

Generated Test Cases via llama-3.1-8b-instant 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:

Here are the 5 detailed test cases for the "User Profiles" feature, covering all the acceptance criteria from the

user story:

****Test Case 1: Verify that users can add and edit their name****

*** **Test Case ID:** UPC-001**

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

*** **Test Case Description:**** This test case covers the acceptance criterion that users can add and edit their name. It will verify that the user can input a new name and save it successfully.

*** **Test Suite:** User Profiles**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user is logged in to the platform.

+ The user has a profile created with default values.

*** **Test Data:**** No test data needed.

*** **Test Steps:****

1. Log in to the platform and go to the profile editing page.

2. Input a new name in the name field.

3. Save the profile with the new name.

4. Verify that the name has been updated successfully.

*** **Postconditions:****

+ The new name is reflected in the profile.

+ The save button remains active.

*** **Expected Result:**** The new name is saved successfully.

*** **Severity:** Minor**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

****Test Case 2: Verify that users can set privacy settings for each field****

*** **Test Case ID:** UPC-002**

*** **Test Case Title:** Verify that users can set privacy settings for each field**

*** **Test Case Description:**** This test case covers the acceptance criterion that users can set privacy settings for each field. It will verify that the user can select different privacy settings for each field and save it successfully.

*** **Test Suite:** User Profiles**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user is logged in to the platform.

+ The user has a profile created with default values.

*** **Test Data:** No test data needed.**

*** **Test Steps:****

1. Log in to the platform and go to the profile editing page.

2. Select different privacy settings for each field (e.g., public, friends only, private).

3. Save the profile with the selected privacy settings.

4. Verify that the selected privacy settings are applied successfully.

*** **Postconditions:****

+ The selected privacy settings are reflected in the profile.

+ The save button remains active.

*** **Expected Result:** The selected privacy settings are saved successfully.**

*** **Severity:** Minor**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

****Test Case 3: Verify that the profile picture supports standard image formats****

*** **Test Case ID:** UPC-003**

*** **Test Case Title:** Verify that the profile picture supports standard image formats**

*** **Test Case Description:**** This test case covers the acceptance criterion that the profile picture should support standard image formats. It will verify that the platform accepts various image formats (e.g., JPEG, PNG, GIF).

*** **Test Suite:** User Profiles**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user is logged in to the platform.

+ The user has a profile created with default values.

*** **Test Data:**** Test image files with different formats (e.g., JPEG, PNG, GIF).

*** **Test Steps:****

1. Log in to the platform and go to the profile editing page.

2. Upload a test image file in a supported format (e.g., JPEG).

3. Verify that the image is displayed correctly.

4. Upload a test image file in an unsupported format (e.g., MP3).

5. Verify that an error message is displayed.

*** **Postconditions:****

+ The uploaded image is displayed correctly in the profile.

+ An error message is displayed for the unsupported image format.

*** **Expected Result:**** The uploaded image is displayed correctly in the profile.

*** **Severity:** Major**

*** **Priority:** High**

*** **Type of Testing:** Functional Testing**

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

* **Test Case 4: Verify that users can edit their location**

* **Test Case ID:** UPC-004

* **Test Case Title:** Verify that users can edit their location

* **Test Case Description:** This test case covers the acceptance criterion that users can add and edit their location. It will verify that the user can input a new location and save it successfully.

* **Test Suite:** User Profiles

* **Test Priority:** Medium

* **Preconditions:**

+ The user is logged in to the platform.

+ The user has a profile created with default values.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Log in to the platform and go to the profile editing page.

2. Input a new location in the location field.

3. Save the profile with the new location.

4. Verify that the location has been updated successfully.

* **Postconditions:**

+ The new location is reflected in the profile.

+ The save button remains active.

* **Expected Result:** The new location is saved successfully.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that users receive a confirmation upon saving their profile****

*** **Test Case ID:** UPC-005**

*** **Test Case Title:** Verify that users receive a confirmation upon saving their profile**

*** **Test Case Description:**** This test case covers the acceptance criterion that users should receive a confirmation upon saving their profile. It will verify that a confirmation message is displayed after saving the profile.

*** **Test Suite:** User Profiles**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user is logged in to the platform.

+ The user has a profile created with default values.

*** **Test Data:** No test data needed.**

*** **Test Steps:****

1. Log in to the platform and go to the profile editing page.

2. Make any changes to the profile fields.

3. Save the profile.

4. Verify that a confirmation message is displayed.

*** **Postconditions:****

+ A confirmation message is displayed.

+ The save button remains active.

*** **Expected Result:** A confirmation message is displayed after saving the profile.**

*** **Severity:** Minor**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

Data - 2

Project name -

Social Media Platform

Project description -

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

Feature name -

News Feed

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five detailed test cases that cover each acceptance criterion:

****Test Case 1: News Feed Displays Posts from Followed Accounts****

* **Test Case ID***: NF-001

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

* **Test Suite***: News Feed Module

* **Test Priority***: Medium

* **Preconditions***:

+ User account created

+ At least one followed account exists

* **Test Data***: No test data needed

* **Test Steps***:

1. Log in to the social media platform as a user who has followed at least one account.
2. Navigate to the news feed.
3. Verify that posts from the followed accounts are displayed in the news feed.
4. Check that the posts are displayed in chronological order.

* **Postconditions***: News feed is updated and reflects changes to followed accounts.

* **Expected Result***: Posts from followed accounts are displayed in the news feed in chronological order.

* **Severity***: Major

* **Priority***: Medium

* **Type of Testing***: Functional Testing

* **Test Case Approach***: Positive

****Test Case 2: Content Shown in Chronological or Algorithmic Order****

* **Test Case ID***: NF-002

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

* **Test Suite***: News Feed Module

* **Test Priority**: High

* **Preconditions**:

+ User account created

+ Multiple posts from various accounts exist in the news feed

* **Test Data**: No test data needed

* **Test Steps**:

1. Log in to the social media platform as a user with multiple posts in the news feed.

2. Navigate to the news feed.

3. Verify that the most recent posts are displayed first.

4. Check that older posts are displayed below the most recent ones.

5. Test with different algorithmic ordering scenarios (e.g., popularity, relevance).

* **Postconditions**: News feed is updated, and content is reordered according to the selected algorithm.

* **Expected Result**: Content is displayed in chronological or algorithmic order.

* **Severity**: Critical

* **Priority**: High

* **Type of Testing**: Functional Testing

* **Test Case Approach**: Positive

Test Case 3: Users Can Interact with Posts (Like, Comment, Share)

* **Test Case ID**: NF-003

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

* **Test Suite**: News Feed Module

* **Test Priority**: High

* **Preconditions**:

+ User account created

+ Multiple posts exist in the news feed

* **Test Data** : No test data needed

* **Test Steps** :

1. Log in to the social media platform as a user who can interact with posts.
2. Navigate to the news feed.
3. Select a post and verify that the like, comment, and share options are available.
4. Test with each option by interacting with a post.
5. Verify that the post updates reflect the user's interaction.

* **Postconditions** : Post is updated to reflect user interactions.

* **Expected Result** : Users can interact with posts (like, comment, share) with the expected results being reflected in the post updates.

* **Severity** : Critical

* **Priority** : High

* **Type of Testing** : Functional Testing

* **Test Case Approach** : Positive

****Test Case 4: The Feed Updates in Real-time****

* **Test Case ID** : NF-004

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

* **Test Suite** : News Feed Module

* **Test Priority** : High

* **Preconditions** :

+ User account created

+ Multiple posts exist in the news feed

+ Fresh posts added to the news feed during the test

* **Test Data** : No test data needed

* **Test Steps** :

1. Log in to the social media platform as a user who has interacted with posts.
2. Navigate to the news feed and take a screenshot of the feed.
3. Interact with a post (e.g., like, comment, share).
4. Immediately take another screenshot of the news feed.
5. Compare the two screenshots to ensure that the feed has updated.

* **Postconditions** : News feed is updated to reflect changes.

* **Expected Result** : News feed updates in real-time to reflect changes.

* **Severity** : Blocker

* **Priority** : High

* **Type of Testing** : Functional Testing

* **Test Case Approach** : Positive

****Test Case 5: Users Can Filter Content based on Type or Topic****

* **Test Case ID** : NF-005

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

* **Test Suite** : News Feed Module

* **Test Priority** : Medium

* **Preconditions** :

+ User account created

+ Multiple posts of different types exist in the news feed (e.g., images, videos, text)

+ News feed settings can be customized

* **Test Data** : No test data needed

* **Test Steps** :

1. Log in to the social media platform as a user who can customize news feed settings.
2. Navigate to the news feed.
3. Select a filter option (e.g., "Show only image posts").

4. Verify that the posts in the news feed match the selected filter.

5. Test with different filter options and combinations.

* **Postconditions** : News feed is updated to reflect filter changes.

* **Expected Result** : Users can filter content based on type or topic according to the selected filter.

* **Severity** : Major

* **Priority** : Medium

* **Type of Testing** : Functional Testing

* **Test Case Approach** : Negative

Data - 3

Project name -

Social Media Platform

Project description -

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

Feature name -

Messaging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five detailed test cases that cover the acceptance criteria for the "Messaging" feature:

****Test Case 1: Verify that users can send text messages****

Test Case ID: MSC-001

Test Case Title: Verify that users can send text messages

Test Case Description: This test case verifies that users can send text messages to other users on the social media platform, ensuring that the messaging feature functions as expected.

Test Suite: Messaging Feature Suite

Test Priority: High

Preconditions:

- * User is logged into their account
- * Recipient user is online and available for messaging

Test Data: Test user's username and password, recipient user's username

Test Steps:

1. Launch the messaging feature and select the recipient user's name
2. Enter a text message in the composition field
3. Click the send button to send the message
4. Verify that the message is delivered to the recipient user's inbox

Postconditions:

- * Message is delivered to the recipient user's inbox
- * User can see the sent message in their outbox

Expected Result: The message is sent successfully, and the recipient user receives the message.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that users can send images****

Test Case ID: MSC-002

Test Case Title: Verify that users can send images

Test Case Description: This test case verifies that users can send images to other users on the social media platform, ensuring that the messaging feature supports image sharing.

Test Suite: Messaging Feature Suite

Test Priority: High

Preconditions:

- * User is logged into their account
- * Recipient user is online and available for messaging
- * User has an image stored in their device/gallery (if using a mobile app)

Test Data: Test user's username and password, recipient user's username, image file (jpg/png)

Test Steps:

1. Launch the messaging feature and select the recipient user's name
2. Select the image file from the device/gallery
3. Click the send button to send the image
4. Verify that the image is delivered to the recipient user's inbox

Postconditions:

- * Image is delivered to the recipient user's inbox

* User can see the sent image in their outbox

Expected Result: The image is sent successfully, and the recipient user receives the image.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify messages can be sent in real-time****

Test Case ID: MSC-003

Test Case Title: Verify messages can be sent in real-time

Test Case Description: This test case verifies that messages are sent in real-time, ensuring that users receive messages immediately after sending.

Test Suite: Messaging Feature Suite

Test Priority: High

Preconditions:

* User is logged into their account

* Recipient user is online and available for messaging

Test Data: Test user's username and password, recipient user's username

Test Steps:

1. Launch the messaging feature and select the recipient user's name
2. Enter a text message in the composition field
3. Click the send button to send the message
4. Immediately check the recipient user's inbox for the message

Postconditions:

- * Message is delivered to the recipient user's inbox in real-time
- * User can see the sent message in their outbox

Expected Result: The message is delivered to the recipient user's inbox in real-time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that users can start private conversations or group chats****

Test Case ID: MSC-004

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

Test Case Description: This test case verifies that users can start private conversations or group chats with other users on the social media platform, ensuring that the messaging feature supports group chat functionality.

Test Suite: Messaging Feature Suite

Test Priority: Medium

Preconditions:

- * User is logged into their account
- * Recipient users are online and available for messaging

Test Data: Test user's username and password, recipient user's usernames

Test Steps:

1. Launch the messaging feature and select multiple recipient users' names
2. Click the "Start chat" button to initiate a group chat
3. Verify that the group chat is created and all users are added
4. Verify that users can send messages to the group chat

Postconditions:

- * Group chat is created and users can send messages

- * User can see the group chat in their inbox

Expected Result: The group chat is created successfully, and users can send messages to the group chat.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that messages show read receipts****

Test Case ID: MSC-005

Test Case Title: Verify that messages show read receipts

Test Case Description: This test case verifies that messages show read receipts, ensuring that users know when messages have been read.

Test Suite: Messaging Feature Suite

Test Priority: High

Preconditions:

- * User is logged into their account
- * Message has been sent to the recipient user
- * Recipient user has read the message

Test Data: Test user's username and password, recipient user's username

Test Steps:

1. Launch the messaging feature and select the recipient user's name
2. Verify that the message shows a read receipt (e.g., a checkmark or a tick)
3. Verify that the received timestamp is updated to reflect the read status

Postconditions:

- * Message shows a read receipt
- * Received timestamp is updated

Expected Result: The message shows a read receipt.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

These test cases cover all the acceptance criteria for the "Messaging" feature, including sending text messages, images, and videos, as well as starting private conversations or group chats and receiving read receipts.

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:

Here are five detailed test cases for the Content Sharing feature, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can create and share text posts****

*** **Test Case ID:** TC-001**

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

*** **Test Case Description:**** This test case ensures that users can create and share text posts with their followers or publicly. It covers the acceptance criteria of creating and sharing text posts.

*** **Test Suite:** Content Sharing**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in to the social media platform

+ User has a profile set up

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the social media platform as a registered user

2. Click on the "Create Post" button

3. Enter a text post in the text box

4. Select the audience for the post (followers or public)

5. Click the "Share" button

6. Verify that the post is displayed on the user's profile and in the news feed

*** **Postconditions:** Post is displayed on the user's profile and in the news feed**

* **Expected Result:** The user can create and share a text post with their followers or publicly

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that content can be shared with followers or publicly

* **Test Case ID:** TC-002

* **Test Case Title:** Verify that content can be shared with followers or publicly

* **Test Case Description:** This test case ensures that users can share content with their followers or publicly. It covers the acceptance criteria of sharing content with followers or publicly.

* **Test Suite:** Content Sharing

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the social media platform

+ User has a profile set up

+ User has created a post

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the social media platform as a registered user

2. Click on the "Share" button for a post

3. Select the audience for the post (followers or public)

4. Verify that the post is displayed on the user's profile and in the news feed

5. Verify that the post is visible to the selected audience

* **Postconditions:** Post is displayed on the user's profile and in the news feed

* **Expected Result:** The user can share content with their followers or publicly

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that users can edit or delete their posts****

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

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

* **Test Case Description:** This test case ensures that users can edit or delete their posts. It covers the acceptance criteria of editing or deleting posts.

* **Test Suite:** Content Sharing

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the social media platform

+ User has a profile set up

+ User has created a post

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the social media platform as a registered user

2. Click on the "Edit" button for a post

3. Edit the post and click the "Save" button

4. Verify that the post is updated

5. Click on the "Delete" button for a post

6. Verify that the post is deleted

* **Postconditions:** Post is updated or deleted

* **Expected Result:** The user can edit or delete their posts

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that posts support hashtags and tagging****

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

* **Test Case Title:** Verify that posts support hashtags and tagging

* **Test Case Description:** This test case ensures that posts support hashtags and tagging. It covers the acceptance criteria of posts supporting hashtags and tagging.

* **Test Suite:** Content Sharing

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in to the social media platform

+ User has a profile set up

+ User has created a post

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the social media platform as a registered user

2. Create a post with hashtags and tag a user

3. Verify that the hashtags are displayed on the post

4. Verify that the tagged user is notified

* **Postconditions:** Hashtags are displayed on the post and tagged user is notified

* **Expected Result:** Posts support hashtags and tagging

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that users receive feedback through likes, comments, and shares

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

* **Test Case Title:** Verify that users receive feedback through likes, comments, and shares

* **Test Case Description:** This test case ensures that users receive feedback through likes, comments, and shares. It covers the acceptance criteria of users receiving feedback through likes, comments, and shares.

* **Test Suite:** Content Sharing

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the social media platform

+ User has a profile set up

+ User has created a post

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the social media platform as a registered user

2. Create a post

3. Have another user like, comment, or share the post

4. Verify that the user is notified of the activity

5. Verify that the post is displayed with the updated engagement count

* **Postconditions:** User is notified of activity and post is displayed with updated engagement count

* **Expected Result:** Users receive feedback through likes, comments, and shares

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

Data - 5

Project name -

Social Media Platform

Project description -

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

Feature name -

Privacy Settings

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five detailed test cases that cover the acceptance criteria of the "Privacy Settings" feature:

****Test Case 1: Verify that users can set their profile to public, private, or restricted****

Test Case ID	Test Case Title	Test Description
---	---	---
TC-101	Verify that users can set their profile to public, private, or restricted	This test case ensures that users can select their desired privacy settings for their profile.
Test Suite	Privacy Settings	
Test Priority	Medium	
Preconditions	1. User is logged in. 2. User has a profile created.	
Test Data	No test data needed	
Test Steps	1. Log in to the platform. 2. Go to profile settings. 3. Click on the "Privacy settings" dropdown. 4. Select each available option (public, private, restricted). 5. Save the changes.	
Postconditions	User's profile is updated with the new privacy settings.	
Expected Result	User can select and save their desired privacy settings.	
Severity	Minor	
Priority	Medium	
Type of Testing	Functional Testing	
Test Case Approach	Positive	

****Test Case 2: Verify that users can change privacy settings at any time****

Test Case ID	Test Case Title	Test Description
---	---	---
TC-102	Verify that users can change privacy settings at any time	This test case ensures that users can update their privacy settings without restrictions.
Test Suite	Privacy Settings	

| Test Priority | Medium |

| Preconditions | 1. User is logged in. 2. User has a profile created. 3. Initial privacy settings are set to any option. |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the platform. 2. Go to profile settings. 3. Click on the "Privacy settings" dropdown. 4. Select a new option (public, private, restricted). 5. Save the changes. |

| Postconditions | User's profile is updated with the new privacy settings. |

| Expected Result | User can change their privacy settings anytime. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 3: Verify that users can control visibility of individual posts****

| Test Case ID | Test Case Title | Test Description |

| --- | --- | --- |

| TC-103 | Verify that users can control visibility of individual posts | This test case ensures that users can restrict visibility of specific posts based on their privacy settings. |

| Test Suite | Privacy Settings |

| Test Priority | Medium |

| Preconditions | 1. User is logged in. 2. User has a profile created. 3. User has created at least one post. |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the platform. 2. Go to profile settings. 3. Scroll down to a specific post. 4. Click on the post's settings dropdown. 5. Select "Restrict visibility" and choose an option (public, private, restricted). 6. Save the changes. |

| Postconditions | Post's visibility is updated according to the user's settings. |

| Expected Result | User can control the visibility of individual posts. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 4: Verify that the system respects privacy settings immediately****

| Test Case ID | Test Case Title | Test Description |

| --- | --- | --- |

| TC-104 | Verify that the system respects privacy settings immediately | This test case ensures that the platform updates visibility settings in real-time. |

| Test Suite | Privacy Settings |

| Test Priority | High |

| Preconditions | 1. User is logged in. 2. User has a profile created. 3. User has updated their privacy settings. |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the platform. 2. Check if the updated visibility settings work as expected for different user roles (e.g., friends, followers, public). |

| Postconditions | System behavior conforms to user's updated privacy settings. |

| Expected Result | System respects user's updated privacy settings immediately. |

| Severity | Critical |

| Priority | High |

| Type of Testing | System Testing |

| Test Case Approach | Positive |

****Test Case 5: Verify that users are notified of any changes in privacy settings****

Test Case ID	Test Case Title	Test Description
---	---	---
TC-105	Verify that users are notified of any changes in privacy settings	This test case ensures that users receive notifications about changes in their privacy settings.
Test Suite	Privacy Settings	
Test Priority	Medium	
Preconditions	1. User is logged in. 2. User has a profile created. 3. Initial privacy settings are set to any option.	
Test Data	No test data needed	
Test Steps	1. Log in to the platform. 2. Go to profile settings. 3. Click on the "Privacy settings" dropdown. 4. Select a new option (public, private, restricted). 5. Save the changes. 6. Verify that the user receives a notification about the changes.	
Postconditions	User receives a notification about the updated privacy settings.	
Expected Result	User is notified of any changes in privacy settings.	
Severity	Minor	
Priority	Medium	
Type of Testing	Functional Testing	
Test Case Approach	Positive	

These five test cases ensure that all acceptance criteria related to privacy settings are thoroughly tested, covering public, private, and restricted settings, as well as notifications and system behavior.

Data - 6

Project name -

Social Media Platform

Project description -

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

Feature name -

Notifications

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five detailed test cases that thoroughly cover the acceptance criteria for the Notifications feature:

****Test Case 1: Verify that users receive notifications for likes****

Test Case ID: NT001

Test Case Title: Verify that users receive notifications for likes

Test Case Description:

This test case is designed to verify that users receive notifications when they receive likes on their posts,

photos, or videos. The test case will cover the scenario where a user posts content and receives likes from other users.

Test Suite: Notifications Module

Test Priority: High

Preconditions:

- * User has created a profile
- * User has posted content
- * Other users have liked the content

Test Data: User's profile and posted content

Test Steps:

1. Log in to the application as a user
2. Post content
3. Log in to the application as another user
4. Like the content posted by the first user
5. Verify that the first user receives a notification

Postconditions:

- * User's notification inbox is updated

Expected Result: User receives a notification with a link to the relevant activity (in this case, the liked content)

Severity: Critical

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 2: Verify that users are notified of new friend requests****

Test Case ID: NT002

Test Case Title: Verify that users are notified of new friend requests

Test Case Description:

This test case is designed to verify that users receive notifications when they receive new friend requests. The test case will cover the scenario where a user sends a friend request to another user.

Test Suite: Notifications Module

Test Priority: High

Preconditions:

- * User has created a profile
- * The user has sent a friend request

Test Data: User's profile and friend request status

Test Steps:

1. Log in to the application as a user
2. Send a friend request to another user
3. Verify that the recipient receives a notification

Postconditions:

- * Recipient's profile is updated to show the friend request

Expected Result: Recipient receives a notification with a link to the relevant activity (in this case, the friend request)

Severity: Critical

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 3: Verify that users are notified of new mentions****

Test Case ID: NT003

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

Test Case Description:

This test case is designed to verify that users receive notifications when they are mentioned in posts, comments, or messages. The test case will cover the scenario where a user is mentioned in a post.

Test Suite: Notifications Module

Test Priority: High

Preconditions:

- * User has created a profile
- * The user has been mentioned in a post

Test Data: User's profile and post content

Test Steps:

1. Log in to the application as a user
2. Post content mentioning another user
3. Log in to the application as the mentioned user
4. Verify that the mentioned user receives a notification

Postconditions:

- * Mentioned user's notification inbox is updated

Expected Result: Mentioned user receives a notification with a link to the relevant activity (in this case, the post that mentioned them)

Severity: Critical

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 4: Verify that users can customize notification preferences****

Test Case ID: NT004

Test Case Title: Verify that users can customize notification preferences

Test Case Description:

This test case is designed to verify that users can customize their notification preferences. The test case will cover the scenarios where a user chooses to receive or not receive notifications for likes, comments, and shares.

Test Suite: Notifications Module

Test Priority: Medium

Preconditions:

- * User has created a profile
- * User has notification preferences set up

Test Data: User's profile and notification preferences

Test Steps:

1. Log in to the application as a user
2. Go to the notification preferences page
3. Select the notification types to receive or not receive
4. Verify that the user receives notifications or not

Postconditions:

- * User's notification inbox is updated or not

Expected Result: User receives or is not receives notifications as per their preferences

Severity: Major

Priority: Medium

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 5: Verify that notifications include a link to the relevant activity****

Test Case ID: NT005

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

Test Case Description:

This test case is designed to verify that notifications include a link to the relevant activity. The test case will cover the scenarios where a user receives a notification with a link to a post, a comment, or a message.

Test Suite: Notifications Module

Test Priority: High

Preconditions:

- * User has created a profile
- * User has received a notification

Test Data: User's profile and notification content

Test Steps:

1. Log in to the application as a user
2. Receive a notification
3. Click on the link in the notification
4. Verify that the user is taken to the relevant activity

Postconditions:

- * User is taken to the relevant activity

Expected Result: User is taken to the relevant activity as per the link in the notification

Severity: Critical

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

These test cases provide comprehensive coverage of the acceptance criteria for the Notifications feature, ensuring that users receive notifications for likes, comments, shares, and friend requests, and that notifications include a link to the relevant activity.

