testing.

Test Case Approach: Positive, Negative (if inappropriate recommendations are offered in certain conditions).

Test Case ID: FIT-WT-005

Test Case Title: Verify that the app sends weather alerts for extreme conditions.

Test Case Description: This test case verifies the functionality of weather alerts and ensures they are triggered for specific extreme weather events like thunderstorms, heatwaves, or blizzards.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User account is logged in.
- Location services are enabled on the device.
- Internet connection is available.
- User has enabled weather alerts in the app settings.

Test Data:

- Simulated data for extreme weather events.

Test Steps:

1. Launch the Fitness Tracker app and log in.
2. Navigate to settings and ensure weather alerts are enabled.
3. Simulate different extreme weather conditions (e.g., thunderstorm warning, heatwave alert).
4. Verify that the app sends a timely notification about the simulated extreme weather event.

Postconditions:

- App is still running. A notification has been received.

Expected Result: The app successfully sends appropriate alerts for extreme weather conditions according to user settings.

Severity: Blocker

Priority: High

Type of Testing: System Testing, Functional Testing, User Interface Testing.

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:

## Test Cases for Mood Tracking Feature in Fitness Tracker App

User Story: 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.

Test Suite: User Interface - Mood Tracking

Test Case 1:

* **Test Case ID:** TC_MT_001
* **Test Case Title:** Verify that users can log their mood before a workout.
* **Test Case Description:** This test case will ensure users can accurately select their mood from a predefined list before logging a workout.

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the fitness tracker app.

    * User is on the workout logging screen.

* **Test Data:** Predefined mood options (e.g., Happy, Sad, Neutral, Anxious, Tired).

* **Test Steps:**

    1. Navigate to the workout logging screen.

    2. Click on the mood selection field.

    3. Select a mood option from the list (e.g., "Happy").

    4. Click on the "Start Workout" button.

* **Postconditions:**

    * Workout logged with the selected mood.

* **Expected Result:** The selected mood is successfully recorded in the workout log.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TC_MT_002

* **Test Case Title:** Verify that users can log their energy levels before and after a workout.

* **Test Case Description:** This test case will validate the functionality of logging energy levels both before and after a workout.

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged into the fitness tracker app.

    * User is on the workout logging screen.

* **Test Data:** Energy levels from 1 to 10.

* **Test Steps:**

    1. Navigate to the workout logging screen.

    2. Select an energy level before starting the workout.

    3. Complete the workout.

    4. Enter the energy level after the workout completion.

* **Postconditions:**

    * Workout logged with both pre and post-workout energy levels.

* **Expected Result:** Both input energy levels are recorded accurately in the workout log.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_MT_003

* **Test Case Title:** Verify that users receive insights into mood and energy trends over time.

* **Test Case Description:** This test case will assess the functionality of displaying mood and energy trend visualizations.

* **Test Priority:** High

* **Preconditions:**

  * User has logged multiple workouts with mood and energy level data.

  * User has access to the analytics or insights section of the app.

* **Test Data:** Historical workout data with varying mood and energy levels.

* **Test Steps:**

  1. Access the analytics or insights section.

  2. Navigate to the mood and energy trend visualizations.

  3. Observe the displayed trends for mood and energy levels over time.

* **Postconditions:**

  * Trend visualizations are displayed correctly.

* **Expected Result:** Trends in mood and energy levels are presented visually, showing potential patterns and changes over time.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TC_MT_004

* **Test Case Title:** Verify that users receive personalized recommendations based on their mood and energy data.

* **Test Case Description:** This test case will ensure the recommendation engine provides relevant suggestions based on entered mood and energy data.

* **Test Priority:** Medium
* **Preconditions:**
    * User has logged at least three workouts with mood and energy level data.
    * User is accessing the personalized recommendations feature.
* **Test Data:**
    * User selects different moods and energy levels.
    * Variability in workout types and intensity.
* **Test Steps:**
    1. Access the personalized recommendations section.
    2. Observe the recommendations provided based on logged data.
    3. Repeat steps 1 & 2 with varying moods and energy levels.

* **Postconditions:**
    * Recommendations are updated based on changed user input.

* **Expected Result:**

Recommendations are tailored to the user's mood and energy levels, offering workouts, activities, or tips that align with their current state.

* **Severity:** Critical
* **Priority:** High

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

Test Case 5:

* **Test Case ID:** TC_MT_005
* **Test Case Title:** Verify that users can set goals for improving mood and energy levels.
* **Test Case Description:** This test case will validate the functionality for setting target mood and energy levels.

* **Test Priority:** Medium
* **Preconditions:**
    * User is logged into the fitness tracker app.
    * User is on the settings or goals screen.
* **Test Data:** Target mood and energy levels (within realistic ranges).
* **Test Steps:**
    1. Navigate to the settings or goals section.
    2. Locate the options for setting mood and energy level goals.
    3. Input desired target mood and energy levels.

* **Postconditions:**
    * Goals are saved and displayed correctly.

* **Expected Result:** User-defined mood and energy level goals are stored and can be viewed within the app.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **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:

## Test Cases for Recipe Creation Feature

Test Case ID: RC-001

Test Case Title: Verify that users can input and save ingredients for a recipe.

Test Case Description: This test case verifies that users can successfully enter a list of ingredients with corresponding quantities for a new recipe and that these ingredients are saved correctly.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

* User is logged into the platform.

* Page is loaded successfully.

* User has access to the recipe creation form.

Test Data:

* List of ingredients with various quantities (e.g., 2 cups, ½ teaspoon, 1 lb)

Test Steps:

1. Navigate to the recipe creation page.

2. Enter each ingredient from the test data into the designated ingredient field.

3. Enter the corresponding quantity for each ingredient.

4. Click the "Save Recipe" button.

5. Verify that the entered ingredients and quantities are displayed in the recipe summary.

Postconditions:

* Recipe is successfully saved with all ingredients and quantities intact.

Expected Result: All entered ingredients and quantities should be correctly saved and displayed in the recipe summary.

Severity: High

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: RC-002

Test Case Title: Verify that users can input and save cooking instructions for a recipe.

Test Case Description: This test case verifies that users can enter detailed cooking instructions for a new recipe and save them successfully.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

* User is logged into the platform.

* Page is loaded successfully.

* User has access to the recipe creation form.

Test Data:

* Simulated step-by-step cooking instructions for a recipe (e.g., "Preheat oven to 375°F", "In a medium bowl, whisk together...", etc.)

Test Steps:

1. Navigate to the recipe creation page.

2. Enter the test data into the designated cooking instructions area.

3. Format the instructions with appropriate line breaks and paragraph breaks.

4. Click the "Save Recipe" button.

5. Verify that the entire set of instructions is displayed in the recipe summary.

Postconditions:

* Recipe is successfully saved with all instructions intact.

Expected Result: All entered cooking instructions, formatted correctly, should be saved and displayed accurately in the recipe summary.

Severity: High

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: RC-003

Test Case Title: Verify that users can upload photos for their recipes.

Test Case Description: This test case verifies that users can select and upload images to accompany their recipes.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

* User is logged into the platform.

* Page is loaded successfully.

* User has access to the recipe creation form with a photo upload field.

Test Data:

* A sample image file (e.g., JPEG, PNG) less than the platform's maximum upload size.

Test Steps:

1. Navigate to the recipe creation page.

2. Locate the "Upload Photo" field or button.

3. Select the sample image file from the user's device.

4. Click the "Upload" button.

5. Verify that the uploaded image is displayed successfully in the recipe preview.

Postconditions:

* Recipe is successfully saved with the uploaded photo.

Expected Result: The uploaded image should be displayed correctly in the recipe preview and be saved with the recipe data.

Severity: Medium

Priority: Medium

Type of Testing: Functional

Test Case Approach: Positive, negative (trying to upload an image larger than the platform limit, an unsupported file type.)

Test Case ID: RC-004

Test Case Title: Verify that saved recipes are accessible in the user's profile.

Test Case Description: This test case ensures that newly created recipes are correctly associated with the user's profile and are visible to them.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

* User has successfully created a recipe as described in RC-001, RC-002, and RC-003.

* User is logged into the platform.

* User has access to their profile page.

Test Data:

* N/A

Test Steps:

1. Log out and then log back in to the platform.

2. Navigate to the user's profile page.

3. Verify that the created recipe is listed under their "My Recipes" section or a similar designated area.

Postconditions:

* Recipe is correctly displayed in the user's profile.

Expected Result: The created recipe, including its title, ingredients, instructions, and photo (if uploaded), should be visible under the user's "My Recipes" section.

Severity: Medium

Priority: Medium

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: RC-005

Test Case Title: Verify that other users can view recipes created by registered users.

Test Case Description: This test case ensures that recipes created by other users are publicly accessible and viewable on the platform.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

* Another user account is created and logged in.

* The recipe created in RC-001, RC-002 and RC-003 is saved.

* The recipe is not set to private.

Test Data:

* N/A

Test Steps:

1. Log in to the secondary user account.

2. Navigate to the recipe listing page or use the search function.

3. Find the recipe created by the primary user.

4. Open the recipe and verify that the details (title, ingredients, instructions, photo) are visible and accurate.

Postconditions:

* Recipe is correctly displayed for the viewing user.