Project name -

Social Media Platform

Project description -

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

Feature name -

Search Functionality

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are 5 detailed test cases for the Search Functionality feature of the Social Media Platform:

****Test Case 1: Verify that users can search for content using keywords****

* Test Case ID: SMPSF-001

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

* Test Case Description:

This test case is designed to ensure that users can search for content on the platform using keywords. The test case will cover the acceptance criterion "Users can search for content using keywords or hashtags."

* Test Suite: Search Functionality

* Test Priority: High

* Preconditions:

- + User is logged in to the platform
- + User has created content on the platform

* Test Data: None

* Test Steps:

1. Log in to the platform with a valid user account
2. Type a random keyword in the search bar
3. Click the search button
4. Verify that the search results include content that matches the keyword

* Postconditions:

- + Verify that the user can navigate to the search results page
- + Verify that the search results are displayed in order of relevance

* Expected Result:

The search results page is displayed with content that matches the keyword.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

****Test Case 2: Verify that search results include profiles, posts, and topics****

* Test Case ID: SMPSF-002

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

* Test Case Description:

This test case is designed to ensure that search results on the platform include profiles, posts, and topics. The test case will cover the acceptance criterion "Search results include profiles, posts, and topics."

* Test Suite: Search Functionality

* Test Priority: High

* Preconditions:

- + User is logged in to the platform
- + User has created or has access to content on the platform

* Test Data: None

* Test Steps:

1. Log in to the platform with a valid user account
2. Search for a random keyword or hashtag
3. Verify that the search results include profiles, posts, and topics

* Postconditions:

- + Verify that the search results are displayed in order of relevance

* Expected Result:

The search results include profiles, posts, and topics.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

Test Case 3: Verify that results are displayed in order of relevance

* Test Case ID: SMPSF-003

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

* Test Case Description:

This test case is designed to ensure that the platform displays search results in order of relevance. The test case

will cover the acceptance criterion "Results are displayed in order of relevance."

* Test Suite: Search Functionality

* Test Priority: High

* Preconditions:

+ User is logged in to the platform

+ User has created or has access to content on the platform

* Test Data: None

* Test Steps:

1. Log in to the platform with a valid user account

2. Search for a random keyword or hashtag

3. Verify that the search results are displayed in order of relevance

4. Test the relevance of the search results by searching for the same keyword or hashtag multiple times

* Postconditions:

+ Verify that the search results are displayed in the correct order

* Expected Result:

The search results are displayed in order of relevance.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

****Test Case 4: Verify that users can filter search results by content type****

* Test Case ID: SMPSF-004

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

* Test Case Description:

This test case is designed to ensure that users can filter search results on the platform by content type. The test case will cover the acceptance criterion "Users can filter search results by content type."

* Test Suite: Search Functionality

* Test Priority: Medium

* Preconditions:

+ User is logged in to the platform

+ User has created or has access to content on the platform

* Test Data: None

* Test Steps:

1. Log in to the platform with a valid user account

2. Search for a random keyword or hashtag

3. Select a specific content type (e.g., posts, profiles, or topics) from the filter options

4. Verify that the search results are filtered by the selected content type

* Postconditions:

+ Verify that the filter options are displayed correctly

* Expected Result:

The search results are filtered correctly by the selected content type.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

****Test Case 5: Verify that the search functionality provides suggestions as the user types****

* Test Case ID: SMPSF-005

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

* Test Case Description:

This test case is designed to ensure that the search functionality on the platform provides suggestions as the user types. The test case will cover the acceptance criterion "The search functionality provides suggestions as the user types."

* Test Suite: Search Functionality

* Test Priority: High

* Preconditions:

+ User is logged in to the platform

+ User has created or has access to content on the platform

* Test Data: None

* Test Steps:

1. Log in to the platform with a valid user account

2. Type a slow random keyword or hashtag in the search bar

3. Verify that the search functionality provides suggestions as the user types

* Postconditions:

+ Verify that the suggestions are relevant to the typed keyword or hashtag

* Expected Result:

The search functionality provides relevant suggestions as the user types.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

Data - 8

Project name -

Social Media Platform

Project description -

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

privacy settings, notifications, and analytics.

Feature name -

Hashtags and Tagging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five detailed test cases for the "Hashtags and Tagging" feature:

****Test Case 1: Add Hashtags to Post****

*** **Test Case ID:** SME-HAS-001**

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

*** **Test Case Description:** As a user, I want to add relevant hashtags to my posts, so that I can categorize my content and make it more discoverable to others.**

*** **Test Suite:** Hashtags and Tagging**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in to the platform.

+ User has a post that can be edited.

* **Test Data:** Current date and time, random hashtags (e.g., #test, #example)

* **Test Steps:**

1. Log in to the platform with a valid account.
2. Select a post from the user's feed that can be edited.
3. Click on the "Edit Post" button.
4. Enter random hashtags in the "Hashtags" field (e.g., #test, #example).
5. Save the post with the updated hashtags.

* **Postconditions:**

+ The hashtags are visible on the post.

+ The hashtags are clickable.

* **Expected Result:** User can successfully add hashtags to their posts.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Tag Other Profiles****

* **Test Case ID:** SME-HAS-002

* **Test Case Title:** Verify that users can tag other profiles.

* **Test Case Description:** As a user, I want to tag other profiles in my posts, so that I can increase the visibility and reach of my content.

* **Test Suite:** Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the platform.

+ User has a post that can be edited.

+ Another user exists with a valid account on the platform.

* **Test Data:** Current date and time, username of the tagged user

* **Test Steps:**

1. Log in to the platform with a valid account.

2. Select a post from the user's feed that can be edited.

3. Click on the "Edit Post" button.

4. Enter the username of another user in the "Tag" field (e.g., @johnDoe).

5. Save the post with the tagged user.

* **Postconditions:**

+ The tagged user receives a notification and a comment is added to the post with the username of the tagged user.

+ The tagged user name is clickable.

* **Expected Result:** User can successfully tag other profiles in their posts.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Clickable Hashtags****

* **Test Case ID:** SME-HAS-003

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

* **Test Case Description:** As a user, I want to click on hashtags to explore related content, so that I can discover new and engaging content on the platform.

* **Test Suite:** Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the platform.

+ A post with hashtags exists in the platform.

* **Test Data:** Current date and time, existing hashtags (e.g., #fashion, #beauty)

* **Test Steps:**

1. Select a post with hashtags from the feed.

2. Click on a hashtag (e.g., #fashion).

3. Verify that the hashtag leads to a list of related content (posts, images, videos).

* **Postconditions:**

+ The hashtag is clickable.

+ Related content is displayed.

* **Expected Result:** Hashtag clicks lead to related content.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Search for Content using Hashtags****

* **Test Case ID:** SME-HAS-004

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

* **Test Case Description:** As a user, I want to search for content using hashtags, so that I can easily discover content that interests me.

* **Test Suite:** Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the platform.

+ Multiple posts with existing hashtags exist in the platform.

* **Test Data:** Current date and time, random hashtags (e.g., #test, #example)

* **Test Steps:**

1. Log in to the platform with a valid account.
2. Click on the "Search" button.
3. Enter a random hashtag in the search field (e.g., #test).
4. Verify that search results display relevant content (posts, images, videos).

* **Postconditions:**

+ The hashtag search results display relevant content.

+ The content includes posts, images, and videos.

* **Expected Result:** Users can search for content using hashtags.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Hashtag Suggestions****

* **Test Case ID:** SME-HAS-005

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

* **Test Case Description:** As a user, I want the system to suggest popular hashtags as I type, so that I can easily discover popular topics and save time while writing posts.

* **Test Suite:** Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the platform.

+ Multiple posts with existing hashtags exist in the platform.

* **Test Data:** Current date and time, random search terms (e.g., "movie", "travel")

* **Test Steps:**

1. Log in to the platform with a valid account.
2. Click on the "Create Post" button.
3. Enter a random search term in the "Hashtags" field (e.g., "movie").
4. Verify that the system suggests popular hashtags as the user types.

* **Postconditions:**

- + Suggested hashtags appear below the search field.
- + Clicking on a suggested hashtag navigates the user to a list of related content.

* **Expected Result:** System suggests popular hashtags as users type.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional 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:

Here are the 5 detailed test cases for the Emojis and Reactions feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that Users can react to posts with a variety of emojis****

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**** |

| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

1.1	Verify that Users can react to posts with a variety of emojis	This test case will verify that users can
-----	---	---

react to posts using a variety of emojis.	Emojis and Reactions	High	User logged in, post is visible	No test data needed	1. Open the app and navigate to a post. 2. Select an emoji button on the post. 3. Choose a random emoji from the collection. 4. Verify the emoji is displayed on the post.	Emojis are visible on posts when reacted.	Blocker	High	System Testing	Positive
---	----------------------	------	---------------------------------	---------------------	--	---	---------	------	----------------	----------

1.2	Verify that Users cannot react to posts without logging in	This test case will verify that users are unable
-----	--	--

to react to posts if they are not logged in. | Emojis and Reactions | High | User not logged in, post is visible |

No test data needed | 1. Open the app and navigate to a post. 2. Attempt to select an emoji button on the post.

3. Verify an error message is displayed. | Error message displayed, no reaction added to post. | Blocker | High |

System Testing | Negative |

| 1.3 | Verify that Users can switch between emoji reactions on a post | This test case will verify that users can switch between different emoji reactions on a post. | Emojis and Reactions | Medium | User logged in, post is visible | No test data needed | 1. Open the app and navigate to a post with an emoji reaction. 2. Select a different emoji. 3. Verify the new emoji is displayed on the post. | Emoji switch successful, reaction updated. |

Major | Medium | System Testing | Positive |

| 1.4 | Verify that Users can remove their emoji reaction from a post | This test case will verify that users can remove their emoji reaction from a post. | Emojis and Reactions | Medium | User logged in, post is visible with user's reaction | No test data needed | 1. Open the app and navigate to a post with the user's reaction. 2. Select the "Remove reaction" button. 3. Verify the reaction is removed from the post. | Reaction removed from post. |

Major | Medium | System Testing | Positive |

| 1.5 | Verify that The emoji reaction count is updated correctly | This test case will verify that the emoji reaction count is updated correctly when users react or remove reactions. | Emojis and Reactions | Low | User logged in, post is visible with one reaction | No test data needed | 1. Open the app and navigate to a post with one reaction. 2. Have another user add an emoji reaction. 3. Verify the reaction count is updated. | Reaction count updated correctly. | Minor | Low | System Testing | Positive |

****Test Case 2: Verify that Emojis can be used in comments and messages****

| ****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**** |

| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

| 2.1 | Verify that Users can add emojis to comments | This test case will verify that users can add emojis to

comments. | Emojis and Reactions | High | User logged in, comment section visible | No test data needed | 1. Open the app and navigate to a post with a comment section. 2. Type an emoji into the comment input field. 3. Verify the emoji is visible in the comment. | Emojis visible in comments. | Blocker | High | System Testing | Positive |

| 2.2 | Verify that Users can add emojis to messages | This test case will verify that users can add emojis to messages. | Emojis and Reactions | High | User logged in, messaging feature active | No test data needed | 1. Open the app and navigate to the messaging feature. 2. Type an emoji into the message input field. 3. Verify the emoji is visible in the message. | Emojis visible in messages. | Blocker | High | System Testing | Positive |

| 2.3 | Verify that Emojis are preserved when sending a message or comment | This test case will verify that emojis are preserved when sending a message or comment. | Emojis and Reactions | Medium | User logged in, messaging feature active | No test data needed | 1. Open the app and navigate to the messaging feature. 2. Type an emoji into the message input field. 3. Send the message. 4. Verify the emoji is preserved upon reception. | Emoji preserved on receipt. | Major | Medium | System Testing | Positive |

****Test Case 3: Verify that Users can see a count of reactions on posts****

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**
---	---	---	---	---	---	---	---	---	---	---	---	---	---

| 3.1 | Verify that Reaction count is displayed on posts | This test case will verify that the reaction count is displayed on posts. | Emojis and Reactions | High | User logged in, post is visible | No test data needed | 1. Open the app and navigate to a post. 2. Verify the reaction count is displayed. | Reaction count visible on post. | Blocker | High | System Testing | Positive |

| 3.2 | Verify that Reaction count is updated when new reactions are added | This test case will verify that reaction count is updated when new reactions are added. | Emojis and Reactions | High | User logged in, post is visible with one reaction | No test data needed | 1. Open the app and navigate to a post with one reaction. 2.

Have another user add an emoji reaction. 3. Verify the reaction count is updated. | Reaction count updated correctly. | Blocker | High | System Testing | Positive |

| 3.3 | Verify that Reaction count is preserved when post is loaded or refreshed | This test case will verify that reaction count is preserved when post is loaded or refreshed. | Emojis and Reactions | Medium | User logged in, post is visible | No test data needed | 1. Open the app and navigate to a post. 2. Load or refresh the post. 3. Verify the reaction count remains displayed. | Reaction count preserved on post reload. | Major | Medium | System Testing | Positive |

****Test Case 4: Verify that The platform should support a wide range of emojis****

| ****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**** |

| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

| 4.1 | Verify that Platform supports a wide range of emojis | This test case will verify that the platform supports a wide range of emojis. | Emojis and Reactions | High | No prerequisites | No test data needed | 1. Open the app and navigate to the emoji selector. 2. Attempt to select various emojis. 3. Verify that all emojis are available and functional. | Wide range of emojis supported and functional. | Blocker | High | System Testing | Positive |

| 4.2 | Verify that Platform displays emojis correctly across different devices | This test case will verify that the platform displays emojis correctly across different devices. | Emojis and Reactions | High | No prerequisites | No test data needed | 1. Open the app on multiple devices. 2. Attempt to select and display emojis. 3. Verify that emojis appear correctly across devices. | Emojis display correctly across devices. | Blocker | High | System Testing | Positive |

| 4.3 | Verify that Platform handles edge cases for emojis | This test case will verify that the platform handles edge cases for emojis, such as duplicate or missing emojis. | Emojis and Reactions | Medium | No prerequisites | No test data needed | 1. Open the app and attempt to select duplicate emojis. 2. Attempt to select a missing

emoji. 3. Verify that the platform handles these scenarios correctly. | Edge cases handled correctly by platform. | Major | Medium | System Testing | Positive |

****Test Case 5: Verify that Users can remove or change their reactions****

| ****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**** |

| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

| 5.1 | Verify that Users can remove their reaction from a post | This test case will verify that users can remove their reaction from a post. | Emojis and Reactions | High | User logged in, post is visible with user's reaction | No test data needed | 1. Open the app and navigate to a post with the user's reaction. 2. Select the "Remove reaction" button. 3. Verify the reaction is removed from the post. | Reaction removed from post. | Blocker | High | System Testing | Positive |

| 5.2 | Verify that Users can change their reaction on a post | This test case will verify that users can change their reaction on a post. | Emojis and Reactions | Medium | User logged in, post is visible with user's reaction | No test data needed | 1. Open the app and navigate to a post with the user's reaction. 2. Select a different emoji. 3. Verify the new emoji is displayed on the post. | Emacs updated correctly on post. | Major | Medium | System Testing | Positive |

Please note that the severity and priority levels may need to be adjusted based on the specific requirements and risks associated with each test case.

Social Media Platform

Project description -

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

Feature name -

Live Streaming

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are the detailed test cases for the Live Streaming feature:

****Test Case 1: Start and Stop Live Stream****

*** **Test Case ID:** LIV011**

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

*** **Test Case Description:** As a user, I want to start and stop a live stream to ensure that I can broadcast**

real-time video content to my followers or the public. This test case verifies that users can initiate and terminate a live stream without issues.

* **Test Suite:** Live Streaming

* **Test Priority:** High

* **Preconditions:**

+ User has a verified account on the social media platform.

+ User has sufficient permissions to start a live stream.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform with a verified account.
2. Navigate to the live streaming page.
3. Click the "Start Live Stream" button.
4. Confirm that the live stream starts successfully.
5. Click the "Stop Live Stream" button.

* **Postconditions:**

+ Live stream is terminated.

+ Streamer's account is credited with the live streaming activity.

* **Expected Result:** Users can start and stop live streams successfully.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Public vs. Private Live Stream****

* **Test Case ID:** LIV012

* **Test Case Title:** Verify that live streams can be public or limited to followers.

* **Test Case Description:** As a user, I want to control the visibility of my live stream to ensure that I can share it with my followers or the public. This test case verifies that users can set the visibility of their live stream to public or private.

* **Test Suite:** Live Streaming

* **Test Priority:** Medium

* **Preconditions:**

+ User has a verified account on the social media platform.

+ User has sufficient permissions to start a live stream.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform with a verified account.
2. Navigate to the live streaming page.
3. Click the "Start Live Stream" button.
4. Select the visibility option (public or private).
5. Confirm that the live stream is visible to the selected audience.

* **Postconditions:**

+ Live stream is visible to the selected audience.

+ Streamer's account is credited with the live streaming activity.

* **Expected Result:** Users can set the visibility of their live stream to public or private.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Viewer Interaction****

* **Test Case ID:** LIV013

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

* **Test Case Description:** As a user, I want to engage with the streamer during a live stream to ensure that I can participate in real-time conversations. This test case verifies that viewers can comment and react to the live stream.

* **Test Suite:** Live Streaming

* **Test Priority:** Medium

* **Preconditions:**

- + User has a verified account on the social media platform.
- + User is watching a live stream.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform with a verified account.
2. Navigate to the live streaming page.
3. Watch a live stream.
4. Comment on the live stream.
5. React to the live stream.

* **Postconditions:**

- + Comment and reaction are visible to the streamer.
- + Streamer's account is credited with the live streaming activity.

* **Expected Result:** Viewers can interact with the streamer through comments and reactions.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Viewer Count and Live Comments****

* **Test Case ID:** LIV014

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

* **Test Case Description:** As a user, I want to see the real-time engagement metrics during a live stream to ensure that I can track the audience's participation. This test case verifies that the stream displays the viewer count and live comments.

* **Test Suite:** Live Streaming

* **Test Priority:** Medium

* **Preconditions:**

+ User has a verified account on the social media platform.

+ User is watching a live stream.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform with a verified account.

2. Navigate to the live streaming page.

3. Watch a live stream.

4. Verify that the viewer count is displayed.

5. Verify that live comments are displayed.

* **Postconditions:**

+ Viewer count and live comments are displayed.

+ Streamer's account is credited with the live streaming activity.

* **Expected Result:** The stream displays viewer count and live comments.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Save Live Stream for Later Viewing****

* **Test Case ID:** LIV015

* **Test Case Title:** Verify that users can save the live stream for later viewing.

* **Test Case Description:** As a user, I want to save a live stream for later viewing to ensure that I can revisit the content at a later time. This test case verifies that users can save a live stream for later viewing.

* **Test Suite:** Live Streaming

* **Test Priority:** Low

* **Preconditions:**

+ User has a verified account on the social media platform.

+ User has watched a live stream.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform with a verified account.
2. Navigate to the live streaming page.
3. Watch a live stream.
4. Click the "Save for Later" button.
5. Verify that the live stream is saved for later viewing.

* **Postconditions:**

+ Live stream is saved for later viewing.

+ Streamer's account is credited with the live streaming activity.

* **Expected Result:** Users can save the live stream for later viewing.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 11

Project name -

Social Media Platform

Project description -

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

Feature name -

Stories

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are 5 detailed test cases for the Stories feature, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can create stories with photos, videos, and text overlays****

* Test Case ID: TC001

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

* Test Case Description: This test case ensures that users can create stories with different types of media, including photos, videos, and text overlays, and that these stories are displayed correctly in the platform.

* Test Suite: Stories

* Test Priority: High

* Preconditions:

+ The user is logged in to the platform

+ The user's account is set to public

* Test Data: A valid user account with a public status, a photo, a video, and a text overlay

* Test Steps:

1. Log in to the platform as the test user

2. Go to the "Create Story" page

3. Select a photo and upload it to the platform

4. Select a video and upload it to the platform

5. Add a text overlay to the story

6. Click "Publish" to create the story

* Postconditions:

+ The story is visible to the public

+ The photo, video, and text overlay are displayed correctly in the story

* Expected Result: The user can create a story with photos, videos, and text overlays, and the story is displayed correctly in the platform.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 2: Verify that stories disappear after 24 hours****

* Test Case ID: TC002

* Test Case Title: Verify that stories disappear after 24 hours

* Test Case Description: This test case ensures that stories are deleted from the platform after 24 hours, as specified in the acceptance criteria.

* Test Suite: Stories

* Test Priority: Medium

* Preconditions:

+ The user is logged in to the platform

+ The user has created a story that is set to public

* Test Data: A valid user account, a story that is set to public, and a timestamp tracking system to verify the story's creation time

* Test Steps:

1. Log in to the platform as the test user

2. Create a story that is set to public

3. Record the timestamp of the story's creation

4. Check the platform after 24 hours to verify that the story has disappeared

* Postconditions:

+ The story is no longer visible on the platform

+ The story's content is no longer available

* Expected Result: The story disappears from the platform after 24 hours.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Verify that users can see who viewed their stories****

* Test Case ID: TC003

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

* Test Case Description: This test case ensures that users can see who viewed their stories by checking the story's view counter.

* Test Suite: Stories

* Test Priority: Medium

* Preconditions:

+ The user is logged in to the platform

+ The user has created a story that is set to public

* Test Data: A valid user account, a story that is set to public, and a view counter system

* Test Steps:

1. Log in to the platform as the test user

2. Create a story that is set to public

3. View the story as another user and verify that the view counter increments

4. Check the story's view counter to verify that it displays the correct number of views

* Postconditions:

+ The view counter is updated correctly

+ The user can see the correct number of views for their story

* Expected Result: The user can see who viewed their stories by checking the story's view counter.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Verify that stories can be shared publicly or with selected followers****

* Test Case ID: TC004

* Test Case Title: Verify that stories can be shared publicly or with selected followers

* Test Case Description: This test case ensures that users can share their stories with either the public or with selected followers.

* Test Suite: Stories

* Test Priority: Medium

* Preconditions:

+ The user is logged in to the platform

+ The user has created a story

* Test Data: A valid user account, a story, and a list of followers

* Test Steps:

1. Log in to the platform as the test user

2. Create a story

3. Select the "Share with public" or "Share with selected followers" option

4. Verify that the story is shared correctly based on the chosen option

* Postconditions:

+ The story is shared correctly with either the public or the selected followers

+ The story's visibility is updated correctly

* Expected Result: The user can share their stories with either the public or with selected followers.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Verify that users can add stickers and filters to their stories****

* Test Case ID: TC005

* Test Case Title: Verify that users can add stickers and filters to their stories

* Test Case Description: This test case ensures that users can add stickers and filters to their stories, enhancing their visual appeal.

* Test Suite: Stories

* Test Priority: Low

* Preconditions:

+ The user is logged in to the platform

+ The user has created a story

* Test Data: A valid user account, a story, and a list of available stickers and filters

* Test Steps:

1. Log in to the platform as the test user

2. Create a story

3. Select a sticker or filter to add to the story

4. Verify that the sticker or filter is added correctly to the story

* Postconditions:

+ The sticker or filter is added correctly to the story

+ The story's visual appearance is updated correctly

* Expected Result: The user can add stickers and filters to their stories.

* Severity: Minor

* Priority: Low

* Type of Testing: Functional Testing

* Test Case Approach: Positive

These test cases cover the acceptance criteria for the Stories feature, ensuring that users can create stories with photos, videos, and text overlays, that stories disappear after 24 hours, that users can see who viewed their stories, that stories can be shared publicly or with selected followers, and that users can add stickers and filters

to their stories. By executing these test cases, we can ensure that the Stories feature meets the acceptance criteria and provides a high-quality user experience.

Data - 12

Project name -

Social Media Platform

Project description -

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

Feature name -

Analytics and Insights

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are the 5 detailed test cases for the Analytics and Insights feature:

****Test Case 1: Verify that users can view data on follower demographics****

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

*** **Test Case Description:**** This test case covers the acceptance criterion that users can view data on follower demographics. It ensures that users can access this information in a user-friendly format.

*** **Test Suite:**** Analytics and Insights

*** **Test Priority:**** High

*** **Preconditions:****

- * User must have created a profile on the social media platform

- * User must have enabled analytics tracking for their profile

*** **Test Data:****

- * User's profile information

- * Followee demographic data

*** **Test Steps:****

1. Log in to the social media platform as a registered user with a profile

2. Navigate to the analytics dashboard for the user's profile

3. Select the "Follower Demographics" tab from the analytics dashboard

4. Verify that the follower demographics data is displayed in a user-friendly format

5. Verify that the data includes the required information (e.g., age range, location, interests)

*** **Postconditions:****

- * The user's follower demographics data is successfully retrieved and displayed on the analytics dashboard

- * The analytics data is accurate and up-to-date

*** **Expected Result:**** The user can view their follower demographics data in a user-friendly format.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that post reach and engagement metrics are available****

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

* **Test Case Description:** This test case covers the acceptance criterion that post reach and engagement metrics are available. It ensures that users can access this information in a user-friendly format.

* **Test Suite:** Analytics and Insights

* **Test Priority:** High

* **Preconditions:**

- * User must have created a profile on the social media platform

- * User must have posted content on the platform

* **Test Data:**

- * User's profile information

- * Post data (including engagement metrics)

* **Test Steps:**

1. Log in to the social media platform as a registered user with a profile

2. Navigate to the analytics dashboard for the user's profile

3. Select the "Post Reach and Engagement" tab from the analytics dashboard

4. Verify that the post reach and engagement metrics are displayed in a user-friendly format

5. Verify that the data includes the required information (e.g., reach, likes, comments, shares)

* **Postconditions:**

- * The post reach and engagement metrics are successfully retrieved and displayed on the analytics dashboard

- * The analytics data is accurate and up-to-date

* **Expected Result:** The user can view their post reach and engagement metrics in a user-friendly format.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can track audience growth trends

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

* **Test Case Description:** This test case covers the acceptance criterion that users can track audience growth trends. It ensures that users can access this information in a user-friendly format.

* **Test Suite:** Analytics and Insights

* **Test Priority:** High

* **Preconditions:**

- * User must have created a profile on the social media platform

- * User must have a sufficient following on the platform

* **Test Data:**

- * User's profile information

- * Audience growth trend data

* **Test Steps:**

1. Log in to the social media platform as a registered user with a profile

2. Navigate to the analytics dashboard for the user's profile

3. Select the "Audience Growth Trends" tab from the analytics dashboard

4. Verify that the audience growth trend data is displayed in a user-friendly format

5. Verify that the data includes the required information (e.g., follower growth rate, follower count)

* **Postconditions:**

- * The audience growth trend data is successfully retrieved and displayed on the analytics dashboard

- * The analytics data is accurate and up-to-date

- * **Expected Result:** The user can view their audience growth trends in a user-friendly format.

- * **Severity:** Major

- * **Priority:** High

- * **Type of Testing:** Functional Testing

- * **Test Case Approach:** Positive

Test Case 4: Verify that analytics are displayed in a user-friendly format

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

- * **Test Case Description:** This test case covers the acceptance criterion that analytics are displayed in a user-friendly format. It ensures that users can access this information in a clear and intuitive manner.

- * **Test Suite:** Analytics and Insights

- * **Test Priority:** Medium

- * **Preconditions:**

- * User must have created a profile on the social media platform

- * User must have analytics tracking enabled for their profile

- * **Test Data:**

- * User's profile information

- * Analytics data

- * **Test Steps:**

1. Log in to the social media platform as a registered user with a profile

2. Navigate to the analytics dashboard for the user's profile

3. Verify that the analytics data is displayed in a clear and user-friendly format

4. Verify that the data includes intuitive visualizations (e.g., charts, graphs)

5. Verify that the data can be easily filtered and sorted

- * **Postconditions:**

- * The analytics data is successfully retrieved and displayed on the analytics dashboard
- * The analytics data is accurate and up-to-date
- * **Expected Result:** The user can view their analytics data in a user-friendly format.
- * **Severity:** Minor
- * **Priority:** Medium
- * **Type of Testing:** Functional Testing
- * **Test Case Approach:** Positive

Test Case 5: Verify that data can be filtered by time range and content type

- * **Test Case Title:** Verify that data can be filtered by time range and content type
- * **Test Case Description:** This test case covers the acceptance criterion that data can be filtered by time range and content type. It ensures that users can access this information in a user-friendly format.
- * **Test Suite:** Analytics and Insights
- * **Test Priority:** Medium
- * **Preconditions:**
 - * User must have created a profile on the social media platform
 - * User must have analytics tracking enabled for their profile
- * **Test Data:**
 - * User's profile information
 - * Analytics data
- * **Test Steps:**

1. Log in to the social media platform as a registered user with a profile
2. Navigate to the analytics dashboard for the user's profile
3. Select the "Filter" option to apply time range and content type filtering
4. Verify that the data can be successfully filtered by time range and content type
5. Verify that the filtered data is displayed in a user-friendly format

* **Postconditions:**

* The data is successfully filtered by time range and content type

* The filtered data is accurate and up-to-date

* **Expected Result:** The user can filter data by time range and content type.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional 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 five detailed test cases for the "Events and Groups" feature, following the Test Case Creation Template:

****Test Case 1: Verify that Users can Create and Join Events****

*** **Test Case ID:** EVCJ01**

*** **Test Case Title:** Verify that Users can Create and Join Events**

*** **Test Case Description:**** As a user, I want to create and join events, so that I can connect with others who share my interests. This test case verifies that users can successfully create and join events on the platform.

*** **Test Suite:** Events and Groups**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in on the platform

+ User has a valid profile

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the platform as a user
2. Navigate to the events page and click on the "Create Event" button
3. Fill in the event details (name, date, time, location, description)
4. Click on the "Join Event" button for a created event
5. Verify that the event is created successfully and the user is added as an attendee

* **Postconditions:**

+ Event is created and visible on the platform

+ User is listed as an attendee for the event

* **Expected Result:** The event is created successfully, and the user can join it.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that Users can Join Interest-Based Groups****

* **Test Case ID:** EJB01

* **Test Case Title:** Verify that Users can Join Interest-Based Groups

* **Test Case Description:** As a user, I want to join interest-based groups, so that I can connect with others who share my interests. This test case verifies that users can successfully join groups on the platform.

* **Test Suite:** Events and Groups

* **Test Priority:** High

* **Preconditions:**

+ User is logged in on the platform

+ User has a valid profile

+ Group exists on the platform

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user
2. Navigate to the groups page and find a group of interest
3. Click on the "Join Group" button for the selected group
4. Verify that the user is added as a member of the group

* **Postconditions:**

+ User is listed as a member of the group

+ Group discussions become visible to the user

* **Expected Result:** The user can join the group successfully and access its discussions.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Group Discussions are Organized by Topic****

* **Test Case ID:** EGD01

* **Test Case Title:** Verify that Group Discussions are Organized by Topic

* **Test Case Description:** As a user, I want to participate in organized discussions within groups, so that I can engage with others who share my interests. This test case verifies that group discussions are organized by topic on the platform.

* **Test Suite:** Events and Groups

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in on the platform

+ User has a valid profile

+ Group exists on the platform

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user

2. Navigate to a group and find a discussion

3. Verify that the discussion is organized by topic and has a clear structure

* **Postconditions:**

+ Discussions are visible and accessible within the group

+ Topics are organized and easy to navigate

* **Expected Result:** Group discussions are organized by topic, allowing users to engage in cohesive and relevant conversations.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that Event Details Include Date, Time, Location, and Description****

* **Test Case ID:** EED01

* **Test Case Title:** Verify that Event Details Include Date, Time, Location, and Description

* **Test Case Description:** As a user, I want to view accurate and comprehensive event details, so that I can make informed decisions about attending events. This test case verifies that event details include date, time, location, and description on the platform.

* **Test Suite:** Events and Groups

* **Test Priority:** High

* **Preconditions:**

+ User is logged in on the platform

+ User has a valid profile

+ Event exists on the platform

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user

2. Navigate to the events page and select an event

3. Verify that the event details include date, time, location, and description

* **Postconditions:**

+ Event details are visible to the user

+ Detail information is accurate and up-to-date

* **Expected Result:** Event details include the required information, such as date, time, location, and description.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that Users can RSVP to Events and See the List of Attendees****

* **Test Case ID:** ER01

* **Test Case Title:** Verify that Users can RSVP to Events and See the List of Attendees

* **Test Case Description:** As a user, I want to RSVP to events and view the list of attendees, so that I can connect with others who are attending the event. This test case verifies that users can successfully RSVP and view the list of attendees on the platform.

* **Test Suite:** Events and Groups

* **Test Priority:** High

* **Preconditions:**

+ User is logged in on the platform

+ User has a valid profile

+ Event exists on the platform

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user

2. Navigate to the events page and select an event
3. Click on the "RSVP" button for the selected event
4. Verify that the user's RSVP status is updated and visible
5. Click on the "Attendees" tab to view the list of attendees

* **Postconditions:**

- + User's RSVP status is updated and visible
- + List of attendees is visible to the user

* **Expected Result:** Users can successfully RSVP to events and view the list of attendees, allowing for connections and notifications.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional 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:

Here are five detailed test cases that cover the acceptance criteria for the Monetization Options feature in the Social Media Platform:

****Test Case 1: Verify that Users can Apply for Monetization Options****

*** **Test Case ID:** MON-001**

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

*** **Test Case Description:** This test case ensures that users can successfully apply for monetization options and provided necessary information.**

*** **Test Suite:** Monetization Options**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in to their account

+ User has not already applied for monetization options

*** **Test Data:** No test data needed**

****Test Steps:****

1. Navigate to the monetization options page
2. Click on the "Apply" button
3. Fill out the application form with required information
4. Submit the application
5. Verify that the application is successfully submitted

****Postconditions:****

- + User receives a confirmation message
- + The application status is updated in the user's account settings

****Expected Result:**** User can successfully apply for monetization options

****Severity:**** Minor

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that Monetization Includes Advertising Revenue and Sponsored Content****

****Test Case ID:**** MON-002

****Test Case Title:**** Verify that monetization includes advertising revenue and sponsored content

****Test Case Description:**** This test case ensures that users can view and understand monetization options, including advertising revenue and sponsored content.

****Test Suite:**** Monetization Options

****Test Priority:**** High

****Preconditions:****

- + User is logged in to their account
- + User has applied for monetization options

****Test Data:**** No test data needed

****Test Steps:****

1. Navigate to the monetization options page
2. Click on the "Understanding Monetization" link
3. Review the available monetization options, including advertising revenue and sponsored content
4. Verify that the information is accurate and up-to-date

****Postconditions:****

- + User understands the monetization options available to them
- + The user has an accurate understanding of advertising revenue and sponsored content

****Expected Result:**** User can view and understand monetization options

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 3: Verify that Users Can Receive Payments through the Platform****

****Test Case ID:**** MON-003

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

****Test Case Description:**** This test case ensures that users can receive payments successfully through the platform.

****Test Suite:**** Monetization Options

****Test Priority:**** High

****Preconditions:****

- + User is logged in to their account
- + User has applied for monetization options
- + User has earned revenue (e.g. through advertising or sponsored content)

****Test Data:**** Test data for payment processing (e.g. fake payment information)

****Test Steps:****

1. Navigate to the payment page
2. Click on the "Request Payment" button
3. Verify that the payment is successfully processed
4. Ensure that the user's earnings are accurately updated in their account

****Postconditions:****

- + User's earnings are accurately updated in their account
- + User can view their payment history

****Expected Result:**** User can receive payments successfully

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 4: Verify that Detailed Earnings Reports are Available****

****Test Case ID:**** MON-004

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

****Test Case Description:**** This test case ensures that users can view detailed earnings reports through the platform.

****Test Suite:**** Monetization Options

****Test Priority:**** Medium

****Preconditions:****

- + User is logged in to their account
- + User has applied for monetization options
- + User has earned revenue (e.g. through advertising or sponsored content)

****Test Data:**** No test data needed

* **Test Steps:**

1. Navigate to the payment page
2. Click on the "View Earnings Report" button
3. Verify that the detailed earnings report is accurate and up-to-date
4. Review the report for any discrepancies or issues

* **Postconditions:**

- + User can view a detailed earnings report
- + The report accurately reflects the user's earnings

* **Expected Result:** User can view a detailed earnings report

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that Monetization Eligibility Requirements are Clearly Stated****

* **Test Case ID:** MON-005

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

* **Test Case Description:** This test case ensures that users are aware of and understand the requirements for monetization eligibility.

* **Test Suite:** Monetization Options

* **Test Priority:** High

* **Preconditions:**

- + User is logged in to their account
- + User has not applied for monetization options

* **Test Data:** No test data needed

* **Test Steps:**

1. Navigate to the monetization options page
2. Click on the "Eligibility Requirements" link
3. Review the requirements for monetization eligibility
4. Verify that the requirements are accurate and up-to-date

* **Postconditions:**

- + User understands the requirements for monetization eligibility
- + The user is aware of any necessary steps to become eligible

* **Expected Result:** User can view and understand monetization eligibility requirements

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

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:

Here are five detailed test cases for the Explore or Discover Tab feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that the Explore Tab shows trending and popular content****

*** **Test Case ID:** EX-001**

*** **Test Case Title:** Verify that the Explore Tab shows trending and popular content**

*** **Test Case Description:**** This test case verifies that the Explore Tab displays trending and popular content, including posts, accounts, and topics.

*** **Test Suite:** Explore Tab Feature Suite**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in

+ User has a profile

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Access the Explore Tab from the navigation menu
2. Verify that the Tab displays trending and popular content
3. Scroll down to check if the content is displayed in a clean and organized manner

* **Postconditions:** None

* **Expected Result:** The Explore Tab displays trending and popular content, including posts, accounts, and topics.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Acceptance Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that content recommendations are personalized based on user interests****

* **Test Case ID:** EX-002

* **Test Case Title:** Verify that content recommendations are personalized based on user interests

* **Test Case Description:** This test case verifies that content recommendations displayed in the Explore Tab are personalized based on the user's interests.

* **Test Suite:** Explore Tab Feature Suite

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in

+ User has a profile

+ User has expressed interests

* **Test Data:** Test interest and demographic data

* **Test Steps:**

1. Access the Explore Tab from the navigation menu
2. Verify that the recommendations displayed are based on the user's interests

3. Analyze the data displayed to ensure it matches the user's profile

* **Postconditions:** None

* **Expected Result:** Content recommendations displayed are personalized based on the user's interests.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can filter content by category or topic

* **Test Case ID:** EX-003

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

* **Test Case Description:** This test case verifies that users can filter content displayed in the Explore Tab by category or topic.

* **Test Suite:** Explore Tab Feature Suite

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in

+ User has a profile

+ Explore Tab is accessible

* **Test Data:** Test data for various categories and topics

* **Test Steps:**

1. Access the Explore Tab from the navigation menu

2. Verify that filters by category or topic are displayed

3. Test filtering content by category or topic

* **Postconditions:** None

* **Expected Result:** Users can filter content by category or topic and display the corresponding content.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that the explore tab updates regularly with new content

* **Test Case ID:** EX-004

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

* **Test Case Description:** This test case verifies that the Explore Tab updates regularly with new content and shows refresh rates higher than 2 minutes.

* **Test Suite:** Explore Tab Feature Suite

* **Test Priority:** Low

* **Preconditions:**

+ User is logged in

+ User has a profile

+ Explore Tab is accessible

* **Test Data:** Perform repeat tests for 30 minutes each, observing update rates.

* **Test Steps:**

1. Access the Explore Tab from the navigation menu

2. Verify that the tab updates regularly (refresh rate higher than 2 minutes)

3. Perform repeat tests for at least 30 minutes to validate the data displayed in that time

* **Postconditions:** None

* **Expected Result:** The Explore Tab updates regularly with new content and maintains a refresh rate higher than 2 minutes.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Performance Testing

* **Test Case Approach:** Negative

Test Case 5: Verify that users can save or follow content from the explore tab

* **Test Case ID:** EX-005

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

* **Test Case Description:** This test case verifies that users are able to save or follow content from the Explore Tab, and this feature should work when a non-logged-in user is checked.

* **Test Suite:** Explore Tab Feature Suite

* **Test Priority:** High

* **Preconditions:**

+ User logged in

+ User has a profile

+ Content from the Explore Tab is accessible

* **Test Data:** No test data needed

* **Test Steps:**

1. Access the Explore Tab from the navigation menu
2. Verify that save or follow buttons on the explore tab work
3. After clicking on either button, check if the button works accordingly, (Save: appears as an option for content clicked after, new option should appear.)

* **Postconditions:** None

* **Expected Result:** Users can save or follow content from the explore tab and, after clicking on save, it adds a new option to the user account based on the selected content.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Acceptance Testing

Data - 16

Project name -

Social Media Platform

Project description -

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

Feature name -

Accessibility Features

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five test cases that cover all the acceptance criteria for the Accessibility Features of the Social Media

Platform:

****Test Case 1: Verify that alt text is provided for images****

* ****Test Case ID:**** AC-001

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

* ****Test Case Description:**** This test case ensures that alt text is provided for images on the platform, making it accessible to users with visual impairments.

* ****Test Suite:**** Accessibility Features

* ****Test Priority:**** High

* ****Preconditions:**** User is logged in, image is uploaded to the platform

* ****Test Data:**** Image with alt text, image without alt text

* ****Test Steps:****

1. Log in to the platform as a user with visual impairment.
2. Upload an image with alt text to the platform.
3. Verify that the alt text is displayed when the image is hovered over.
4. Upload an image without alt text to the platform.
5. Verify that the image is not displayed when the image is hovered over.

* ****Postconditions:**** Image with alt text is displayed correctly, image without alt text is not displayed.

* ****Expected Result:**** Alt text is provided for images on the platform.

* ****Severity:**** Critical

* ****Priority:**** High

* ****Type of Testing:**** Functional Testing

* ****Test Case Approach:**** Positive

****Test Case 2: Verify that screen reader compatibility is available****

* **Test Case ID:** AC-002

* **Test Case Title:** Verify that screen reader compatibility is available

* **Test Case Description:** This test case ensures that the platform is compatible with screen readers, making it accessible to users with visual impairments.

* **Test Suite:** Accessibility Features

* **Test Priority:** High

* **Preconditions:** User is logged in, screen reader software is installed

* **Test Data:** None

* **Test Steps:**

1. Log in to the platform as a user with visual impairment.
2. Install and configure screen reader software.
3. Verify that the platform is navigable using the screen reader.
4. Verify that all content is read aloud by the screen reader.

* **Postconditions:** Platform is navigable using screen reader, all content is read aloud.

* **Expected Result:** Screen reader compatibility is available on the platform.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that keyboard shortcuts can be used for navigation****

* **Test Case ID:** AC-003

* **Test Case Title:** Verify that keyboard shortcuts can be used for navigation

* **Test Case Description:** This test case ensures that keyboard shortcuts are available for navigation on the platform, making it accessible to users with mobility impairments.

* **Test Suite:** Accessibility Features

* **Test Priority:** High

* **Preconditions:** User is logged in, keyboard shortcuts are enabled

* **Test Data:** None

* **Test Steps:**

1. Log in to the platform as a user with mobility impairment.
2. Enable keyboard shortcuts on the platform.
3. Verify that keyboard shortcuts can be used for navigation.
4. Verify that all content is reachable using keyboard shortcuts.

* **Postconditions:** Keyboard shortcuts are available for navigation, all content is reachable.

* **Expected Result:** Keyboard shortcuts can be used for navigation on the platform.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that color contrast and font size options are provided****

* **Test Case ID:** AC-004

* **Test Case Title:** Verify that color contrast and font size options are provided

* **Test Case Description:** This test case ensures that color contrast and font size options are available on the platform, making it accessible to users with visual impairments.

* **Test Suite:** Accessibility Features

* **Test Priority:** High

* **Preconditions:** User is logged in

* **Test Data:** None

* **Test Steps:**

1. Log in to the platform as a user with visual impairment.

2. Verify that color contrast and font size options are available.

3. Verify that the options can be adjusted.

4. Verify that the platform is accessible with the adjusted options.

* **Postconditions:** Color contrast and font size options are available, platform is accessible.

* **Expected Result:** Color contrast and font size options are provided on the platform.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that accessibility features are clearly documented****

* **Test Case ID:** AC-005

* **Test Case Title:** Verify that accessibility features are clearly documented

* **Test Case Description:** This test case ensures that accessibility features are clearly documented on the platform, making it easy for users to find and use the features.

* **Test Suite:** Accessibility Features

* **Test Priority:** High

* **Preconditions:** User is logged in

* **Test Data:** None

* **Test Steps:**

1. Log in to the platform as a user with visual impairment.

2. Search for accessibility features on the platform.

3. Verify that the features are clearly documented.

4. Verify that the documentation can be easily accessed.

* **Postconditions:** Accessibility features are clearly documented, documentation is easily accessible.

* **Expected Result:** Accessibility features are clearly documented on the platform.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 17

Project name -

Social Media Platform

Project description -

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

Feature name -

Safety and Moderation Tools

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

'Community guidelines are enforced consistently.']

Generated Test Cases:

Here are five test cases for the Safety and Moderation Tools feature in the Social Media Platform project:

****Test Case 1: Verify that Content Moderation Algorithms Filter Inappropriate Content****

****Test Case ID:** SMPS-TC-001**

****Test Case Title:** Verify that Content Moderation Algorithms Filter Inappropriate Content**

****Test Case Description:**** This test case ensures that the content moderation algorithms successfully filter out inappropriate content from user-generated content, maintaining a clean and respectful environment on the platform.

****Test Suite:** Safety and Moderation Tools**

****Test Priority:** High**

****Preconditions:****

- * The content moderation algorithms are enabled
- * The user has uploaded a piece of content containing explicit language or imagery
- * The user's account has a history of posting inappropriate content

****Test Data:** No test data needed**

****Test Steps:****

1. Create a new user account with a history of posting explicit content
2. Upload a piece of content (e.g., an image or video) containing explicit language or imagery
3. Verify that the content is flagged for moderation and subsequently removed from the platform

****Postconditions:****

- * The user's account is disabled or restricted for posting explicit content
- * The removed content is reflected in the user's activity logs

****Expected Result:** The content moderation algorithms successfully filter out inappropriate content and**

remove it from the platform.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that Users Can Report Harmful or Inappropriate Content****

****Test Case ID:**** SMPS-TC-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 harmful or inappropriate content on the platform, contributing to maintaining a safer online environment.

****Test Suite:**** Safety and Moderation Tools

****Test Priority:**** Medium

****Preconditions:****

* The user has encountered a piece of content they consider harmful or inappropriate

* The user is logged in to their account

****Test Data:**** User-generated content containing explicit language or imagery

****Test Steps:****

1. Create a new user account and navigate to a piece of content containing explicit language or imagery
2. Click on the report button to report the content
3. Verify that the content is flagged for moderation and subsequently processed for review

****Postconditions:****

* The reported content is highlighted in the user's activity logs

* The reporting user receives a confirmation message indicating that their report has been submitted

****Expected Result:**** Users can successfully report harmful or inappropriate content, leading to moderation and potential removal.

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 3: Verify that Blocking Capabilities Are Available to Users****

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

****Test Case Title:** Verify that Blocking Capabilities Are Available to Users**

****Test Case Description:** This test case ensures that users can effectively block other users from interacting with them on the platform, enhancing their online safety.**

****Test Suite:** Safety and Moderation Tools**

****Test Priority:** High**

****Preconditions:****

* The user has a legitimate reason to block another user

* The blocking user is logged in to their account

****Test Data:** A user account with a history of harassment or nuisance behavior**

****Test Steps:****

1. Create a new user account and navigate to a user with a history of harassment or nuisance behavior
2. Click on the block button to block the user
3. Verify that the blocked user is unable to interact with the blocking user

****Postconditions:****

* The blocking user receives a confirmation message indicating that the user has been blocked

* The blocked user is unable to view the blocking user's content or interact with them

****Expected Result:** Users can successfully block other users, preventing unwanted interactions.**

****Severity:** Major**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 4: Verify that Comment Filters Prevent Offensive Language****

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

****Test Case Title:** Verify that Comment Filters Prevent Offensive Language**

****Test Case Description:** This test case ensures that the comment filters effectively prevent the use of offensive language in comments, maintaining a respectful community on the platform.**

****Test Suite:** Safety and Moderation Tools**

****Test Priority:** Medium**

****Preconditions:****

* The comment filters are enabled

* A user attempts to post a comment containing explicit language

****Test Data:** A comment containing explicit language**

****Test Steps:****

1. Create a new user account and navigate to a piece of content

2. Enter a comment containing explicit language

3. Verify that the comment is flagged for moderation and prevented from being posted

****Postconditions:****

* The user receives a notification that their comment has been blocked

* The content owner receives a notification that a comment was blocked

****Expected Result:** Comment filters successfully prevent the use of offensive language in comments.**

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 5: Verify that Community Guidelines Are Enforced Consistently****

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

****Test Case Title:** Verify that Community Guidelines Are Enforced Consistently**

****Test Case Description:**** This test case ensures that the community guidelines are enforced consistently across the platform, ensuring a fair and inclusive environment for all users.

****Test Suite:** Safety and Moderation Tools**

****Test Priority:** High**

****Preconditions:****

* The community guidelines are well-defined and publicly accessible

* A user posts content or engages in behavior that violates the community guidelines

****Test Data:**** User-generated content or behavior violating community guidelines

****Test Steps:****

1. Create a new user account and navigate to a piece of content or engage in behavior violating community guidelines
2. Verify that the violating content or behavior is flagged for moderation and subsequently processed for review

3. Check that the enforcement of community guidelines is consistent and fair across the platform

****Postconditions:****

* The violating content or behavior is removed or modified to align with community guidelines

* The user receives a notification explaining the reason for moderation

****Expected Result:**** Community guidelines are enforced consistently and fairly across the platform.

****Severity:** Major**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

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:

Here are five detailed test cases for the Multi-Language Support feature, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can select their preferred interface language****

*** **Test Case ID:** ML-001**

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

*** **Test Case Description:**** This test case ensures that users can successfully select their preferred interface language from the available options.

*** **Test Suite:** Multi-Language Support**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in to the platform

+ User has not selected a preferred language before

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the platform as a user
2. Go to the settings or preferences page
3. Click on the language dropdown selection
4. Select a preferred language from the list
5. Verify that the selected language is displayed correctly in the interface

*** **Postconditions:**** Close the settings or preferences page

*** **Expected Result:**** The selected language is displayed correctly in the interface, and the changes are saved successfully

*** **Severity:** Critical**

*** **Priority:** High**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

****Test Case 2: Verify that the platform offers content translation features****

*** **Test Case ID:** ML-002**

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

*** **Test Case Description:** This test case ensures that the platform provides a content translation feature that allows users to display posts or messages in their preferred language.**

*** **Test Suite:** Multi-Language Support**

*** **Test Priority:** Medium**

*** **Preconditions:****

+ User is logged in to the platform

+ User has selected a preferred language

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the platform as a user

2. Post a message in a language other than the user's selected language

3. Click on the translate button

4. Verify that the message is translated correctly into the user's preferred language

5. Verify that other users can see the translated message when they view the post

*** **Postconditions:** Close the post creation page**

*** **Expected Result:** The message is translated correctly into the user's preferred language**

*** **Severity:** Major**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

****Test Case 3: Verify that posts and messages can be translated into different languages****

* **Test Case ID:** ML-003

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

* **Test Case Description:** This test case ensures that posts and messages can be successfully translated into different languages, supporting a wide range of languages.

* **Test Suite:** Multi-Language Support

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the platform

+ User has selected a preferred language

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user
2. Post a message in a language other than the user's selected language
3. Click on the translate button
4. Select a different language from the dropdown list
5. Verify that the message is translated correctly into the selected language
6. Verify that other users can see the translated message when they view the post

* **Postconditions:** Close the post creation page

* **Expected Result:** The message is successfully translated into the selected language

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that language preferences are saved and applied consistently****

* **Test Case ID:** ML-004

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

* **Test Case Description:** This test case ensures that the user's language preference is saved and applied consistently across the platform, including in posts, messages, and pages.

* **Test Suite:** Multi-Language Support

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the platform

+ User has selected a preferred language before

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user

2. Go to different pages and sections of the platform

3. Verify that the language preference is applied consistently across all pages and sections

4. Verify that the language preference is saved and applied correctly even after logging out and logging back in

* **Postconditions:** Close the platform

* **Expected Result:** The language preference is saved and applied consistently across the platform

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that the platform supports a wide range of languages****

* **Test Case ID:** ML-005

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

* **Test Case Description:** This test case ensures that the platform supports a wide range of languages,

including less common languages, and that users can select and use any supported language without issues.

* **Test Suite:** Multi-Language Support

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the platform

+ User has not selected a preferred language before

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user
2. Go to the settings or preferences page
3. Verify that a comprehensive list of languages is available
4. Select an uncommon language and verify that the platform loads correctly
5. Test translating a post or message into the selected language to ensure that it works correctly

* **Postconditions:** Close the settings or preferences page

* **Expected Result:** The platform supports a wide range of languages, and all languages are translated and displayed correctly in the interface

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Note that these test cases are comprehensive and cover all specified acceptance criteria. They are designed to ensure that the Multi-Language Support feature is thoroughly tested and meets the requirements specified in the user story.

Project name -

Social Media Platform

Project description -

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

Feature name -

Cross-Platform Integration

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five detailed test cases that cover the acceptance criteria for the Cross-Platform Integration feature:

****Test Case 1: Verify that users can link their accounts with other social media platforms****

Test Case ID: CCP-001

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

Test Case Description: As a user, I want to link my account with other social media platforms so that I can connect with friends and share content across multiple platforms. This test case verifies that users can successfully link their accounts with other social media platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- * User must have a valid account on the Social Media Platform.
- * User must have a valid account on the external social media platform (e.g. Facebook, Twitter, etc.).

Test Data: User credentials for Social Media Platform and external social media platform.

Test Steps:

1. Log in to the Social Media Platform with valid user credentials.
2. Click on the "Link Accounts" button next to the external social media platform (e.g. Facebook).
3. Enter the external social media platform credentials.
4. Confirm the link request.
5. Verify that the accounts are successfully linked.

Postconditions:

- * User can see the linked account in their profile settings.
- * User can share content from one platform to the other.

Expected Result: User can successfully link their accounts with other social media platforms.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that content can be cross-posted to other platforms****

Test Case ID: CCP-002

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

Test Case Description: As a user, I want to be able to share content from one social media platform to another so that I can reach a wider audience. This test case verifies that content can be successfully cross-posted to other platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- * User must have a valid account on the Social Media Platform.
- * User must have linked their account with other social media platforms (e.g. Facebook, Twitter).

Test Data: Valid content (text, image, video) to be shared.

Test Steps:

1. Log in to the Social Media Platform with valid user credentials.
2. Create and publish valid content (text, image, video).
3. Click on the "Share" button next to the external social media platform (e.g. Facebook).
4. Verify that the content is successfully cross-posted to the external social media platform.
5. Verify that the content is visible on the external social media platform.

Postconditions:

- * Content is visible on the external social media platform.
- * User receives notifications when content is shared or commented on.

Expected Result: Content can be successfully cross-posted to other platforms.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that users can sign in using single sign-on (SSO)****

Test Case ID: CCP-003

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

Test Case Description: As a user, I want to be able to sign in to the Social Media Platform using SSO so that I can simplify my login process. This test case verifies that users can successfully sign in using SSO.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User must have linked their account with other social media platforms (e.g. Google, Facebook).

Test Data: Valid credentials for SSO provider (e.g. Google, Facebook).

Test Steps:

1. Log in to the Social Media Platform as a guest user.
2. Click on the "Sign in with SSO" button.
3. Enter valid credentials for the SSO provider.
4. Verify that user is successfully signed in to the Social Media Platform.
5. Verify that user's account is linked to their SSO provider.

Postconditions:

* User is successfully signed in to the Social Media Platform.

* User's account is linked to their SSO provider.

Expected Result: Users can successfully sign in using single sign-on (SSO).

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that integration with external media sources is available****

Test Case ID: CCP-004

Test Case Title: Verify that integration with external media sources is available

Test Case Description: As a user, I want to be able to access and share content from external media sources so that I can enhance my user experience. This test case verifies that integration with external media sources is available.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

* User must have linked their account with other social media platforms (e.g. YouTube, Flickr).

Test Data: Valid embedded content from external media source (e.g. YouTube video, Flickr image).

Test Steps:

1. Log in to the Social Media Platform with valid user credentials.
2. Create and publish valid content (text, image).
3. Embed external media content (e.g. YouTube video, Flickr image) into content.
4. Verify that the external media content is visible and playable on the Social Media Platform.

Postconditions:

* External media content is visible and playable on the Social Media Platform.

* User receives notifications when content is shared or commented on.

Expected Result: Integration with external media sources is available.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that content sharing across platforms is seamless****

Test Case ID: CCP-005

Test Case Title: Verify that content sharing across platforms is seamless

Test Case Description: As a user, I want to be able to share content seamlessly across multiple platforms so that I can enhance my user experience. This test case verifies that content sharing across platforms is seamless.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User must have linked their account with other social media platforms (e.g. Facebook, Twitter).

Test Data: Valid content (text, image, video) to be shared.

Test Steps:

1. Log in to the Social Media Platform with valid user credentials.
2. Create and publish valid content (text, image, video).
3. Click on the "Share" button next to the external social media platform (e.g. Facebook).
4. Verify that the content is successfully shared on the external social media platform.
5. Verify that the content is visible on the external social media platform.
6. Verify that the user can like, comment, and share the content on the external social media platform.

Postconditions:

* Content is visible on the external social media platform.

* User receives notifications when content is shared or commented on.

Expected Result: Content sharing across platforms is seamless.

Severity: Critical

Priority: High

Type of Testing: Functional 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, as per the provided Test Case Creation Template:

****Test Case 1: Verify that Users can Earn Points and Badges for their Activities****

Test Case ID: GWR-001

Test Case Title: Verify that Users can Earn Points and Badges for their Activities

Test Case Description: This test case will verify that users can earn points and badges for their activities on the social media platform.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User has a valid account on the social media platform
- User has completed all necessary profile setup and verification steps

Test Data: User's activity history and reward points/badges system setup

Test Steps:

1. Log in to the social media platform with a valid user account.
2. Complete various activities such as posting content, sharing posts, commenting, and liking.
3. Review the user's profile to verify that they have earned points and badges for their activities.

4. Check that the points and badges system is accurately capturing user activity and awarding rewards.

Postconditions: Points and badges are accurately displayed on the user's profile.

Expected Result: User earns points and badges for their activities and can see a clear indication of their progress.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that Leaderboards Display Top Users based on Points or Achievements****

Test Case ID: GWR-002

Test Case Title: Verify that Leaderboards Display Top Users based on Points or Achievements

Test Case Description: This test case will verify that the leaderboard feature accurately displays top users based on points or achievements earned on the social media platform.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- Leaderboard feature is set up and enabled
- Users have earned points or achievements on the social media platform
- Leaderboard data has been populated

Test Data: User data and leaderboard setup

Test Steps:

1. Access the leaderboard feature to view the top users based on points or achievements.
2. Verify that the leaderboard is accurately displaying the correct users.
3. Review user pages to ensure that their points or achievements are accurately reflected on the leaderboard.
4. Check that the leaderboard is being updated in real-time as users earn or lose points/achievements.

Postconditions: Leaderboard reflects correct user data.

Expected Result: Leaderboard displays top users based on points or achievements.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that Challenges are Available for Users to Complete and Earn Rewards****

Test Case ID: GWR-003

Test Case Title: Verify that Challenges are Available for Users to Complete and Earn Rewards

Test Case Description: This test case will verify that challenges are available for users to complete and earn rewards on the social media platform.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- Challenges feature is set up and enabled
- Users have access to the challenges feature

Test Data: Challenge setup and related data

Test Steps:

1. Access the challenges feature to view available challenges.
2. Select a challenge to complete.
3. Complete the challenge requirements.
4. Verify that user is rewarded for completing the challenge.

Postconditions: Challenge completed, reward earned.

Expected Result: Users can complete challenges and earn rewards.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that Rewards can Include Virtual Goods, Discounts, or Exclusive Content****

Test Case ID: GWR-004

Test Case Title: Verify that Rewards can Include Virtual Goods, Discounts, or Exclusive Content

Test Case Description: This test case will verify that rewards can include virtual goods, discounts, or exclusive content.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- Rewards system is set up and enabled
- Users have access to rewards

Test Data: Reward setup and related data

Test Steps:

1. Review the rewards system to ensure it includes virtual goods, discounts, or exclusive content options.
2. Test redeeming rewards, verifying that rewards are properly dispensed.
3. Check that rewards are properly credited to user accounts.
4. Verify that rewards are only available to users who meet specific conditions.

Postconditions: Rewards are properly dispensed and credited to user accounts.

Expected Result: Rewards system includes virtual goods, discounts, or exclusive content.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that Users can View their Progress and Achievements in a Dedicated Section****

Test Case ID: GWR-005

Test Case Title: Verify that Users can View their Progress and Achievements in a Dedicated Section

Test Case Description: This test case will verify that users can view their progress and achievements in a dedicated section.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User has completed various activities on the social media platform
- User has earned points and badges

Test Data: User activity history and reward points/badges system setup

Test Steps:

1. Log in to the social media platform with a valid user account.
2. Access the dedicated section for viewing progress and achievements.
3. Verify that user's progress and achievements are accurately displayed.
4. Check that the dedicated section includes clear information about next steps for continued engagement.

Postconditions: User can view progress and achievements.

Expected Result: User can view their progress and achievements in a dedicated section.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 21

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Creation

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five test cases for the Task Creation feature, following the provided template:

****Test Case 1: Verify that users can enter a task title and description****

Test Case ID	TC001
---	---
Test Case Title	Verify that users can enter a task title and description
Test Case Description	This test case aims to validate that users can provide detailed information about a task, specifically the title and description. It will cover two acceptance criteria: tasks can be created, and users can add a description to tasks.
Test Suite	Task Creation
Test Priority	High
Preconditions	User is logged in, access to task creation page
Test Data	Task title: "Example Task", Task description: "This is an example task"
Test Steps	1. Log in to the system. 2. Click on the "Create Task" button. 3. Enter the task title "Example Task". 4. Enter the task description "This is an example task". 5. Click on the "Save Task" button.
Postconditions	Task is displayed on the task list with the entered title and description.
Expected Result	Users can successfully enter a task title and description.
Severity	Minor
Priority	High
Type of Testing	Functional Testing
Test Case Approach	Positive

****Test Case 2: Verify that users can set a due date for a task****

| Test Case ID | TC002 |

| --- | --- |

| Test Case Title | Verify that users can set a due date for a task |

| Test Case Description | This test case aims to validate that users can assign a due date to a task, covering one acceptance criterion: tasks can be assigned a due date. |

| Test Suite | Task Creation |

| Test Priority | High |

| Preconditions | User is logged in, access to task creation page |

| Test Data | Due date: Today + 7 days |

| Test Steps | 1. Log in to the system. 2. Click on the "Create Task" button. 3. Enter the task title "Example Task". 4. Set the due date to today + 7 days. 5. Click on the "Save Task" button. |

| Postconditions | Task is displayed on the task list with the assigned due date. |

| Expected Result | Users can successfully set a due date for a task. |

| Severity | Minor |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 3: Verify that users can assign a task to a team member****

| Test Case ID | TC003 |

| --- | --- |

| Test Case Title | Verify that users can assign a task to a team member |

| Test Case Description | This test case aims to validate that users can assign a task to a specific team member, covering one acceptance criterion: tasks can be assigned to team members. |

| Test Suite | Task Creation |

| Test Priority | High |

| Preconditions | User is logged in, access to task creation page, team member is added to the system |

| Test Data | Team member: "John Doe" |

| Test Steps | 1. Log in to the system. 2. Click on the "Create Task" button. 3. Enter the task title "Example Task". 4. Assign the task to "John Doe". 5. Click on the "Save Task" button. |

| Postconditions | Task is displayed on the task list with the assigned team member. |

| Expected Result | Users can successfully assign a task to a team member. |

| Severity | Minor |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 4: Verify that the system saves and displays created tasks****

| Test Case ID | TC004 |

| --- | --- |

| Test Case Title | Verify that the system saves and displays created tasks |

| Test Case Description | This test case aims to validate that created tasks are saved and displayed in the task list, covering two acceptance criteria: the system saves created tasks, and tasks are displayed. |

| Test Suite | Task Creation |

| Test Priority | Medium |

| Preconditions | User is logged in, access to task creation page |

| Test Data | Task title: "Example Task", Task description: "This is an example task", Due date: Today + 7 days, Team member: "John Doe" |

| Test Steps | 1. Create a new task with the described data. 2. Save the task. 3. Verify that the task is displayed in the task list. |

| Postconditions | Task is displayed on the task list with the entered data. |

| Expected Result | Created tasks are saved and displayed in the task list. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 5: Verify that the system validates required fields (task title)****

| Test Case ID | TC005 |

| --- | --- |

| Test Case Title | Verify that the system validates required fields (task title) |

| Test Case Description | This test case aims to validate that the system checks for the mandatory task title field when creating a new task, covering one acceptance criterion: the system validates required fields. |

| Test Suite | Task Creation |

| Test Priority | Medium |

| Preconditions | User is logged in, access to task creation page |

| Test Data | No task title entered |

| Test Steps | 1. Click on the "Create Task" button. 2. Do not enter a task title. 3. Attempt to save the task. |

| Postconditions | System displays error message indicating that task title is required. |

| Expected Result | System requires users to enter a task title. |

| Severity | Major |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Negative |

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Assignment

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are the test cases for the Task Assignment feature of the Task Management Tool, following the Test Case Creation Template:

****Test Case 1: Verify that Users can select team members to assign tasks****

*** **Test Case ID:** TC01**

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

* **Test Case Description:** This test case verifies that users can select team members to assign tasks from the task assignment page. The user should be able to browse through the list of team members and select one to assign the task.

* **Test Suite:** TaskAssignmentTestSuite

* **Test Priority:** High

* **Preconditions:** User is logged in, team members are created, and tasks are created.

* **Test Data:** Test user, teams, and tasks.

* **Test Steps:**

1. Log in as a user with admin privileges.
2. Navigate to the task assignment page.
3. Browse through the list of team members and select one to assign the task.
4. Verify that the selected team member is assigned to the task.

* **Postconditions:** Team member is successfully assigned to the task.

* **Expected Result:** The selected team member is assigned to the task, and the task is updated with the assignee's information.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that Assignees receive notifications about their tasks

* **Test Case ID:** TC02

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

* **Test Case Description:** This test case verifies that assignees receive notifications about their assigned tasks. The system should send a notification to the assignee after task assignment.

* **Test Suite:** TaskAssignmentTestSuite

* **Test Priority:** Medium

* **Preconditions:** User is logged in, team member is created, and task is created.

* **Test Data:** Test user, teams, and tasks.

* **Test Steps:**

1. Log in as a user with admin privileges.

2. Assign a task to a team member.

3. Verify that the assignee receives a notification about the assigned task.

4. Check the notification settings to verify that the notification is sent successfully.

* **Postconditions:** Assignee receives notification about the assigned task.

* **Expected Result:** The assignee receives a notification about the assigned task, and the notification is updated in the system.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that the system displays assigned tasks in the assignee's task list****

* **Test Case ID:** TC03

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

* **Test Case Description:** This test case verifies that the system displays assigned tasks in the assignee's task list. The assignee should see the assigned tasks in their task list.

* **Test Suite:** TaskAssignmentTestSuite

* **Test Priority:** Medium

* **Preconditions:** User is logged in, team member is created, and task is created.

* **Test Data:** Test user, teams, and tasks.

* **Test Steps:**

1. Log in as a user with admin privileges.
2. Assign a task to a team member.
3. Log in as the assignee and navigate to their task list.
4. Verify that the assigned task is displayed in the assignee's task list.

* **Postconditions:** Assignee sees the assigned task in their task list.

* **Expected Result:** The assigned task is displayed in the assignee's task list, and the task is updated with the assignee's information.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that users can reassign tasks to different team members

* **Test Case ID:** TC04

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

* **Test Case Description:** This test case verifies that users can reassign tasks to different team members. The user should be able to select a new team member to assign the task.

* **Test Suite:** TaskAssignmentTestSuite

* **Test Priority:** Medium

* **Preconditions:** User is logged in, team members are created, and tasks are created.

* **Test Data:** Test user, teams, and tasks.

* **Test Steps:**

1. Log in as a user with admin privileges.
2. Retrieve the previously created task and assignee.
3. Reassign the task to a different team member.

4. Verify that the task is updated with the new assignee's information.

* **Postconditions:** Task is successfully reassigned to a different team member.

* **Expected Result:** The task is updated with the new assignee's information, and the notification is sent to the new assignee.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that task assignment history is tracked****

* **Test Case ID:** TC05

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

* **Test Case Description:** This test case verifies that task assignment history is tracked in the system. The system should record the assignment history of each task.

* **Test Suite:** TaskAssignmentTestSuite

* **Test Priority:** Low

* **Preconditions:** User is logged in, team members are created, and tasks are created.

* **Test Data:** Test user, teams, and tasks.

* **Test Steps:**

1. Log in as a user with admin privileges.

2. Create multiple task assignments with the same assignee and different start and end dates.

3. Retrieve the task assignment history and verify that it is correctly recorded.

* **Postconditions:** Task assignment history is correctly recorded in the system.

* **Expected Result:** The task assignment history is correctly recorded in the system, and the history can be retrieved when needed.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Each test case is designed to cover one or more of the acceptance criteria for the Task Assignment feature. These test cases ensure that the system behaves correctly and meets the requirements specified in the user story.

Data - 23

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Deadline Reminders

Feature description -

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

Userstory title -

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

Acceptance criteria -

['Users can set reminders for tasks based on deadlines.', 'Reminders can be customized by urgency level.', 'The system sends notifications for upcoming deadlines.', 'Users can choose the delivery method for reminders

(e.g., email, in-app).', 'Reminder settings can be adjusted or disabled.']

Generated Test Cases:

Here are five detailed test cases for the Task Deadline Reminders feature:

****Test Case 1: Verify that users can set reminders for tasks based on deadlines****

Test Case ID	**TC-001-D-001**
---	---
Test Case Title	Verify that users can set reminders for tasks based on deadlines
Test Case Description	As a user, I want to set reminders for tasks based on deadlines so that I can complete tasks on time. This test case covers the acceptance criterion that users can set reminders for tasks based on deadlines.
Test Suite	Task Management Tool
Test Priority	High
Preconditions	1. The user has logged in to the Task Management Tool.
Test Data	No test data needed
Test Steps	1. Create a new task with a deadline set for, for example, 2 days from now.
	2. Attempt to save the task by clicking on the "Save" button.
	3. Verify that the reminder is displayed in the task details page.
	4. Check that the reminder is sent to the user's email account.
Postconditions	None
Expected Result	The reminder is displayed in the task details page and sent to the user's email account.
Severity	Major
Priority	High
Type of Testing	Functional Testing

| **Test Case Approach** | Positive |

Test Case 2: Verify that reminders can be customized by urgency level

| **Test Case ID** | **TC-001-D-002** |

| --- | --- |

| **Test Case Title** | Verify that reminders can be customized by urgency level |

| **Test Case Description** | As a user, I want to customize reminders by urgency level so that I can prioritize tasks based on their importance. This test case covers the acceptance criterion that reminders can be customized by urgency level. |

| **Test Suite** | Task Management Tool |

| **Test Priority** | Medium |

| **Preconditions** | 1. The user has logged in to the Task Management Tool. |

| **Test Data** | No test data needed |

| **Test Steps** | 1. Create a new task with a deadline set for, for example, 2 days from now. |

| | 2. Open the task details page and click on the "Edit Reminder" button. |

| | 3. Select the "High Priority" reminder option. |

| | 4. Verify that the reminder is displayed in the task details page with a corresponding urgency level. |

| **Postconditions** | None |

| **Expected Result** | The reminder is displayed in the task details page with a corresponding urgency level. |

| **Severity** | Minor |

| **Priority** | Medium |

| **Type of Testing** | Functional Testing |

| **Test Case Approach** | Positive |

Test Case 3: Verify that the system sends notifications for upcoming deadlines

| **Test Case ID** | **TC-001-D-003** |

| --- | --- |

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

| **Test Case Description** | As a user, I want to receive notifications for upcoming task deadlines so that I can stay on track. This test case covers the acceptance criterion that the system sends notifications for upcoming deadlines. |

| **Test Suite** | Task Management Tool |

| **Test Priority** | High |

| **Preconditions** | 1. The user has logged in to the Task Management Tool. |

| **Test Data** | No test data needed |

| **Test Steps** | 1. Create a new task with a deadline set for, for example, 2 days from now. |

| | 2. Open the task details page and verify that the reminder is displayed with a countdown to the deadline. |

| | 3. Verify that the user receives a notification email with the reminder details. |

| **Postconditions** | None |

| **Expected Result** | The user receives a notification email with the reminder details. |

| **Severity** | Major |

| **Priority** | High |

| **Type of Testing** | Functional Testing |

| **Test Case Approach** | Positive |

Test Case 4: Verify that users can choose the delivery method for reminders

| **Test Case ID** | **TC-001-D-004** |

| --- | --- |

| **Test Case Title** | Verify that users can choose the delivery method for reminders |

| **Test Case Description** | As a user, I want to choose the delivery method for reminders so that I can

receive notifications in the format I prefer. This test case covers the acceptance criterion that users can choose the delivery method for reminders. |

| **Test Suite** | Task Management Tool |

| **Test Priority** | Medium |

| **Preconditions** | 1. The user has logged in to the Task Management Tool. |

| **Test Data** | No test data needed |

| **Test Steps** | 1. Create a new task with a deadline set for, for example, 2 days from now. |

| | 2. Open the task details page and click on the "Edit Reminder" button. |

| | 3. Select the "In-App Notification" delivery method. |

| | 4. Verify that the reminder is displayed in the task details page and sent via in-app notification. |

| **Postconditions** | None |

| **Expected Result** | The reminder is displayed in the task details page and sent via in-app notification. |

| **Severity** | Minor |

| **Priority** | Medium |

| **Type of Testing** | Functional Testing |

| **Test Case Approach** | Positive |

Test Case 5: Verify that reminder settings can be adjusted or disabled

| **Test Case ID** | **TC-001-D-005** |

| --- | --- |

| **Test Case Title** | Verify that reminder settings can be adjusted or disabled |

| **Test Case Description** | As a user, I want to adjust or disable reminder settings so that I can customize the notification experience. This test case covers the acceptance criterion that reminder settings can be adjusted or disabled. |

| **Test Suite** | Task Management Tool |

| **Test Priority** | Medium |

| ****Preconditions**** | 1. The user has logged in to the Task Management Tool. |

| ****Test Data**** | No test data needed |

| ****Test Steps**** | 1. Create a new task with a reminder set for, for example, 2 days from now. |

| | 2. Open the task details page and click on the "Edit Reminder" button. |

| | 3. Click on the "Disable Reminders" button. |

| | 4. Verify that the reminder is no longer displayed in the task details page. |

| ****Postconditions**** | None |

| ****Expected Result**** | The reminder is no longer displayed in the task details page. |

| ****Severity**** | Minor |

| ****Priority**** | Medium |

| ****Type of Testing**** | Functional Testing |

| ****Test Case Approach**** | Positive |

Data - 24

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

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

high, medium, or low, based on their importance to the project or team.

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are five detailed test cases that cover the acceptance criteria for the Task Prioritization feature:

****Test Case 1: Verify that users can set priority levels for tasks****

Test Case ID: TC001

Test Case Title: Verify that users can set priority levels for tasks

Test Case Description: This test case ensures that users can assign high, medium, or low priority levels to tasks. It covers the requirement that users can set priority levels for tasks.

Test Suite: Task Management Tool - Task Prioritization

Test Priority: High

Preconditions:

- * The user has logged in to the task management tool.
- * The user has a task created and saved.

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool.
2. Navigate to the task creation page.
3. Enter a task description and save the task.

4. Click on the priority dropdown menu and select High, Medium, or Low.
5. Verify that the task's priority level has changed accordingly.

Postconditions:

- * The task is updated with the selected priority level.
- * The user is notified of the updated priority level.

Expected Result: The task's priority level is updated correctly, and the user is notified of the change.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that tasks can be sorted based on priority levels****

Test Case ID: TC002

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

Test Case Description: This test case ensures that tasks are sorted correctly based on their priority levels. It covers the requirement that tasks can be sorted based on priority levels.

Test Suite: Task Management Tool - Task Prioritization

Test Priority: Medium

Preconditions:

- * The user has logged in to the task management tool.
- * The user has multiple tasks created and saved, with different priority levels.

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool.
2. Navigate to the task list page.
3. Click on the sort dropdown menu and select Priority.

4. Verify that tasks are sorted in descending order (High, Medium, Low).

5. Re-sort the tasks in ascending order (Low, Medium, High).

Postconditions:

- * The tasks are sorted correctly based on their priority levels.

- * The user is notified of the updated task list.

Expected Result: Tasks are sorted correctly based on their priority levels.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that users can change the priority level of a task****

Test Case ID: TC003

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

Test Case Description: This test case ensures that users can update the priority level of an existing task. It covers the requirement that users can change the priority level of a task.

Test Suite: Task Management Tool - Task Prioritization

Test Priority: Medium

Preconditions:

- * The user has logged in to the task management tool.

- * The user has a task created and saved.

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool.

2. Navigate to the task detail page.

3. Click on the priority dropdown menu and select a new priority level (High, Medium, or Low).

4. Verify that the task's priority level has changed accordingly.

Postconditions:

- * The task is updated with the new priority level.
- * The user is notified of the updated priority level.

Expected Result: The task's priority level is updated correctly, and the user is notified of the change.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that the system visually distinguishes tasks based on priority****

Test Case ID: TC004

Test Case Title: Verify that the system visually distinguishes tasks based on priority

Test Case Description: This test case ensures that tasks with different priority levels are visually distinguished from each other. It covers the requirement that the system visually distinguishes tasks based on priority.

Test Suite: Task Management Tool - Task Prioritization

Test Priority: Low

Preconditions:

- * The user has logged in to the task management tool.
- * The user has multiple tasks created and saved, with different priority levels.

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool.
2. Navigate to the task list page.
3. Verify that tasks with different priority levels are visually distinguished (e.g., high-priority tasks might be displayed in bold or with a red label).

Postconditions:

- * The tasks are visually distinguished based on their priority levels.
- * The user is notified of the updated task list.

Expected Result: Tasks with different priority levels are visually distinguished from each other.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that tasks without a specified priority default to Medium****

Test Case ID: TC005

Test Case Title: Verify that tasks without a specified priority default to Medium

Test Case Description: This test case ensures that tasks without a specified priority default to the Medium priority level. It covers the requirement that tasks without a specified priority default to Medium.

Test Suite: Task Management Tool - Task Prioritization

Test Priority: Low

Preconditions:

- * The user has logged in to the task management tool.
- * The user has a task created and saved without a specified priority.

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool.
2. Navigate to the task detail page.
3. Verify that the task's priority level defaults to Medium.

Postconditions:

- * The task's priority level defaults to Medium.

* The user is notified of the updated priority level.

Expected Result: The task's priority level defaults to Medium correctly.

Severity: Trivial

Priority: Low

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:

Here are five detailed test cases that cover all the acceptance criteria for the "Subtasks" feature:

****Test Case 1: Verify Subtasks Can Be Created under Main Tasks****

* Test Case ID: TSC-001

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

* Test Case Description: This test case verifies that users can create subtasks under main tasks and that these subtasks are successfully added to the main task.

* Test Suite: Subtasks Module

* Test Priority: High

* Preconditions:

+ User has a valid account and is logged in.

+ The user has access to a task that can be broken down into subtasks.

* Test Data: No test data needed.

* Test Steps:

1. Log in to the Task Management Tool as a valid user.
2. Navigate to a task that can be broken down into subtasks.
3. Click on the "Add Subtask" button.
4. Enter the subtask details, including description and due date.
5. Click on the "Add Subtask" button to save the subtask.

* Postconditions:

+ The subtask is successfully added to the main task.

+ The subtask is displayed in the task list.

* Expected Result: The subtask is successfully created and added to the main task.

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

****Test Case 2: Verify Subtasks Can Have Their Own Due Dates and Assignees****

- * Test Case ID: TSC-002
- * Test Case Title: Verify that subtasks can have their own due dates and assignees.
- * Test Case Description: This test case verifies that subtasks can have their own due dates and assignees, and that these details are correctly displayed.
- * Test Suite: Subtasks Module
- * Test Priority: High
- * Preconditions:
 - + User has a valid account and is logged in.
 - + The user has access to a task that can be broken down into subtasks.
- * Test Data: No test data needed.
- * Test Steps:
 1. Log in to the Task Management Tool as a valid user.
 2. Navigate to a task that can be broken down into subtasks.
 3. Click on the "Add Subtask" button.
 4. Enter the subtask details, including description, due date, and assignee.
 5. Click on the "Add Subtask" button to save the subtask.
- * Postconditions:
 - + The subtask due date and assignee are displayed correctly.
 - + The subtask is correctly assigned to the specified user.
- * Expected Result: The subtask has its own due date and assignee, and these details are correctly displayed.

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

****Test Case 3: Verify System Tracks Subtask Progress****

- * Test Case ID: TSC-003
- * Test Case Title: Verify that the system tracks the progress of each subtask.
- * Test Case Description: This test case verifies that the system tracks the progress of each subtask, including completion status.
- * Test Suite: Subtasks Module
- * Test Priority: High
- * Preconditions:
 - + User has a valid account and is logged in.
 - + The user has access to a task that can be broken down into subtasks.
- * Test Data: No test data needed.
- * Test Steps:
 1. Log in to the Task Management Tool as a valid user.
 2. Navigate to a task that can be broken down into subtasks.
 3. Mark a subtask as complete.
 4. Verify that the subtask completion status is updated.
- * Postconditions:
 - + The subtask completion status is updated correctly.
 - + The system correctly calculates the main task completion status.
- * Expected Result: The system tracks the progress of each subtask and accurately updates completion status.
- * Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Verify Subtasks Can Be Marked as Complete Independently of Main Task****

* Test Case ID: TSC-004

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

* Test Case Description: This test case verifies that subtasks can be marked as complete independently of the main task, without affecting the main task completion status.

* Test Suite: Subtasks Module

* Test Priority: High

* Preconditions:

+ User has a valid account and is logged in.

+ The user has access to a task that can be broken down into subtasks.

* Test Data: No test data needed.

* Test Steps:

1. Log in to the Task Management Tool as a valid user.

2. Navigate to a task that can be broken down into subtasks.

3. Mark a subtask as complete.

4. Verify that the main task completion status is not updated.

* Postconditions:

+ The subtask completion status is updated correctly.

+ The main task completion status remains unchanged.

* Expected Result: Subtasks can be marked as complete independently of the main task.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Verify Dependencies between Subtasks Can Be Set****

* Test Case ID: TSC-005

* Test Case Title: Verify that dependencies between subtasks can be set.

* Test Case Description: This test case verifies that dependencies between subtasks can be set and correctly enforced by the system.

* Test Suite: Subtasks Module

* Test Priority: High

* Preconditions:

+ User has a valid account and is logged in.

+ The user has access to a task that can be broken down into subtasks with dependencies.

* Test Data: No test data needed.

* Test Steps:

1. Log in to the Task Management Tool as a valid user.

2. Navigate to a task that can be broken down into subtasks with dependencies.

3. Set a dependency between two subtasks.

4. Verify that the system correctly enforces the dependency.

* Postconditions:

+ The dependency between subtasks is correctly set.

+ The system correctly enforces the dependency.

* Expected Result: Dependencies between subtasks can be set and correctly enforced by the system.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

These test cases cover all the acceptance criteria for the "Subtasks" feature and ensure that the system behaves correctly when creating, managing, and tracking subtasks.

Data - 26

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Dependencies

Feature description -

Users can define dependencies between tasks, specifying the order in which tasks should be completed to ensure smooth workflow and project progression. Task dependencies help in identifying critical paths and managing interdependencies between tasks.

Userstory title -

As a user, I want to set task dependencies so that tasks are completed in the correct sequence.

Acceptance criteria -

['Users can set dependencies between tasks.', 'The system displays dependencies in the task details.', 'Tasks cannot be marked as complete if dependent tasks are not completed.', 'Users can update or remove dependencies as needed.', 'The system provides alerts or warnings if a dependent task is not completed.']

Generated Test Cases:

Here are the detailed test cases based on the provided Test Case Creation Template:

****Test Case 1: Verify that users can set dependencies between tasks****

****Test Case ID:** TCTD-001**

****Test Case Title:** Verify that users can set dependencies between tasks**

****Test Case Description:**** As a user, I want to set dependencies between tasks so that tasks are completed in the correct sequence. This test case verifies that the system allows users to establish dependencies between tasks.

****Test Suite:** Task Dependencies**

****Test Priority:** High**

****Preconditions:****

- * The user has created at least two tasks
- * The user has the necessary permissions to create and manage tasks

****Test Data:** No test data needed**

****Test Steps:****

1. Log in to the system and navigate to the task list
2. Select two tasks from the list and click on the "Add Dependency" button

3. Select the dependency type (e.g., "depends on") and specify the dependent task

4. Verify that the dependency is successfully added

****Postconditions:****

- * The dependency is visible in the task details

- * The system displays a message indicating that the dependency has been added

****Expected Result:**** The system successfully adds a dependency between two tasks

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that the system displays dependencies in the task details****

****Test Case ID:**** TCTD-002

****Test Case Title:**** Verify that the system displays dependencies in the task details

****Test Case Description:**** As a user, I want to see dependencies displayed in the task details so that I can understand the task sequence. This test case verifies that the system displays dependencies in the task details.

****Test Suite:** Task Dependencies**

****Test Priority:** Medium**

****Preconditions:****

- * The user has created at least two tasks with dependencies
- * The user has the necessary permissions to view task details

****Test Data:** No test data needed**

****Test Steps:****

1. Log in to the system and navigate to the task details
2. Verify that the dependencies are displayed in the task details
3. Check if the dependency direction (e.g., "depends on") is correctly displayed

****Postconditions:****

- * The dependencies are visible in the task details
- * The system displays a correct representation of the task sequence

****Expected Result:** The system correctly displays dependencies in the task details**

****Severity:** Minor**

****Priority:** Medium**

****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****

****Test Case ID:** TCTD-003**

****Test Case Title:** Verify that tasks cannot be marked as complete if dependent tasks are not completed**

****Test Case Description:** As a user, I want to ensure that tasks are completed in the correct sequence so that dependencies are respected. This test case verifies that tasks cannot be marked as complete if dependent tasks are not completed.**

****Test Suite:** Task Dependencies**

****Test Priority:** High**

****Preconditions:****

* The user has created at least two tasks with dependencies

* The dependent tasks are not completed

****Test Data:** No test data needed**

****Test Steps:****

1. Log in to the system and navigate to the task details
2. Try to marked the dependent task as completed
3. Verify that the system prevents the task from being marked as completed

****Postconditions:****

* The system displays an error message

* The task remains incomplete

****Expected Result:**** The system prevents tasks from being marked as completed if dependent tasks are not completed

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 4: Verify that users can update or remove dependencies as needed****

****Test Case ID:**** TCTD-004

****Test Case Title:**** Verify that users can update or remove dependencies as needed

****Test Case Description:**** As a user, I want to be able to update or remove dependencies as needed so that the task sequence can be adjusted. This test case verifies that users can update or remove dependencies.

****Test Suite:**** Task Dependencies

****Test Priority:**** Medium

****Preconditions:****

- * The user has created at least two tasks with dependencies
- * The user has the necessary permissions to manage task dependencies

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the system and navigate to the task list
2. Select the task with dependencies and click on the "Edit Dependency" button
3. Update the dependency (e.g., change the dependency type) and save
4. Verify that the dependency is successfully updated
5. Try to remove the dependency
6. Verify that the dependency is successfully removed

****Postconditions:****

- * The updated dependency is visible in the task details
- * The removed dependency is no longer visible in the task details

****Expected Result:**** The system allows users to update or remove dependencies successfully

****Severity:**** Minor

****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****

****Test Case ID:**** TCTD-005

****Test Case Title:**** Verify that the system provides alerts or warnings if a dependent task is not completed

****Test Case Description:**** As a user, I want to be notified when a dependent task is not completed so that I can take corrective action. This test case verifies that the system provides alerts or warnings.

****Test Suite:**** Task Dependencies

****Test Priority:**** High

****Preconditions:****

* The user has created at least two tasks with dependencies

* The dependent tasks are not completed

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the system and navigate to the task list
2. Verify that the system displays an alert or warning message indicating that a dependent task is not completed
3. Check if the alert or warning message is displayed on the task details page

****Postconditions:****

- * The system displays the alert or warning message
- * The user is notified of the incomplete dependent task

****Expected Result:**** The system correctly displays alerts or warnings if a dependent task is not completed

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

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 detailed test cases for the Task Comments and Collaboration feature:

****Test Case 1:****

****Test Case ID:**** TC-001

****Test Case Title:**** Verify that users can add comments to tasks

****Test Suite:**** Task Management Tool

****Test Priority:**** High

****Preconditions:****

- * User is logged in with a valid account
- * A task is created and assigned to the user
- * User has the necessary permissions to comment on tasks
- * Task has no existing comments

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the Task Management Tool with a valid account
2. Navigate to the task assigned to the user
3. Click the "Add Comment" button
4. Enter a comment in the text area
5. Click the "Save" button to submit the comment
6. Verify that the comment is displayed on the task

****Postconditions:****

- * Comment is saved successfully
- * Comment is displayed in chronological order
- * User is notified of new comment

****Expected Result:**** The user is able to add a comment to the task successfully.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2:****

****Test Case ID:**** TC-002

****Test Case Title:**** Verify that comments can include text, attachments, and links

****Test Suite:**** Task Management Tool

****Test Priority:**** High

****Preconditions:****

* User is logged in with a valid account

* A task is created and assigned to the user

* User has the necessary permissions to add comments with attachments and links

* Task has no existing comments

****Test Data:**** Test file for attachment and URL for link

****Test Steps:****

1. Log in to the Task Management Tool with a valid account
2. Navigate to the task assigned to the user
3. Click the "Add Comment" button

4. Enter a comment in the text area
5. Attach a file to the comment
6. Add a link to the comment
7. Click the "Save" button to submit the comment
8. Verify that the comment is displayed with attachment and link

****Postconditions:****

- * Comment is saved successfully
- * Attachment is uploaded successfully
- * Link is added successfully
- * Comment is displayed with attachment and link

****Expected Result:**** The user is able to add a comment with text, attachments, and links successfully.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 3:****

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

****Test Case Title:**** Verify that users can reply to comments

****Test Suite:**** Task Management Tool

****Test Priority:**** High

****Preconditions:****

- * User is logged in with a valid account
- * A task is created and assigned to the user
- * Task has existing comments
- * User has the necessary permissions to reply to comments

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the Task Management Tool with a valid account
2. Navigate to the task assigned to the user
3. View one of the existing comments
4. Click the "Reply" button below the comment
5. Enter a reply in the text area
6. Click the "Save" button to submit the reply
7. Verify that the reply is displayed below the original comment

****Postconditions:****

- * Reply is saved successfully
- * Reply is displayed below original comment
- * Comment thread is updated

****Expected Result:**** The user is able to reply to an existing comment successfully.

****Severity:**** Major

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 4:****

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

****Test Case Title:** Verify that users can edit their comments**

****Test Suite:** Task Management Tool**

****Test Priority:** High**

****Preconditions:****

- * User is logged in with a valid account
- * A task is created and assigned to the user
- * Task has existing comments
- * User is the author of one of the existing comments
- * User has the necessary permissions to edit comments

****Test Data:** No test data needed**

****Test Steps:****

1. Log in to the Task Management Tool with a valid account
2. Navigate to the task assigned to the user
3. View one of the existing comments
4. Click the "Edit" button below the comment
5. Edit the comment in the text area

6. Click the "Save" button to submit the edited comment

7. Verify that the edited comment is displayed

****Postconditions:****

* Comment is edited successfully

* Edited comment is displayed

* Comment thread is updated

****Expected Result:**** The user is able to edit their existing comment successfully.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 5:****

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

****Test Case Title:**** Verify that the system notifies users of new comments

****Test Suite:**** Task Management Tool

****Test Priority:**** High

****Preconditions:****

* User is logged in with a valid account

* A task is created and assigned to the user

* Task has no existing comments

* User has the necessary permissions to be notified of new comments

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the Task Management Tool with a valid account
2. Set up a new task with no existing comments
3. Have another user add a comment to the task
4. Log in to the test account and verify that a notification is displayed
5. Click on the notification to view the new comment

****Postconditions:****

- * User is notified of new comment
- * User can view the new comment
- * Notification is cleared after viewing

****Expected Result:**** The user is notified of new comments in a timely manner.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

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:

Here are five test cases that cover all acceptance criteria for the Feature: File Attachments:

****Test Case 1:****

Test Case ID: TC_FileAttachments_001

Test Case Title: Verify that users can attach multiple files to a task.

Test Case Description: This test case ensures that users can attach multiple files of different types to a task without any issues. It also covers the aspect of file addition and validation.

Test Suite: Task Management Tool - File Attachments Feature

Test Priority: Medium

Preconditions:

- * User is logged in with a valid account
- * User is on a task creation or editing page
- * Task has no existing attachments

Test Data: No test data needed

Test Steps:

1. Log in to the application with a valid account.
2. Click on the task creation or editing page.
3. Click on the file attachment button.
4. Select multiple files of different types (e.g., DOCX, PNG, and MP3).
5. Click on the "Add Attachment" button.

Postconditions:

- * Multiple files are successfully attached to the task.
- * Each attached file appears in the task's attachment list.

Expected Result: Multiple files are successfully attached to the task.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2:****

Test Case ID: TC_FileAttachments_002

Test Case Title: Verify that attachments can include various file types (documents, images, etc.).

Test Case Description: This test case ensures that users can attach various file types, including documents, images, and other resources, to a task without any issues.

Test Suite: Task Management Tool - File Attachments Feature

Test Priority: Medium

Preconditions:

- * User is logged in with a valid account
- * User is on a task creation or editing page

Test Data: Various file types (e.g., PDF, JPEG, MP3, and CSV)

Test Steps:

1. Log in to the application with a valid account.
2. Click on the task creation or editing page.
3. Click on the file attachment button.
4. Select a mixture of file types (e.g., a PDF, a JPEG, an MP3, and a CSV).
5. Click on the "Add Attachment" button.

Postconditions:

- * Various file types are successfully attached to the task.
- * Each attached file appears in the task's attachment list.

Expected Result: Various file types (documents, images, etc.) are successfully attached to the task.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3:****

Test Case ID: TC_FileAttachments_003

Test Case Title: Verify that users can view and download attachments.

Test Case Description: This test case ensures that users can view and download attachments successfully without any issues.

Test Suite: Task Management Tool - File Attachments Feature

Test Priority: Medium

Preconditions:

- * User is logged in with a valid account
- * User is on a task creation or editing page with attached files

Test Data: Valid attachment file types and sizes

Test Steps:

1. Log in to the application with a valid account.
2. Click on the task creation or editing page with attached files.
3. Click on the attached file to view it.
4. Verify that the file is correctly viewed.
5. Click on the "Download Attachment" button.
6. Verify that the file is successfully downloaded.

Postconditions:

- * Attached file is viewed and downloaded correctly.
- * User can access the downloaded file without issues.

Expected Result: Viewed and downloaded attachments work as expected.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4:****

Test Case ID: TC_FileAttachments_004

Test Case Title: Verify that the system limits the size of attachments to prevent data overload.

Test Case Description: This test case ensures that the system prevents large file attachments from exceeding the set limits to prevent data overload.

Test Suite: Task Management Tool - File Attachments Feature

Test Priority: High

Preconditions:

- * User is logged in with a valid account
- * System attachment size limit is set to 10MB

Test Data: A file larger than the set attachment size limit (e.g., 20MB)

Test Steps:

1. Log in to the application with a valid account.
2. Click on the task creation or editing page.
3. Try to add a file larger than the set attachment size limit (e.g., 20MB).
4. Verify that the system does not allow the excessive file attachment.

Postconditions:

- * Large file attachment is rejected by the system.
- * User is informed of the file size limitation.

Expected Result: File sizes that exceed attachment limits are not uploaded.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative

****Test Case 5:****

Test Case ID: TC_FileAttachments_005

Test Case Title: Verify that users can remove attachments if needed.

Test Case Description: This test case ensures that users can remove attachments without issues if they decide they no longer need them.

Test Suite: Task Management Tool - File Attachments Feature

Test Priority: Medium

Preconditions:

- * User is logged in with a valid account
- * User is on a task creation or editing page with attached files

Test Data: Valid attachment files

Test Steps:

1. Log in to the application with a valid account.
2. Click on the task creation or editing page with attached files.
3. Select the attachment to be removed.
4. Click the "Remove Attachment" button.
5. Verify that the attachment is removed.

Postconditions:

- * Attachment is successfully removed.
- * Removed attachment does not appear on the task's attachment list.

Expected Result: Attachment can be removed.

Severity: Major

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:

Here are five detailed test cases for the Recurring Tasks feature of the Task Management Tool, following the provided Test Case Creation Template:

****Test Case 1: Setting Recurrence Pattern****

* Test Case ID: RT-001

* Test Case Title: Verify that users can set tasks to recur at specified intervals.

* Test Suite: Recurring Tasks Feature

* Test Priority: High

* Preconditions:

+ User is logged in to the system.

+ Task is created with basic information (title, description, deadline).

* Test Data: Task title, recurring interval (daily, weekly, monthly).

* Test Steps:

1. Log in to the system and navigate to the task creation page.
2. Create a new task with basic information and select the recurring interval (daily, weekly, monthly).
3. Verify that the recurrence pattern is displayed correctly in the task details page.
4. Save the task and return to the task list page.
5. Verify that the task is marked as recurring in the task list page.

* Postconditions:

+ Task is created with recurring pattern saved.

+ Recurring task is listed in the task list page.

* Expected Result: Users can successfully set tasks to recur at specified intervals (daily, weekly, monthly).

- * Severity: Critical
- * Priority: High
- * Type of Testing: System Testing
- * Test Case Approach: Positive

****Test Case 2: System Generates Instances of Recurring Tasks****

- * Test Case ID: RT-002
- * Test Case Title: Verify that the system generates instances of recurring tasks automatically.
- * Test Suite: Recurring Tasks Feature
- * Test Priority: High
- * Preconditions:
 - + User is logged in to the system.
 - + Task is created with recurring pattern.
 - + System's date and time are set to the next occurrence of the recurring task.
- * Test Data: Task title, recurring interval (daily, weekly, monthly).
- * Test Steps:
 1. Log in to the system and navigate to the task creation page.
 2. Create a new task with basic information and select the recurring interval (daily, weekly, monthly).
 3. Save the task and return to the task list page.
 4. Verify that an instance of the recurring task is generated and listed in the task list page.
 5. Verify that the instance has the correct due date and time.
- * Postconditions:
 - + System generates instances of recurring tasks automatically.
 - + Recurring task instances are listed in the task list page.
- * Expected Result: The system correctly generates instances of recurring tasks automatically.
- * Severity: Critical

* Priority: High

* Type of Testing: System Testing

* Test Case Approach: Positive

****Test Case 3: Modifying Recurrence Pattern****

* Test Case ID: RT-003

* Test Case Title: Verify that users can modify the recurrence pattern of a recurring task.

* Test Suite: Recurring Tasks Feature

* Test Priority: Medium

* Preconditions:

+ User is logged in to the system.

+ Recurring task is created with recurrence pattern.

* Test Data: Recurrence pattern (daily, weekly, monthly), new due dates.

* Test Steps:

1. Log in to the system and navigate to the task details page of a recurring task.

2. Click the "Edit" button to modify the recurrence pattern.

3. Select a new recurrence pattern and due dates.

4. Save the changes and return to the task list page.

5. Verify that the recurrence pattern is updated correctly.

* Postconditions:

+ Recurrence pattern is updated correctly.

+ New instances of the recurring task are generated.

* Expected Result: Users can successfully modify the recurrence pattern of a recurring task.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Cancelling Recurrence****

* Test Case ID: RT-004

* Test Case Title: Verify that users can cancel the recurrence of a recurring task.

* Test Suite: Recurring Tasks Feature

* Test Priority: Medium

* Preconditions:

+ User is logged in to the system.

+ Recurring task is created with recurrence pattern.

* Test Data: None.

* Test Steps:

1. Log in to the system and navigate to the task details page of a recurring task.

2. Click the "Cancel Recurrence" button.

3. Verify that the recurrence pattern is cancelled correctly.

4. Verify that new instances of the recurring task are not generated.

* Postconditions:

+ Recurrence pattern is cancelled correctly.

+ New instances of the recurring task are not generated.

* Expected Result: Users can successfully cancel the recurrence of a recurring task.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Notifications for Upcoming Instances****

- * Test Case ID: RT-005
- * Test Case Title: Verify that the system notifies users of upcoming instances of recurring tasks.
- * Test Suite: Recurring Tasks Feature
- * Test Priority: Medium
- * Preconditions:
 - + User is logged in to the system.
 - + Recurring task is created with recurrence pattern.
 - + System's date and time are set to the day before the next occurrence of the recurring task.
- * Test Data: Task title, recurring interval (daily, weekly, monthly).
- * Test Steps:
 1. Log in to the system and navigate to the task list page.
 2. Verify that a notification is displayed for the upcoming instance of the recurring task.
 3. Verify that the notification contains the correct due date and time.
- * Postconditions:
 - + Notification is displayed correctly for the upcoming instance of the recurring task.
 - + Notification contains the correct due date and time.
- * Expected Result: The system correctly notifies users of upcoming instances of recurring tasks.
- * Severity: Major
- * Priority: Medium
- * 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:

Here are five test cases for the Task Tags and Labels feature:

****Test Case 1: Verify that users can add tags and labels to tasks****

* Test Case ID: T001

* Test Case Title: Verify that users can add tags and labels to tasks

* Test Case Description: This test case verifies that users can successfully add tags and labels to tasks.

* Test Suite: Task Management Module

* Test Priority: High

* Preconditions:

+ User is logged in to the system

+ Task exists in the system

* Test Data: No test data needed

* Test Steps:

1. Log in to the system as a user with permissions to add tags and labels.

2. Open a task in the task list view.

3. Click on the "Add Tag" button and enter a valid tag name.

4. Click on the "Add Label" button and enter a valid label name.

5. Verify that the tag and label are displayed prominently in the task list view.

* Postconditions:

+ The tag and label are stored in the database

+ The tag and label are associated with the task

* Expected Result: The user can successfully add tags and labels to tasks.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 2: Verify that tags can be customized and edited****

* Test Case ID: T002

* Test Case Title: Verify that tags can be customized and edited

* Test Case Description: This test case verifies that users can successfully customize and edit tags.

* Test Suite: Task Management Module

* Test Priority: High

* Preconditions:

- + User is logged in to the system
- + Task exists in the system with associated tags

* Test Data: No test data needed

* Test Steps:

1. Log in to the system as a user with permissions to edit tags.
2. Open a task in the task list view that has associated tags.
3. Click on the "Edit Tag" button next to a tag.
4. Type in a new tag name and click "Save".
5. Verify that the tag is updated successfully.

* Postconditions:

- + The tag is updated in the database
- + The tag is associated with the task

* Expected Result: The user can successfully customize and edit tags.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Verify that tasks can be filtered and sorted based on tags and labels****

* Test Case ID: T003

* Test Case Title: Verify that tasks can be filtered and sorted based on tags and labels

* Test Case Description: This test case verifies that users can successfully filter and sort tasks based on tags and labels.

* Test Suite: Task Management Module

* Test Priority: High

* Preconditions:

- + User is logged in to the system
- + Multiple tasks exist in the system with different tags and labels

* Test Data: No test data needed

* Test Steps:

1. Log in to the system as a user with permissions to filter and sort tasks.
2. Open the task list view.
3. Click on the "Filter" button next to the tag label.
4. Select a tag or label to filter by.
5. Verify that only tasks with the selected tag or label are displayed.

* Postconditions:

- + The task list is updated to show only tasks with the selected tag or label
- + The sorting is applied correctly

* Expected Result: The user can successfully filter and sort tasks based on tags and labels.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Verify that system displays tags and labels prominently in the task list****

* Test Case ID: T004

* Test Case Title: Verify that system displays tags and labels prominently in the task list

* Test Case Description: This test case verifies that the system displays tags and labels prominently in the task list view.

* Test Suite: Task Management Module

* Test Priority: Low

* Preconditions:

- + User is logged in to the system
- + Multiple tasks exist in the system with different tags and labels

* Test Data: No test data needed

* Test Steps:

1. Log in to the system as a user with permissions to view the task list.
2. Open the task list view.
3. Verify that tags and labels are displayed prominently in the task list view.
4. Verify that tags and labels are displayed in the correct position.

* Postconditions:

- + The task list displays tags and labels as expected

* Expected Result: The system displays tags and labels prominently in the task list.

* Severity: Minor

* Priority: Low

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Verify that users can remove tags and labels if needed****

* Test Case ID: T005

* Test Case Title: Verify that users can remove tags and labels if needed

* Test Case Description: This test case verifies that users can successfully remove tags and labels from tasks.

* Test Suite: Task Management Module

* Test Priority: High

* Preconditions:

- + User is logged in to the system
- + Task exists in the system with associated tags and labels

* Test Data: No test data needed

* Test Steps:

1. Log in to the system as a user with permissions to remove tags and labels.
2. Open a task in the task list view that has associated tags and labels.
3. Click on the "Remove Tag" button next to a tag.
4. Click on the "Remove Label" button next to a label.
5. Verify that the tag and label are removed successfully.

* Postconditions:

- + The tag and label are removed from the task
- + The tag and label are no longer associated with the task

* Expected Result: The user can successfully remove tags and labels from tasks.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

Data - 31

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Status Tracking

Feature description -

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

Userstory title -

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

Acceptance criteria -

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

Generated Test Cases:

Here are five detailed test cases for the Task Status Tracking feature:

****Test Case 1: Verify that users can update the status of tasks****

*** **Test Case ID:** TST-001**

*** **Test Case Title:** Verify that users can update the status of tasks**

*** **Test Case Description:**** In this test case, we verify that users can update the status of tasks accurately and successfully. We will test the status update feature with various types of tasks, including tasks with and without deadlines, tasks with and without descriptions, and tasks with and without attachments.

*** **Test Suite:** Task Status Tracking**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user is logged in to the task management tool

+ The user has created or selected a task to update

*** **Test Data:** No test data needed**

****Test Steps:****

1. Log in to the task management tool as a valid user
2. Create or select a task with a status of "To Do"
3. Click the status dropdown menu and select "In Progress"
4. Verify that the task status is updated to "In Progress"
5. Repeat steps 2-4 with different task statuses (e.g., "Completed")

****Postconditions:****

- + The task status is updated correctly
- + The user can view the updated task status

****Expected Result:**** The task status is updated successfully to the selected status.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that the system displays task status prominently in the task list****

****Test Case ID:**** TST-002

****Test Case Title:**** Verify that the system displays task status prominently in the task list

****Test Case Description:**** In this test case, we verify that the system displays the task status prominently in the task list. We will test the task list view with various task statuses and verify that the statuses are displayed correctly.

****Test Suite:**** Task Status Tracking

****Test Priority:**** Medium

****Preconditions:****

- + The user is logged in to the task management tool
- + The user has created or selected tasks with varying statuses

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a valid user
2. Create or select tasks with different statuses (e.g., "To Do", "In Progress", "Completed")
3. Navigate to the task list view
4. Verify that the task statuses are displayed prominently in the task list

* **Postconditions:**

- + The task statuses are displayed correctly in the task list
- + The user can view the task statuses in the task list

* **Expected Result:** The task statuses are displayed prominently in the task list.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that users can view the progress percentage of tasks****

* **Test Case ID:** TST-003

* **Test Case Title:** Verify that users can view the progress percentage of tasks

* **Test Case Description:** In this test case, we verify that users can view the progress percentage of tasks.

We will test the progress percentage display feature with tasks that have varying statuses and completion percentages.

* **Test Suite:** Task Status Tracking

* **Test Priority:** Medium

* **Preconditions:**

- + The user is logged in to the task management tool
- + The user has created or selected tasks with different statuses and completion percentages

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a valid user
2. Create or select tasks with different statuses (e.g., "To Do", "In Progress", "Completed") and completion percentages (e.g., 0%, 50%, 100%)
3. Navigate to the task view
4. Verify that the progress percentage is displayed correctly for each task

* **Postconditions:**

- + The progress percentages are displayed correctly for each task
- + The user can view the progress percentages for each task

* **Expected Result:** The progress percentages are displayed correctly for each task.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that the system provides notifications for status changes****

* **Test Case ID:** TST-004

* **Test Case Title:** Verify that the system provides notifications for status changes

* **Test Case Description:** In this test case, we verify that the system provides notifications for task status changes. We will test the notification feature with tasks that have their status updated by the user or other users.

* **Test Suite:** Task Status Tracking

* **Test Priority:** High

* **Preconditions:**

- + The user is logged in to the task management tool

+ The user has created or selected tasks with varying statuses

+ The system has notification settings enabled

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a valid user

2. Create or select tasks with different statuses (e.g., "To Do", "In Progress", "Completed")

3. Update the task status to a new status (e.g., "Completed" to "To Do")

4. Verify that a notification is sent to the user and other users with notification settings enabled

* **Postconditions:**

+ A notification is sent to the user and other users with notification settings enabled

+ The user and other users can view the notification

* **Expected Result:** A notification is sent to the user and other users with notification settings enabled.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that task status can be filtered and sorted****

* **Test Case ID:** TST-005

* **Test Case Title:** Verify that task status can be filtered and sorted

* **Test Case Description:** In this test case, we verify that task status can be filtered and sorted. We will test the filtering and sorting feature with tasks that have varying statuses.

* **Test Suite:** Task Status Tracking

* **Test Priority:** Medium

* **Preconditions:**

+ The user is logged in to the task management tool

+ The user has created or selected tasks with different statuses

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a valid user
2. Create or select tasks with different statuses (e.g., "To Do", "In Progress", "Completed")
3. Navigate to the task list view
4. Apply filters and sort options to view tasks with specific statuses
5. Verify that the tasks are displayed correctly after applying filters and sort options

* **Postconditions:**

- + The tasks are displayed correctly after applying filters and sort options
- + The user can view tasks with specific statuses after applying filters and sort options

* **Expected Result:** The tasks are displayed correctly after applying filters and sort options.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 32

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Filtering and Sorting

Feature description -

Users can filter and sort tasks based on various criteria such as due date, priority, assignee, or project. Filtering and sorting options help users focus on specific tasks or view tasks from different perspectives, enhancing task management and organization.

Userstory title -

As a user, I want to filter and sort tasks so that I can focus on specific tasks or view tasks from different perspectives.

Acceptance criteria -

['Users can filter tasks based on criteria such as due date, priority, assignee, or project.', 'Tasks can be sorted by various attributes.', 'The system provides multiple filter and sort options in the task list.', 'Users can save and apply custom filter and sort settings.', 'Filters and sorts do not affect the actual data, only the view.']

Generated Test Cases:

Here are 10 detailed test cases that cover the specified acceptance criteria for the Task Filtering and Sorting feature:

****Test Case 1: Verify that Users Can Filter Tasks by Due Date****

* ****Test Case ID:**** TF01

* ****Test Case Title:**** Verify that Users Can Filter Tasks by Due Date

* ****Test Case Description:**** This test case will verify that users can filter tasks based on due date and view only tasks with upcoming deadlines.

* ****Test Suite:**** Task Filtering and Sorting

* ****Test Priority:**** High

* ****Preconditions:****

+ User is logged in and has tasks assigned

+ Tasks have due dates set

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Click on the "Due Date" filter dropdown
4. Select a due date range (e.g., "Tomorrow" or "Next Week")
5. Verify that only tasks with due dates within the selected range are displayed
6. Verify that no tasks are displayed if the due date range is set to a date prior to the current date

* **Postconditions:** Refresh the task list

* **Expected Result:** Only tasks with due dates within the selected range are displayed

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that Users Can Sort Tasks by Priority****

* **Test Case ID:** TS02

* **Test Case Title:** Verify that Users Can Sort Tasks by Priority

* **Test Case Description:** This test case will verify that users can sort tasks based on priority and view tasks from high to low priority.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Medium

* **Preconditions:** User is logged in and has tasks assigned with different priority levels

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Click on the "Priority" sort dropdown
4. Select the "High to Low" sort option
5. Verify that tasks are displayed in order of highest to lowest priority

* **Postconditions:** Refresh the task list

* **Expected Result:** Tasks are displayed in order of highest to lowest priority

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Users Can Save Custom Filter and Sort Settings****

* **Test Case ID:** TF03

* **Test Case Title:** Verify that Users Can Save Custom Filter and Sort Settings

* **Test Case Description:** This test case will verify that users can save custom filter and sort settings and apply them in subsequent sessions.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** High

* **Preconditions:** User is logged in and has tasks assigned

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Filter tasks by due date and sort by priority
4. Click on the "Save Settings" button

5. Log out and log back in to the task management tool

6. Verify that the custom filter and sort settings are applied by default

* **Postconditions:** Refresh the task list

* **Expected Result:** Custom filter and sort settings are saved and applied in subsequent sessions

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that Filters and Sorts Do Not Affect the Actual Data****

* **Test Case ID:** TF04

* **Test Case Title:** Verify that Filters and Sorts Do Not Affect the Actual Data

* **Test Case Description:** This test case will verify that filters and sorts do not impact the underlying data and only affect the view.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Medium

* **Preconditions:** User is logged in and has tasks assigned

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool

2. Navigate to the task list

3. Filter tasks by due date

4. Create a new task outside of the filtered range

5. Verify that the new task is still displayed in the actual task list

* **Postconditions:** Refresh the task list

* **Expected Result:** The actual task list remains unchanged after applying filters and sorts

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that Users Can Clear Custom Filter and Sort Settings****

* **Test Case ID:** TF05

* **Test Case Title:** Verify that Users Can Clear Custom Filter and Sort Settings

* **Test Case Description:** This test case will verify that users can clear custom filter and sort settings stored in the task management tool.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Low

* **Preconditions:** User has saved custom filter and sort settings

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Click on the "Clear Settings" button
4. Verify that the custom filter and sort settings are cleared

* **Postconditions:** Refresh the task list

* **Expected Result:** Custom filter and sort settings are cleared

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 6: Verify that Task List Refreshes After Filter and Sort Changes****

* **Test Case ID:** TF06

* **Test Case Title:** Verify that Task List Refreshes After Filter and Sort Changes

* **Test Case Description:** This test case will verify that the task list refreshes automatically after filter and sort changes.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Medium

* **Preconditions:** User is logged in and has tasks assigned

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Filter tasks by due date
4. Verify that the task list refreshes and displays only tasks within the filtered range

* **Postconditions:** Refresh the task list

* **Expected Result:** Task list refreshes automatically after filter and sort changes

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 7: Verify that User Can Filter Tasks by Assignee****

* **Test Case ID:** TF07

* **Test Case Title:** Verify that User Can Filter Tasks by Assignee

* **Test Case Description:** This test case will verify that users can filter tasks based on the assignee.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** High

* **Preconditions:** User is logged in and has tasks assigned to multiple team members

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Click on the "Assignee" filter dropdown
4. Select a team member as the assignee
5. Verify that only tasks assigned to the selected team member are displayed

* **Postconditions:** Refresh the task list

* **Expected Result:** Only tasks assigned to the selected team member are displayed

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 8: Verify that User Can Sort Tasks by Project

* **Test Case ID:** TS08

* **Test Case Title:** Verify that User Can Sort Tasks by Project

* **Test Case Description:** This test case will verify that users can sort tasks based on the project.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Medium

* **Preconditions:** User is logged in and has tasks assigned to multiple projects

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Click on the "Project" sort dropdown
4. Select a project from the dropdown
5. Verify that tasks are displayed in order of the selected project

* **Postconditions:** Refresh the task list

* **Expected Result:** Tasks are displayed in order of the selected project

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 9: Verify that Task List Remains in Original Order When No Filter or Sort is Applied****

* **Test Case ID:** TF09

* **Test Case Title:** Verify that Task List Remains in Original Order When No Filter or Sort is Applied

* **Test Case Description:** This test case will verify that the task list remains in its original order when no filter or sort is applied.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Low

* **Preconditions:** User is logged in and has tasks assigned

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Verify that the task list remains in its original order

* **Postconditions:** Refresh the task list

* **Expected Result:** Task list remains in its original order when no filter or sort is applied

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 10: Verify that Task List Updates Dynamically as Filter and Sort Options Change

* **Test Case ID:** TF10

* **Test Case Title:** Verify that Task List Updates Dynamically as Filter and Sort Options Change

* **Test Case Description:** This test case will verify that the task list updates dynamically as filter and sort options change.

* **Test Suite:** Task Filtering and Sorting

* **Test Priority:** Medium

* **Preconditions:** User is logged in and has tasks assigned

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool
2. Navigate to the task list
3. Filter tasks by due date and sort by priority
4. Click on the "Clear Settings" button
5. Verify that the task list updates dynamically as the filter and sort options change

* **Postconditions:** Refresh the task list

* **Expected Result:** Task list updates dynamically as filter and sort options change

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Each of these test cases covers one or more acceptance criteria and ensures that the Task Filtering and Sorting feature is thoroughly tested, reducing the likelihood of defects and ensuring high-quality product delivery.

Data - 33

Project name -

Task Management Tool

Project description -

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

Feature name -

Time Tracking

Feature description -

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

Userstory title -

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Acceptance criteria -

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

Generated Test Cases:

Here are the 5 detailed test cases based on the provided Test Case Creation Template:

****Test Case 1: Verify that Users Can Start and Stop a Timer for Each Task****

* **Test Case ID:** TCT001

* **Test Case Title:** Verify that users can start and stop a timer for each task

* **Test Case Description:** As a user, I want to be able to start and stop a timer for each task to track the time spent. This test case will verify that the timer functionality works correctly.

* **Test Suite:** Time Tracking Module

* **Test Priority:** High

* **Preconditions:**

- + User has logged in to the system

- + User has a task created

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the system as a user
2. Create a new task
3. Click the "Start Timer" button next to the task
4. Wait for 30 seconds
5. Click the "Stop Timer" button next to the task

* **Postconditions:**

- + Timer should be stopped

- + Time spent on the task should be recorded

* **Expected Result:** The timer starts and stops correctly, and time spent on the task is recorded.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

* **Test Case 2: Verify that the System Records and Displays Total Time Spent on Tasks**

* **Test Case ID:** TCT002

* **Test Case Title:** Verify that the system records and displays total time spent on tasks

* **Test Case Description:** As a user, I want to see the total time spent on tasks to manage my time efficiently. This test case will verify that the system records and displays the total time spent.

* **Test Suite:** Time Tracking Module

* **Test Priority:** High

* **Preconditions:**

- + User has logged in to the system

- + User has multiple tasks created

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the system as a user
2. Create multiple tasks
3. Start and stop timers for each task
4. Verify that the total time spent on tasks is displayed correctly

* **Postconditions:**

- + Total time spent on tasks should be accurate

* **Expected Result:** The system accurately records and displays the total time spent on tasks.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Users Can Manually Enter or Adjust Time Entries****

* **Test Case ID:** TCT003

* **Test Case Title:** Verify that users can manually enter or adjust time entries

* **Test Case Description:** As a user, I want to be able to manually enter or adjust time entries for tasks to ensure accuracy. This test case will verify that users can manually enter or adjust time entries.

* **Test Suite:** Time Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

- + User has logged in to the system

- + User has a task created

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the system as a user
2. Create a new task
3. Manually enter a time entry for the task
4. Adjust the time entry
5. Verify that the time entry is updated correctly

* **Postconditions:**

- + Time entry should be updated correctly

* **Expected Result:** Users can manually enter or adjust time entries correctly.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that Time Tracking Data is Stored and Can be Exported****

* **Test Case ID:** TCT004

* **Test Case Title:** Verify that time tracking data is stored and can be exported

* **Test Case Description:** As a user, I want to be able to export time tracking data to use in other tools.

This test case will verify that time tracking data is stored and can be exported.

* **Test Suite:** Time Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

+ User has logged in to the system

+ User has multiple tasks with time tracking data

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the system as a user

2. Export time tracking data

3. Verify that the exported data is accurate

* **Postconditions:**

+ Time tracking data should be exported correctly

* **Expected Result:** Time tracking data is stored and can be exported correctly.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that the System Provides Reports on Time Spent per Task or Project****

* **Test Case ID:** TCT005

* **Test Case Title:** Verify that the system provides reports on time spent per task or project

* **Test Case Description:** As a user, I want to see reports on time spent per task or project to manage my resources effectively. This test case will verify that the system provides reports on time spent per task or project.

* **Test Suite:** Time Tracking Module

* **Test Priority:** High

* **Preconditions:**

+ User has logged in to the system

+ User has multiple tasks with time tracking data

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the system as a user

2. View reports on time spent per task or project

3. Verify that the reports are accurate

* **Postconditions:**

+ Reports should be accurate

* **Expected Result:** The system provides accurate reports on time spent per task or project.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** 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:

Here are 5 comprehensive test cases for the Task Templates feature, carefully crafted to cover each acceptance criterion and ensure a high-quality software product:

****Test Case 1: Verify that Users Can Create and Save Task Templates****

*** **Test Case ID:** TT-001**

*** **Test Case Title:** Verify that Users Can Create and Save Task Templates**

*** **Test Case Description:** As a user, I want to create and save task templates so that I can streamline task creation and ensure consistency.**

*** **Test Suite:** Task Templates**

* **Test Priority:** High

* **Preconditions:** User is logged in, Task Management Tool is open

* **Test Data:** No test data needed

* **Test Steps:**

1. Navigate to Task Management Tool and click on "Templates" tab

2. Click on "Create Template" button

3. Fill in template name, description, and other required fields

4. Save the template

* **Postconditions:** Template is saved, available for selection in Task Management Tool

* **Expected Result:** Successfully created and saved task template

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that Templates Can Include Predefined Task Structures, Descriptions, Assignees, and Due Dates****

* **Test Case ID:** TT-002

* **Test Case Title:** Verify that Templates Can Include Predefined Task Structures, Descriptions, Assignees, and Due Dates

* **Test Case Description:** As a user, I want to create templates with predefined task structures, descriptions, assignees, and due dates so that I can ensure consistency in task creation.

* **Test Suite:** Task Templates

* **Test Priority:** Medium

* **Preconditions:** User is logged in, Task Management Tool is open, template is created

* **Test Data:** Template with predefined task structure, description, assignee, and due date

* **Test Steps:**

1. Create a new task using the template
2. Verify that task structure, description, assignee, and due date from the template are populated
3. Save the task

* **Postconditions:** Task created with predefined template details

* **Expected Result:** Task created with predefined template details

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Users Can Apply Templates to New Tasks or Projects****

* **Test Case ID:** TT-003

* **Test Case Title:** Verify that Users Can Apply Templates to New Tasks or Projects

* **Test Case Description:** As a user, I want to apply templates to new tasks or projects so that I can streamline task creation.

* **Test Suite:** Task Templates

* **Test Priority:** High

* **Preconditions:** User is logged in, Task Management Tool is open, template is created

* **Test Data:** No test data needed

* **Test Steps:**

1. Create a new task or project
2. Select a template from the available list
3. Fill in any required fields
4. Save the task or project

* **Postconditions:** Task or project created with template details

* **Expected Result:** Successfully applied template to new task or project

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that Templates Can Be Edited or Deleted as Needed

* **Test Case ID:** TT-004

* **Test Case Title:** Verify that Templates Can Be Edited or Deleted as Needed

* **Test Case Description:** As a user, I want to edit or delete templates as needed so that I can maintain up-to-date and relevant task templates.

* **Test Suite:** Task Templates

* **Test Priority:** Medium

* **Preconditions:** User is logged in, Task Management Tool is open, template is created

* **Test Data:** Template to be edited or deleted

* **Test Steps:**

1. Navigate to template list
2. Select a template to edit or delete
3. Edit or delete the template
4. Save the changes

* **Postconditions:** Template updated or deleted

* **Expected Result:** Successfully edited or deleted template

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that the System Displays a List of Available Templates for Selection****

*** **Test Case ID:** TT-005**

*** **Test Case Title:** Verify that the System Displays a List of Available Templates for Selection**

*** **Test Case Description:** As a user, I want to see a list of available templates for selection so that I can easily apply templates to new tasks or projects.**

*** **Test Suite:** Task Templates**

*** **Test Priority:** High**

*** **Preconditions:** User is logged in, Task Management Tool is open**

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Navigate to Task Management Tool
2. Click on "Templates" tab
3. Verify that a list of available templates is displayed

*** **Postconditions:** List of available templates displayed**

*** **Expected Result:** List of available templates is displayed**

*** **Severity:** Minor**

*** **Priority:** High**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

These test cases cover all the acceptance criteria for the Task Templates feature, ensuring that the product meets the desired functionality and quality standards. By executing these test cases, we can identify potential defects early in the development cycle and contribute to a high-quality software product.

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:

Here are five detailed test cases for the Calendar Integration feature, each covering an acceptance criterion from the user story:

****Test Case 1: Verify that Users can Sync Tasks with External Calendars****

* **Test Case ID:** CM-001-001

* **Test Case Title:** Verify that Users can Sync Tasks with External Calendars

* **Test Case Description:** As a user, I want to sync tasks with external calendars (e.g., Google Calendar, Outlook) so that I can manage my time and avoid scheduling conflicts.

* **Test Suite:** Calendar Integration

* **Test Priority:** High

* **Preconditions:**

+ User has an external calendar account (e.g., Google Calendar, Outlook)

+ User is logged in to the Task Management Tool

* **Test Data:** No specific test data needed. System settings will be used to configure external calendar integration.

* **Test Steps:**

1. Log in to the Task Management Tool with valid credentials.
2. Navigate to the "Settings" menu and select the "Calendar Integration" option.
3. Enter the external calendar credentials and authorize the Task Management Tool to access the calendar.
4. Verify that the external calendar is successfully connected to the Task Management Tool.

* **Postconditions:**

+ External calendar is updated to reflect synced tasks.

+ Tasks are displayed in the external calendar view.

* **Expected Result:** Tasks are successfully synced with the external calendar.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Duplication

Feature description -

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

Userstory title -

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

Acceptance criteria -

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

Generated Test Cases:

Here are five detailed test cases, each covering one or more of the acceptance criteria from the user story "Task Duplication".

****Test Case 1: Verify that users can duplicate tasks or task templates****

*** **Test Case Title:**** Verify that users can duplicate tasks or task templates

*** **Test Case Description:**** This test case verifies that users can successfully duplicate tasks or task

templates using the task duplication feature. The test also checks that the duplicated task retains all original details, including descriptions, assignees, and deadlines.

* **Test Suite:** Task Management Tool, Task Duplication

* **Test Priority:** High

* **Preconditions:**

- + The user has logged in with the necessary privileges

- + A task or task template exists in the system

* **Test Data:** A test task with a description, assignee, and deadline

* **Test Steps:**

1. Log in to the system with the necessary privileges
2. Navigate to the task list and select a task or task template to duplicate
3. Click on the "Duplicate" button
4. Confirm that the duplicated task has been saved and displayed in the task list

* **Postconditions:**

- + The duplicated task is listed in the task list

- + The duplicated task retains all original details

* **Expected Result:** The duplicated task is successfully created and visible in the task list, retaining all original details.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 2: Verify that the duplicated task retains all original details, including descriptions, assignees, and deadlines

* **Test Case Title:** Verify that the duplicated task retains all original details, including descriptions,

assignees, and deadlines

* **Test Case Description:** This test case verifies that the duplicated task retains all original details, including descriptions, assignees, and deadlines.

* **Test Suite:** Task Management Tool, Task Duplication

* **Test Priority:** High

* **Preconditions:**

+ The user has logged in with the necessary privileges

+ A task exists in the system with a description, assignee, and deadline

* **Test Data:** A test task with a description, assignee, and deadline

* **Test Steps:**

1. Log in to the system with the necessary privileges

2. Navigate to the task list and select a task to duplicate

3. Click on the "Duplicate" button

4. Check that the duplicated task retains the original description, assignee, and deadline

* **Postconditions:**

+ The duplicated task has the same description, assignee, and deadline as the original task

* **Expected Result:** The duplicated task retains all original details, including descriptions, assignees, and deadlines.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Test Case 3: Verify that users can modify the duplicated task if needed

* **Test Case Title:** Verify that users can modify the duplicated task if needed

* **Test Case Description:** This test case verifies that users can successfully modify the duplicated task.

* **Test Suite:** Task Management Tool, Task Duplication

* **Test Priority:** High

* **Preconditions:**

+ The user has logged in with the necessary privileges

+ A duplicated task exists in the system

* **Test Data:** A duplicated task

* **Test Steps:**

1. Log in to the system with the necessary privileges

2. Navigate to the task list and select the duplicated task to modify

3. Modify the task details (e.g., description, assignee, deadline)

4. Save the changes

* **Postconditions:**

+ The modified task has the new details

* **Expected Result:** The duplicated task can be successfully modified.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

****Test Case 4: Verify that the system clearly indicates that a task has been duplicated****

* **Test Case Title:** Verify that the system clearly indicates that a task has been duplicated

* **Test Case Description:** This test case verifies that the system provides a clear indication when a task has been duplicated.

* **Test Suite:** Task Management Tool, Task Duplication

* **Test Priority:** High

* **Preconditions:**

+ The user has logged in with the necessary privileges

+ A duplicated task exists in the system

* **Test Data:** A duplicated task

* **Test Steps:**

1. Log in to the system with the necessary privileges

2. Navigate to the task list and select the duplicated task

3. Check that the system provides a clear indication that the task has been duplicated (e.g., a ribbon, a popup)

* **Postconditions:**

+ The system provides a clear indication that the task has been duplicated

* **Expected Result:** The system clearly indicates that the task has been duplicated.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

****Test Case 5: Verify that duplicated tasks are saved and displayed in the task list****

* **Test Case Title:** Verify that duplicated tasks are saved and displayed in the task list

* **Test Case Description:** This test case verifies that duplicated tasks are saved and displayed in the task list.

* **Test Suite:** Task Management Tool, Task Duplication

* **Test Priority:** High

* **Preconditions:**

+ The user has logged in with the necessary privileges

+ A duplicated task exists in the system

* **Test Data:** A duplicated task

* **Test Steps:**

1. Log in to the system with the necessary privileges
2. Navigate to the task list
3. Check that the duplicated task is listed in the task list

* **Postconditions:**

+ The duplicated task is listed in the task list

* **Expected Result:** The duplicated task is successfully saved and displayed in the task list.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional

* **Test Case Approach:** Positive

Data - 37

Project name -

Task Management Tool

Project description -

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

Feature name -

Task History

Feature description -

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

Userstory title -

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

Acceptance criteria -

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

Generated Test Cases:

Here are five detailed test cases that thoroughly cover each acceptance criterion for the Task History feature:

****Test Case 1: Verify that users can view a detailed history of changes to tasks****

*** **Test Case ID:** TH-001**

*** **Test Case Title:** Verify that users can view a detailed history of changes to tasks**

*** **Test Case Description:**** This test case ensures that users can view a detailed history of changes made to tasks, including modifications, comments, and other relevant activity.

*** **Test Suite:** Task Management Tool**

*** **Test Priority:** High**

*** **Preconditions:****

+ User has created a task

+ User has made changes to the task (e.g. updated description, added attachments)

*** **Test Data:** Task with multiple changes made**

*** **Test Steps:****

1. Access the task details page and navigate to the Task History section
2. Verify that the history includes a list of changes made to the task
3. Verify that each change includes a description of the modification and the timestamp
4. Verify that the history includes comments made by the user and other team members

*** **Postconditions:****

+ Task history section is visible on the task details page

+ User can view all changes made to the task

****Expected Result:**** The task history section displays all changes made to the task, including modifications, comments, and timestamps.

****Severity:**** Blocker

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that the history includes modifications, comments, and other relevant activity****

****Test Case ID:**** TH-002

****Test Case Title:**** Verify that the history includes modifications, comments, and other relevant activity

****Test Case Description:**** This test case ensures that the task history includes all relevant activity, including modifications, comments, and other updates made to the task.

****Test Suite:**** Task Management Tool

****Test Priority:**** Medium

****Preconditions:****

+ User has created a task

+ User has made multiple changes to the task (e.g. updated description, added attachments, commented on the task)

****Test Data:**** Task with multiple changes made, including modifications and comments

****Test Steps:****

1. Access the task details page and navigate to the Task History section
2. Verify that the history includes a list of all changes made to the task
3. Verify that each change includes a description of the modification and the timestamp
4. Verify that the history includes all comments made by the user and other team members

* **Postconditions:**

+ Task history section is visible on the task details page

+ User can view all changes made to the task

* **Expected Result:** The task history section displays all changes made to the task, including modifications, comments, and timestamps.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can filter the history by type of change or date

* **Test Case ID:** TH-003

* **Test Case Title:** Verify that users can filter the history by type of change or date

* **Test Case Description:** This test case ensures that users can filter the task history by type of change or date.

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* **Preconditions:**

+ User has created a task

+ User has made multiple changes to the task (e.g. updated description, added attachments, commented on the task)

* **Test Data:** Task with multiple changes made

* **Test Steps:**

1. Access the task details page and navigate to the Task History section

2. Open the filter dropdown menu and select a filter type (e.g. "Changes", "Comments", "Attachments")

3. Verify that only the selected type of change is displayed

4. Open the filter dropdown menu and select a date range
5. Verify that only changes made within the selected date range are displayed

* **Postconditions:**

+ User can filter task history by type of change or date

* **Expected Result:** The task history section allows users to filter changes by type or date.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that the system provides timestamps for all changes****

* **Test Case ID:** TH-004

* **Test Case Title:** Verify that the system provides timestamps for all changes

* **Test Case Description:** This test case ensures that the system provides a timestamp for each change made to a task.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* **Preconditions:**

+ User has created a task

+ User has made multiple changes to the task (e.g. updated description, added attachments)

* **Test Data:** Task with multiple changes made

* **Test Steps:**

1. Access the task details page and navigate to the Task History section
2. Verify that each change includes a timestamp
3. Verify that the timestamps are accurate and match the time of each change

* **Postconditions:**

+ Task history section is visible on the task details page

+ User can view all changes made to the task

* **Expected Result:** The task history section displays timestamps for all changes made to the task.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that task history is accessible from the task details page

* **Test Case ID:** TH-005

* **Test Case Title:** Verify that task history is accessible from the task details page

* **Test Case Description:** This test case ensures that the task history is accessible from the task details page.

* **Test Suite:** Task Management Tool

* **Test Priority:** Low

* **Preconditions:**

+ User has created a task

* **Test Data:** Task with no changes made

* **Test Steps:**

1. Access the task details page
2. Verify that the Task History section is visible and accessible
3. Verify that the user can view the task history

* **Postconditions:**

+ Task history section is visible on the task details page

* **Expected Result:** The task history section is accessible from the task details page.

* **Severity:** Trivial

* **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:

Here are five test cases for the Task Management Tool's Task Notifications feature, each covering one or more acceptance criteria from the user story:

****Test Case 1: Verify that users receive notifications for task updates****

- * Test Case ID: NTN-001

- * Test Case Title: Verify that users receive notifications for task updates

- * Test Case Description: This test case aims to ensure that users receive notifications when a task is updated by another user. The test case will cover the acceptance criteria that notifications are sent for task updates and that the system logs notification history.

- * Test Suite: Task Management Tool - Task Notifications

- * Test Priority: High

- * Preconditions:

- * The user is logged in to the Task Management Tool

- * A task is created and assigned to another user

- * The task is updated by the assigned user

- * Test Data: No test data needed

- * Test Steps:

- 1. Log in to the Task Management Tool as the user who was assigned the task

- 2. Navigate to the task and update it by changing its status or description

- 3. Verify that the user receives a notification for the task update

- 4. Check the notification history to ensure that the update is logged

- * Postconditions:

- * The notification should be delivered to the user via email, mobile push notifications, or in-app alerts

- * The update should be visible in the task details
- * Expected Result: The user receives a notification for the task update, and the update is logged in the notification history.
- * Severity: Critical
- * Priority: High
- * Type of Testing: Functional Testing
- * Test Case Approach: Positive

****Test Case 2: Verify that users can customize notification settings based on preferences****

- * Test Case ID: NTN-002
- * Test Case Title: Verify that users can customize notification settings based on preferences
- * Test Case Description: This test case aims to ensure that users can customize their notification settings to receive notifications based on their preferences.
- * Test Suite: Task Management Tool - Task Notifications
- * Test Priority: Medium
- * Preconditions:
 - * The user is logged in to the Task Management Tool
- * Test Data: No test data needed
- * Test Steps:
 1. Log in to the Task Management Tool as the user
 2. Navigate to the notification settings page
 3. Customize the notification settings to receive notifications based on specific criteria (e.g., task updates, comments, mentions)
 4. Verify that the user receives notifications based on their customized settings
- * Postconditions:

- * The notification settings should be saved and applied
- * The user should receive notifications based on their customized settings
- * Expected Result: The user can customize their notification settings to receive notifications based on their preferences.
- * Severity: Major
- * Priority: Medium
- * Type of Testing: Functional Testing
- * Test Case Approach: Positive

****Test Case 3: Verify that notifications provide detailed information about the activity****

- * Test Case ID: NTN-003
- * Test Case Title: Verify that notifications provide detailed information about the activity
- * Test Case Description: This test case aims to ensure that notifications provide detailed information about the activity, such as the task name, description, and the user who made the update.
- * Test Suite: Task Management Tool - Task Notifications
- * Test Priority: High
- * Preconditions:
 - * The user is logged in to the Task Management Tool
 - * A task is created and assigned to another user
 - * The task is updated by the assigned user
- * Test Data: No test data needed
- * Test Steps:
 1. Log in to the Task Management Tool as the user who was assigned the task
 2. Navigate to the task and update it by changing its status or description
 3. Verify that the notification provides detailed information about the activity (e.g., task name, description,

user who made the update)

- * Postconditions:

- * The notification should provide detailed information about the activity

- * The user should be able to access the task details to view the update

- * Expected Result: The notification provides detailed information about the activity, such as the task name, description, and the user who made the update.

- * Severity: Critical

- * Priority: High

- * Type of Testing: Functional Testing

- * Test Case Approach: Positive

****Test Case 4: Verify that the system logs notification history for future reference****

- * Test Case ID: NTN-004

- * Test Case Title: Verify that the system logs notification history for future reference

- * Test Case Description: This test case aims to ensure that the system logs notification history for future reference.

- * Test Suite: Task Management Tool - Task Notifications

- * Test Priority: Medium

- * Preconditions:

- * The user is logged in to the Task Management Tool

- * Multiple tasks are updated by different users

- * Test Data: No test data needed

- * Test Steps:

1. Log in to the Task Management Tool as the user

2. Navigate to the notification history page

3. Verify that the system logs notification history for future reference (e.g., task updates, comments, mentions)

- * Postconditions:

- * The notification history should be logged and visible

- * The user should be able to access the notification history for future reference

- * Expected Result: The system logs notification history for future reference.

- * Severity: Major

- * Priority: Medium

- * Type of Testing: Functional Testing

- * Test Case Approach: Positive

****Test Case 5: Verify that users can choose notification delivery methods (email, mobile, in-app)****

- * Test Case ID: NTN-005

- * Test Case Title: Verify that users can choose notification delivery methods (email, mobile, in-app)

- * Test Case Description: This test case aims to ensure that users can choose notification delivery methods (e.g., email, mobile, in-app).

- * Test Suite: Task Management Tool - Task Notifications

- * Test Priority: Medium

- * Preconditions:

- * The user is logged in to the Task Management Tool

- * Test Data: No test data needed

- * Test Steps:

1. Log in to the Task Management Tool as the user

2. Navigate to the notification settings page

3. Choose different notification delivery methods (e.g., email, mobile, in-app)

4. Verify that the user receives notifications via the chosen delivery methods

* Postconditions:

* The notification settings should be saved and applied

* The user should receive notifications via the chosen delivery methods

* Expected Result: Users can choose notification delivery methods (email, mobile, in-app) to receive notifications.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

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:

Here are the detailed test cases for the Task Archiving feature:

****Test Case 1: Verify that users can archive completed or inactive tasks****

*** **Test Case ID:** TC_001**

*** **Test Case Title:** Verify that users can archive completed or inactive tasks**

*** **Test Case Description:**** This test case verifies that users can successfully archive completed or inactive tasks from the active task list. The objective is to ensure that the system allows users to declutter their task list and keep their focus on active tasks.

*** **Test Suite:** Task Management Tool**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is signed in and has access to a project with active tasks

+ The user has completed or marked a task as inactive

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the task management tool as a user with permissions to archive tasks
2. Navigate to a project with active tasks
3. Mark a task as completed or inactive
4. Click the "Archive" button next to the completed/inactive task

5. Verify that the task is moved to the archived section

* **Postconditions:**

+ The task is archived and no longer appears in the active task list

+ The archived task is still accessible through the archived section

* **Expected Result:** The system successfully archives the completed or inactive task

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that archived tasks are moved to a separate section for easy access

* **Test Case ID:** TC_002

* **Test Case Title:** Verify that archived tasks are moved to a separate section for easy access

* **Test Case Description:** This test case verifies that archived tasks are properly moved to a separate section for easy access. The objective is to ensure that users can still access their archived tasks in a separate section, even after archiving them.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* **Preconditions:**

+ User is signed in and has access to a project with archived tasks

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a user with permissions to archive tasks

2. Navigate to a project with archived tasks

3. Verify that the archived tasks are listed in the separate archived section

4. Check that Archived tasks are sorted in the same order as the original active task list

* **Postconditions:**

+ Archived tasks are only accessible through the archived section

* **Expected Result:** The system moves archived tasks to a separate archived section, still accessible to users

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can search and view archived tasks

* **Test Case ID:** TC_003

* **Test Case Title:** Verify that users can search and view archived tasks

* **Test Case Description:** This test case verifies that users can successfully search and view archived tasks. The objective is to ensure that the system makes it easy for users to find and view their archived tasks.

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* **Preconditions:**

+ User is signed in and has access to a project with archived tasks

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a user with permissions to archive tasks
2. Navigate to a project with archived tasks
3. Use the search function to find a specific archived task
4. Verify that the search results include the archived task
5. Click on the archived task to view its details

* **Postconditions:**

+ The archived task is viewable with all its original details

* **Expected Result:** The system successfully displays archived tasks in search results

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that the system provides an option to restore archived tasks to the active list

* **Test Case ID:** TC_004

* **Test Case Title:** Verify that the system provides an option to restore archived tasks to the active list

* **Test Case Description:** This test case verifies that the system offers an option to restore archived tasks to their original active task list. The objective is to ensure that users have an easy way to undo the archiving of a task.

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* **Preconditions:**

+ User is signed in and has access to a project with archived tasks

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a user with permissions to archive tasks
2. Navigate to a project with archived tasks
3. Select an archived task to restore
4. Click the "Restore" button next to the archived task
5. Verify that the task is moved back to the active task list

* **Postconditions:**

+ The archived task is restored to its original position in the active task list

* **Expected Result:** The system provides an option to restore archived tasks to the active list

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that archived tasks retain all original details and history

* **Test Case ID:** TC_005

* **Test Case Title:** Verify that archived tasks retain all original details and history

* **Test Case Description:** This test case verifies that archived tasks maintain all their original details and history. The objective is to ensure that users can still rely on archived tasks for reference and reporting purposes.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* **Preconditions:**

+ User is signed in and has access to a project with archived tasks

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the task management tool as a user with permissions to archive tasks
2. Navigate to a project with archived tasks
3. Select an archived task to view its details
4. Verify that the archived task retains all its original details, including:
 - Task description
 - Attachments
 - Comments
 - Assignee

- Priority

- Status

5. Verify that the archived task retains its original history, including:

- Creation date and time

- Update history

- Activity logs

* **Postconditions:**

+ The archived task maintains all its original details and history

* **Expected Result:** The system retains all original details and history for archived tasks

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 40

Project name -

Task Management Tool

Project description -

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

Feature name -

Guest Access

Feature description -

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on

specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

Userstory title -

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

Acceptance criteria -

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

Generated Test Cases:

Here are five detailed test cases for the Guest Access feature, each thoroughly covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can invite external collaborators as guests with limited access****

*** **Test Case ID:** TEST-CASE-001**

*** **Test Case Title:** Verify that users can invite external collaborators as guests with limited access**

*** **Test Case Description:** This test case verifies that users can successfully invite external collaborators as guests with limited access to view and collaborate on specific tasks or projects.**

*** **Test Suite:** Guest Access Module**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user has a valid account on the Task Management Tool.

+ The user has specific tasks or projects to share with external collaborators.

*** **Test Data:** Guest email and contact information**

****Test Steps:****

1. Log in to the Task Management Tool as a user.
2. Navigate to the specific task or project page.
3. Click on the "Invite Guest" button.
4. Enter the guest's email and contact information.
5. Select the access level for the guest (view-only or collaborate).

****Postconditions:****

- + The guest receives a notification email with an invitation link to access the task or project.
- + The guest is added to the task or project with the selected access level.

****Expected Result:**** The guest is successfully invited with limited access to view and collaborate on the task or project.

****Severity:**** Minor

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that guests can view and collaborate on specific tasks or projects****

****Test Case ID:**** TEST-CASE-002

****Test Case Title:**** Verify that guests can view and collaborate on specific tasks or projects

****Test Case Description:**** This test case verifies that invited guests can successfully view and collaborate on specific tasks or projects.

****Test Suite:**** Guest Access Module

****Test Priority:**** Medium

****Preconditions:****

- + The user has previously invited the guest with view-only or collaborate access.

****Test Data:**** Guest email and access level

****Test Steps:****

1. The guest clicks on the invitation link to access the task or project.
2. The guest is directed to the task or project page.
3. The guest can view the task or project details.
4. If the guest has collaborate access, the guest can also edit or add comments to the task or project.

****Postconditions:****

+ The guest can view and interact with the task or project as intended.

****Expected Result:**** The guest can successfully view and collaborate on the task or project.

****Severity:**** Trivial

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 3: Verify that the system provides clear distinctions between guest and full access roles****

****Test Case ID:**** TEST-CASE-003

****Test Case Title:**** Verify that the system provides clear distinctions between guest and full access roles

****Test Case Description:**** This test case verifies that the system displays clear differences between guest and full access roles for tasks and projects.

****Test Suite:**** Guest Access Module

****Test Priority:**** High

****Preconditions:****

+ The user has a valid account on the Task Management Tool.

+ The user has tasks or projects to share with guests and themselves.

****Test Data:**** User role (admin/user) and access levels

****Test Steps:****

1. Log in to the Task Management Tool as an admin/user.

2. Navigate to the specific task or project page.
3. Observe the display of guest and admin/user roles.
4. Verify that guests have restricted access to areas.

* **Postconditions:**

+ The system displays clear distinctions between guest and full access roles.

* **Expected Result:** The system displays clear differences between guest and full access roles.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that users can manage and revoke guest access at any time****

* **Test Case ID:** TEST-CASE-004

* **Test Case Title:** Verify that users can manage and revoke guest access at any time

* **Test Case Description:** This test case verifies that users can manage and revoke guest access to tasks and projects at any time.

* **Test Suite:** Guest Access Module

* **Test Priority:** Medium

* **Preconditions:**

+ The user has previously invited guests with access to tasks or projects.

* **Test Data:** Guest email and access level

* **Test Steps:**

1. Log in to the Task Management Tool as a user.
2. Navigate to the specific task or project page.
3. Click on the "Manage Guests" button.
4. Select the guest to revoke access from.

5. Confirm the guest access revocation.

* **Postconditions:**

+ The guest is no longer able to access the task or project.

* **Expected Result:** The user can successfully manage and revoke guest access at any time.

* **Severity:** Trivial

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that guests cannot access sensitive or restricted areas of the platform****

* **Test Case ID:** TEST-CASE-005

* **Test Case Title:** Verify that guests cannot access sensitive or restricted areas of the platform

* **Test Case Description:** This test case verifies that guests cannot access sensitive or restricted areas of the Task Management Tool, such as admin controls or personal data.

* **Test Suite:** Guest Access Module

* **Test Priority:** High

* **Preconditions:**

+ The user has previously invited guests with limited access.

* **Test Data:** Guest email and access level

* **Test Steps:**

1. Log in to the Task Management Tool as a guest with limited access.

2. Attempt to access sensitive or restricted areas (e.g., admin controls, personal data).

3. Observe the system's response.

* **Postconditions:**

+ The guest is denied access to sensitive or restricted areas.

* **Expected Result:** The guest cannot access sensitive or restricted areas of the platform.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

Data - 41

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Import/Export

Feature description -

Users can import tasks from external sources or export tasks to other task management tools or file formats. Import/export capabilities facilitate data migration and integration with other systems, ensuring seamless collaboration and workflow management across platforms.

Userstory title -

As a user, I want to import and export tasks so that I can migrate data between different task management tools.

Acceptance criteria -

['Users can import tasks from external sources in various formats (e.g., CSV, Excel).', 'Users can export tasks to other task management tools or file formats.', 'The system provides clear mapping of fields during import/export.', 'Users can preview imported data before finalizing the process.', 'The system logs

import/export activities for future reference.']

Generated Test Cases:

Here are five detailed test cases for the Task Import/Export feature, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can import tasks from external sources in various formats****

* ****Test Case ID:**** TC001

* ****Test Case Title:**** Verify that users can import tasks from external sources in various formats

* ****Test Case Description:**** This test case verifies that users can import tasks from external sources in various formats, such as CSV and Excel.

* ****Test Suite:**** Task Import/Export

* ****Test Priority:**** High

* ****Preconditions:****

+ User has a valid account and is logged in

+ User has a valid task in CSV/Excel format

* ****Test Data:**** CSV file containing task data

* ****Test Steps:****

1. Log in to the system
2. Navigate to the import tasks page
3. Select the CSV file containing task data
4. Review the task fields and adjust mapping as needed
5. Click the "Import" button

* ****Postconditions:****

+ Imported tasks are created in the system

+ Task fields are mapped correctly

* **Expected Result:** Tasks are successfully imported from external sources in CSV/Excel format

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that users can export tasks to other task management tools or file formats

* **Test Case ID:** TC002

* **Test Case Title:** Verify that users can export tasks to other task management tools or file formats

* **Test Case Description:** This test case verifies that users can export tasks to other task management tools or file formats, such as CSV, Excel, and PDF.

* **Test Suite:** Task Import/Export

* **Test Priority:** High

* **Preconditions:**

+ User has a valid account and is logged in

+ User has tasks to export

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the system
2. Navigate to the export tasks page
3. Select the file format to export to (e.g., CSV, Excel, PDF)
4. Review the task fields and adjust mapping as needed
5. Click the "Export" button

* **Postconditions:**

+ Tasks are exported to the selected file format

+ Task fields are mapped correctly

* **Expected Result:** Tasks are successfully exported to other task management tools or file formats

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that the system provides clear mapping of fields during import/export

* **Test Case ID:** TC003

* **Test Case Title:** Verify that the system provides clear mapping of fields during import/export

* **Test Case Description:** This test case verifies that the system provides clear mapping of fields during import/export, ensuring that task fields are correctly mapped and configured.

* **Test Suite:** Task Import/Export

* **Test Priority:** High

* **Preconditions:**

+ User has a valid account and is logged in

+ User has tasks to import/export

* **Test Data:** CSV file containing task data

* **Test Steps:**

1. Log in to the system
2. Navigate to the import/export tasks page
3. Review the field mapping for import/export
4. Adjust field mapping as needed
5. Perform import/export operation

* **Postconditions:**

+ Field mapping is clear and accurate

+ Task fields are correctly mapped and configured

* **Expected Result:** System provides clear mapping of fields during import/export

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that users can preview imported data before finalizing the process

* **Test Case ID:** TC004

* **Test Case Title:** Verify that users can preview imported data before finalizing the process

* **Test Case Description:** This test case verifies that users can preview imported data before finalizing the process, ensuring that data is correctly imported and formatted.

* **Test Suite:** Task Import/Export

* **Test Priority:** Medium

* **Preconditions:**

+ User has a valid account and is logged in

+ User has a CSV file containing task data

* **Test Data:** CSV file containing task data

* **Test Steps:**

1. Log in to the system

2. Navigate to the import tasks page

3. Select the CSV file containing task data

4. Preview imported data

5. Verify data formatting and accuracy

* **Postconditions:**

+ User has reviewed and confirmed imported data

+ Data is correctly formatted and accurate

* **Expected Result:** Users can preview imported data before finalizing the process

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that the system logs import/export activities for future reference

* **Test Case ID:** TC005

* **Test Case Title:** Verify that the system logs import/export activities for future reference

* **Test Case Description:** This test case verifies that the system logs import/export activities, providing a history of all import/export operations for future reference.

* **Test Suite:** Task Import/Export

* **Test Priority:** Low

* **Preconditions:**

+ User has a valid account and is logged in

+ User has performed import/export operations

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the system

2. Navigate to the import/export history page

3. Review import/export logs

4. Verify log accuracy and completeness

* **Postconditions:**

+ System logs import/export activities accurately

+ Logs are complete and up-to-date

* **Expected Result:** System logs import/export activities for future reference

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases cover all five acceptance criteria and ensure that the Task Import/Export feature is thoroughly tested, providing a high level of confidence in the feature's quality and correctness.

Data - 42

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.

Userstory title -

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

Acceptance criteria -

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The

system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.', 'Priority changes are logged and tracked.']

Generated Test Cases:

Here are five detailed test cases for the Task Prioritization feature of the Task Management Tool:

****Test Case 1: Set and Adjust Task Priorities****

****Test Case ID:**** TP-001

****Test Case Title:**** Verify that users can set and adjust task priorities

****Test Case Description:**** This test case covers the acceptance criterion "Users can set and adjust task priorities." We will verify that users can successfully set and adjust task priorities for a task.

****Test Suite:**** Task Prioritization Module

****Test Priority:**** High

****Preconditions:**** User has created a new task, logged in to the system, and navigated to the task details page.

****Test Data:**** A valid task with no assigned priority.

****Test Steps:****

1. User selects a valid priority level (e.g., High, Medium, Low) from the dropdown menu.
2. User saves the task with the selected priority level.
3. User navigates back to the task details page and verifies that the priority level is displayed correctly.
4. User selects a different valid priority level from the dropdown menu and saves the task again.
5. User verifies that the priority level has been updated correctly.

****Postconditions:**** The task is saved with the updated priority level.

****Expected Result:**** Users can successfully set and adjust task priorities.

****Severity:**** Critical

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 2: Sorting and Filtering Tasks****

****Test Case ID:** TP-002**

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

****Test Case Description:** This test case covers the acceptance criterion "Tasks can be sorted and filtered based on priority levels." We will verify that tasks can be successfully sorted and filtered by priority levels.**

****Test Suite:** Task Prioritization Module**

****Test Priority:** Medium**

****Preconditions:** User has created multiple tasks, each with different priority levels, and logged in to the system.**

****Test Data:** Multiple tasks with different priority levels.**

****Test Steps:****

1. User navigates to the task list page and verifies that tasks are listed in chronological order (by default).
2. User selects the "Sort by priority" option from the dropdown menu.
3. User verifies that tasks are sorted in descending order of priority (highest priority first).
4. User selects the "Filter by priority" option and chooses a specific priority level (e.g., High).
5. User verifies that tasks with the selected priority level are displayed only.

****Postconditions:** Tasks are sorted and filtered correctly by priority levels.**

****Expected Result:** Tasks can be successfully sorted and filtered by priority levels.**

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 3: Visual Indicators of Task Priority****

****Test Case ID:** TP-003**

****Test Case Title:** Verify that the system provides visual indicators of task priority**

****Test Case Description:** This test case covers the acceptance criterion "The system provides visual indicators of task priority." We will verify that the system displays visual indicators for each task based on its priority level.**

****Test Suite:** Task Prioritization Module**

****Test Priority:** Low**

****Preconditions:** User has created multiple tasks with different priority levels, and logged in to the system.**

****Test Data:** Multiple tasks with different priority levels.**

****Test Steps:****

1. User navigates to the task list page and verifies that each task has a visual indicator (e.g., color-coded label or icon) based on its priority level.
2. User selects a task with a valid priority level (e.g., High) and verifies that the visual indicator is displayed correctly.
3. User verifies that the visual indicator changes when the task priority is updated.

****Postconditions:** The system displays visual indicators for task priority correctly.**

****Expected Result:** The system provides visual indicators for task priority.**

****Severity:** Minor**

****Priority:** Low**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 4: Notifications for High-Priority Tasks****

****Test Case ID:** TP-004**

****Test Case Title:** Verify that users receive notifications for high-priority tasks**

****Test Case Description:** This test case covers the acceptance criterion "Users receive notifications for high-priority tasks." We will verify that users receive email notifications for high-priority tasks.**

****Test Suite:** Task Prioritization Module**

****Test Priority:** High**

****Preconditions:** User has created a task with a high priority level, logged in to the system, and configured email notifications.**

****Test Data:** A task with high priority level.**

****Test Steps:****

1. User navigates to the task details page and verifies that the task has a high priority level.
2. User confirms that they are not receiving email notifications for the task (initially).
3. Task's due date approaches, and the system sends an email notification.
4. User verifies that they received the email notification with task details.

****Postconditions:** User received email notifications for high-priority tasks correctly.**

****Expected Result:** Users receive notifications for high-priority tasks.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 5: Tracking Priority Changes****

****Test Case ID:** TP-005**

****Test Case Title:**** Verify that priority changes are logged and tracked

****Test Case Description:**** This test case covers the acceptance criterion "Priority changes are logged and tracked." We will verify that priority changes are recorded in the system's activity logs.

****Test Suite:**** Task Prioritization Module

****Test Priority:**** Medium

****Preconditions:**** User has created a task, logged in to the system, and navigated to the task details page.

****Test Data:**** A task with an initial priority level.

****Test Steps:****

1. User updates the task's priority level.
2. User navigates to the task history page and verifies that the priority change is recorded in the activity log.
3. User verifies that the activity log contains the correct task details and the updated priority level.

****Postconditions:**** Priority changes are recorded in the activity log correctly.

****Expected Result:**** Priority changes are logged and tracked.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

Data - 43

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

Userstory title -

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay updated on task progress.

Acceptance criteria -

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

Generated Test Cases:

Here are five detailed test cases for the Task Collaboration feature of the Task Management Tool:

****Test Case 1: Verify that users can assign tasks to team members****

****Test Case ID:** TCTC001**

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

****Test Case Description:**** This test case will verify that users can successfully assign tasks to team members from the task details page. We will test the functionality of assigning a task to a user, removing a team member from a task, and reassigning a task to another user.

****Test Suite:** Task Collaboration**

****Test Priority:** High**

****Preconditions:****

1. User has created a task and is assigned to the task.
2. Team members have been added to the task.

****Test Data:**** No test data needed

****Test Steps:****

1. Log in as a user with task creation permission.
2. Create a new task and assign it to a team member.
3. Open the task details page and select the assignee.
4. Remove the current assignee and assign the task to another team member.
5. Verify that the task is successfully assigned to the new team member.
6. Remove the team member and reassign the task to the original user.

****Postconditions:**** None

****Expected Result:**** Users can successfully assign tasks to team members from the task details page.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that users can share updates and communicate through comments or messages****

****Test Case ID:**** TCTC002

****Test Case Title:**** Verify that users can share updates and communicate through comments or messages

****Test Case Description:**** This test case will verify that users can successfully share updates and communicate with team members through comments or messages. We will test the functionality of adding comments and messages, sending notifications to team members, and viewing updates in the task details page.

****Test Suite:**** Task Collaboration

****Test Priority:**** High

****Preconditions:****

1. User has created a task and is assigned to the task.
2. Team members have been added to the task.

****Test Data:**** No test data needed

****Test Steps:****

1. Log in as a user with task creation permission.
2. Create a new task and add team members.
3. Add a comment to the task and verify that the team members receive notifications.
4. Send a message to a team member and verify that the message is visible in the task details page.
5. Verify that all updates and comments are visible to assigned team members.

****Postconditions:**** None

****Expected Result:**** Users can successfully share updates and communicate with team members through comments or messages.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 3: Verify that task updates and comments are visible to assigned team members****

****Test Case ID:**** TCTC003

****Test Case Title:**** Verify that task updates and comments are visible to assigned team members

****Test Case Description:**** This test case will verify that task updates and comments are visible to assigned team members and that team members can view task updates and comments in the task details page.

****Test Suite:**** Task Collaboration

****Test Priority:**** High

****Preconditions:****

1. User has created a task and is assigned to the task.
2. Team members have been added to the task.
3. Task updates and comments have been added.

****Test Data:**** No test data needed

****Test Steps:****

1. Log in as an assigned team member.
2. Open the task details page and verify that task updates and comments are visible.
3. Verify that team members can view task updates and comments in the task details page.
4. Verify that users can view task updates and comments in the task history.

****Postconditions:**** None

****Expected Result:**** Task updates and comments are visible to assigned team members.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 4: Verify that users receive notifications for task updates and comments****

****Test Case ID:**** TCTC004

****Test Case Title:**** Verify that users receive notifications for task updates and comments

****Test Case Description:**** This test case will verify that users receive notifications for task updates and comments. We will test the functionality of sending notifications when team members add comments or update task status.

****Test Suite:**** Task Collaboration

****Test Priority:**** High

****Preconditions:****

1. User has created a task and is assigned to the task.

2. Team members have been added to the task.
3. Task updates and comments have been added.

****Test Data:**** No test data needed

****Test Steps:****

1. Log in as a team member.
2. Add a comment to the task and verify that assigned team members receive notifications.
3. Update task status and verify that assigned team members receive notifications.
4. Verify that notifications are sent to team members through email or in-task notifications.

****Postconditions:**** None

****Expected Result:**** Users receive notifications for task updates and comments.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 5: Verify that the system provides a clear view of task collaboration history****

****Test Case ID:**** TCTC005

****Test Case Title:**** Verify that the system provides a clear view of task collaboration history

****Test Case Description:**** This test case will verify that the system provides a clear view of task collaboration history, including task updates, comments, and assignments. We will test the functionality of viewing task history and collaborative activity on the task details page.

****Test Suite:**** Task Collaboration

****Test Priority:**** High

****Preconditions:****

1. User has created a task and is assigned to the task.
2. Team members have been added to the task.

3. Task updates and comments have been added.

****Test Data:**** No test data needed

****Test Steps:****

1. Log in as a user with task creation permission.
2. Open the task details page and select the task history tab.
3. Verify that task history displays clear and concise information about task updates, comments, and assignments.
4. Verify that users can view task collaboration history in the task details page.

****Postconditions:**** None

****Expected Result:**** The system provides a clear view of task collaboration history.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional 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:

Here are five detailed test cases for the Task Management Tool's Task Tracking feature, each designed to cover one or more of the acceptance criteria from the user story:

****Test Case 1: Verify that users can view task statuses****

Test Case ID: TCTT-001

Test Case Title: Verify that users can view task statuses

Test Case Description: This test case aims to ensure that users can view the status of their tasks, including any updates made by team members or project managers.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- * User account is created with necessary permissions
- * A task is created with a status update made by another user

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool with a valid user account.
2. Navigate to the "Tasks" module and select the task created in the preconditions.
3. Verify that the task status is displayed correctly.
4. Update the task status to a new value and save the changes.
5. Refresh the page and verify that the updated status is displayed.

Postconditions:

- * Task status is displayed correctly
- * Task status is updated correctly

Expected Result: The task status is displayed correctly, and updating the status results in a refreshed display of the new status.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that users can track task progress and deadlines****

Test Case ID: TCTT-002

Test Case Title: Verify that users can track task progress and deadlines

Test Case Description: This test case aims to ensure that users can track the progress of tasks, including their deadlines and any updates made by team members or project managers.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- * User account is created with necessary permissions
- * A task is created with a deadline and progress update made by another user

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool with a valid user account.
2. Navigate to the "Tasks" module and select the task created in the preconditions.
3. Verify that the task's deadline and progress are displayed correctly.

4. Update the task's deadline or progress and save the changes.
5. Refresh the page and verify that the updated deadline or progress is displayed.

Postconditions:

- * Task deadline and progress are displayed correctly
- * Task deadline and progress are updated correctly

Expected Result: The task's deadline and progress are displayed correctly, and updating the deadline or progress results in a refreshed display of the new values.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that tasks are updated in real-time****

Test Case ID: TCTT-003

Test Case Title: Verify that tasks are updated in real-time

Test Case Description: This test case aims to ensure that tasks are updated in real-time when changes are made by team members or project managers.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- * User account is created with necessary permissions
- * A task is created with a status update made by another user

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool with a valid user account.
2. Navigate to the "Tasks" module and select the task created in the preconditions.
3. Observe the task's status while another user makes a status update.
4. Verify that the task's status is updated in real-time.

Postconditions:

- * Task status is updated in real-time

Expected Result: The task's status is updated in real-time when another user makes a status update.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that the system provides visual indicators for task progress****

Test Case ID: TCTT-004

Test Case Title: Verify that the system provides visual indicators for task progress

Test Case Description: This test case aims to ensure that the Task Management Tool provides visual indicators for task progress, such as progress bars or color-coded status levels.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- * User account is created with necessary permissions
- * A task is created with a progress update made by another user

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool with a valid user account.
2. Navigate to the "Tasks" module and select the task created in the preconditions.
3. Verify that the task's progress is displayed using a visual indicator (e.g., progress bar or color-coded status level).
4. Update the task's progress and verify that the visual indicator changes accordingly.

Postconditions:

- * Visual indicator for task progress is displayed correctly
- * Visual indicator for task progress is updated correctly

Expected Result: The task's progress is displayed using a visual indicator, and updating the progress results in a refreshed display of the new progress.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that users are notified when task deadlines approach****

Test Case ID: TCTT-005

Test Case Title: Verify that users are notified when task deadlines approach

Test Case Description: This test case aims to ensure that users are notified when task deadlines approach, using notification mechanisms such as email or in-app notifications.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- * User account is created with necessary permissions
- * A task is created with a deadline approaching within the next 24 hours

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool with a valid user account.
2. Set the task created in the preconditions to have a deadline approaching within the next 24 hours.
3. Observe the notification mechanism (e.g., email or in-app notification) to see if a notification is sent to the user.
4. Verify that the notification is sent correctly.

Postconditions:

- * User is notified when task deadline approaches

Expected Result: The user is notified when the task deadline approaches, using the notification mechanism configured in the Task Management Tool.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 45

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

Userstory title -

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

Acceptance criteria -

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.', 'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', '']

Generated Test Cases:

Here are five detailed test cases that cover each of the acceptance criteria from the user story:

****Test Case 1: Verify that Users Receive Notifications for New Tasks****

*** **Test Case ID:** T_TNT_001**

*** **Test Case Title:** Verify that Users Receive Notifications for New Tasks**

*** **Test Case Description:**** This test case aims to verify that users receive notifications when a new task is created, with the assumed scenario that the user has chosen to receive notifications for new tasks.

*** **Test Suite:** Task Notifications**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user has set up their notification settings to receive emails for new tasks.

+ A new task has been created by another user or administrator.

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the application as a user who has enabled notifications for new tasks.
2. Verify that the user receives an email notification for the new task.
3. Open the email notification and verify that it contains the task details.

*** **Postconditions:**** The user has received the email notification for the new task.

*** **Expected Result:**** The user receives the notification email with the task details.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that Users Receive Notifications for Task Updates and Comments****

* **Test Case ID:** T_TNT_002

* **Test Case Title:** Verify that Users Receive Notifications for Task Updates and Comments

* **Test Case Description:** This test case aims to verify that users receive notifications when a task is updated or commented on, with the assumed scenario that the user has chosen to receive notifications for task updates and comments.

* **Test Suite:** Task Notifications

* **Test Priority:** High

* **Preconditions:**

+ The user has set up their notification settings to receive emails for task updates and comments.

+ A task has been updated or commented on by another user or administrator.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the application as a user who has enabled notifications for task updates and comments.
2. Create or edit a task and make some changes.
3. Verify that the user receives an email notification for the task update.
4. Leave a comment on the task.
5. Verify that the user receives an email notification for the comment.

* **Postconditions:** The user has received the email notifications for the task update and comment.

* **Expected Result:** The user receives the notification email for the task update and comment.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that Users Receive Reminders for Upcoming Deadlines

* **Test Case ID:** T_TNT_003

* **Test Case Title:** Verify that Users Receive Reminders for Upcoming Deadlines

* **Test Case Description:** This test case aims to verify that users receive reminders when a task deadline is approaching, with the assumed scenario that the user has chosen to receive reminders for upcoming deadlines.

* **Test Suite:** Task Notifications

* **Test Priority:** High

* **Preconditions:**

+ The user has set up their notification settings to receive reminders for upcoming deadlines.

+ A task has a deadline approaching.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the application as a user who has enabled reminders for upcoming deadlines.
2. Create a task with an approaching deadline.
3. Verify that the user receives an email notification for the reminder.

* **Postconditions:** The user has received the email reminder for the approaching deadline.

* **Expected Result:** The user receives the notification email for the reminder.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that Notification Settings Can Be Customized by Users****

*** **Test Case ID:** T_TNT_004**

*** **Test Case Title:** Verify that Notification Settings Can Be Customized by Users**

*** **Test Case Description:** This test case aims to verify that users can customize their notification settings, including which types of notifications they want to receive.**

*** **Test Suite:** Task Notifications**

*** **Test Priority:** High**

*** **Preconditions:** None**

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the application as a user.
2. Navigate to the notification settings.
3. Verify that the user can enable or disable specific types of notifications (e.g., new tasks, task updates, comments, reminders).
4. Verify that the user can customize the frequency of reminders (e.g., daily, weekly).

*** **Postconditions:** The user has successfully customized their notification settings.**

*** **Expected Result:** The user can successfully customize their notification settings.**

*** **Severity:