Expected Result: The recipe created by the primary user should be accessible to the secondary user and display all relevant information correctly.

Severity: High

Priority: High

Type of Testing: Functional, System

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:

## Recipe Sharing Platform: Recipe Discovery Test Cases

User Story Title: As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Test Case ID: RECIPE-DISCOVERY-001

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

Test Case Description: This test case verifies the functionality of searching for recipes using relevant keywords. It will check if the search results display recipes matching the provided keywords accurately.

Test Suite: Recipe Search Functionality

Test Priority: High

Preconditions:

* A user account is logged in

* There are recipes available on the platform

Test Data:

* Various keywords relevant to recipe categories (e.g., "chocolate," "vegan," "pasta")

Test Steps:

1. Navigate to the recipe search page.
2. Enter a keyword (e.g., "chocolate") in the search bar and click the search button.
3. Verify that the search results display recipes containing the keyword "chocolate" in their title, ingredients, or description.
4. Repeat steps 2-3 with different keywords (e.g., "vegan," "pasta").
5. Ensure that the search results are relevant and accurately match the entered keywords.

Expected Result: The search results page displays a list of recipes relevant to the entered keywords.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RECIPE-DISCOVERY-002

Test Case Title: Verify that users can search for recipes using ingredients.

Test Case Description: This test case verifies the functionality of searching for recipes using specific ingredients. It will check if the search results display recipes containing the specified ingredients.

Test Suite: Recipe Search Functionality

Test Priority: High

Preconditions:

* A user account is logged in

* There are recipes available on the platform

Test Data:

* Specific ingredients (e.g., "chicken," "tomatoes," "chocolate")

Test Steps:

1. Navigate to the recipe search page.

2. Enter an ingredient (e.g., "chicken") in the search bar and click the search button.

3. Verify that the search results display recipes containing the ingredient "chicken" in their ingredients list.

4. Repeat steps 2-3 with different ingredients (e.g., "tomatoes," "chocolate").

5. Ensure that the search results are relevant and accurately match the entered ingredients.

Expected Result: The search results page displays a list of recipes containing the specified ingredients.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RECIPE-DISCOVERY-003

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

Test Case Description: This test case verifies the functionality of filtering search results by different cuisine types.

Test Suite: Recipe Search Functionality

Test Priority: High

Preconditions:

* A user account is logged in

* There are recipes available with associated cuisine types (e.g., Italian, Mexican, Indian) on the platform

Test Data:

* Different cuisine types (e.g., "Italian", "Mexican", "Indian")

Test Steps:

1. Navigate to the recipe search page.

2. Enter a general keyword (e.g., "pasta") in the search bar and click the search button.

3. Click on the "Cuisine" filter option.

4. Select a cuisine type (e.g., "Italian").

5. Verify that the search results display only recipes belonging to the selected cuisine type.

6. Repeat steps 3-5 with different cuisine types.

Expected Result: The search results are filtered to display recipes matching the selected cuisine type.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RECIPE-DISCOVERY-004

Test Case Title: Verify that users can filter search results by dietary preferences.

Test Case Description: This test case verifies the functionality of filtering search results by various dietary preferences (e.g., vegetarian, vegan, gluten-free).

Test Suite: Recipe Search Functionality

Test Priority: High

Preconditions:

* A user account is logged in

* There are recipes available with associated dietary preferences on the platform.

Test Data:

* Various dietary preferences (e.g., "vegetarian," "vegan," "gluten-free")

Test Steps:

1. Navigate to the recipe search page.

2. Enter a general keyword (e.g., "soup") in the search bar and click the search button.

3. Click on the "Dietary" filter option.

4. Select a dietary preference (e.g., "vegetarian").

5. Verify that the search results display only recipes matching the selected dietary preference.

6. Repeat steps 3-5 with different dietary preferences.

Expected Result: The search results are filtered to display recipes matching the selected dietary preference.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RECIPE-DISCOVERY-005

Test Case Title: Verify that recipe details are accessible from search results.

Test Case Description: This test case verifies that clicking on a search result takes the user to a detailed view of the recipe.

Test Suite: Recipe Search Functionality

Test Priority: High

Preconditions:

* A user account is logged in

* There are recipes available with accessible details on the platform

Test Data:

* Any recipe from the search results list

Test Steps:

1. Navigate to the recipe search page.
2. Enter a keyword in the search bar and click the search button.
3. Click on any recipe title from the search results list.
4. Verify that the user is redirected to a detailed recipe page with information such as ingredients, instructions, dietary tags, and images.

Expected Result: The user is successfully redirected to the detailed recipe page for the selected recipe.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive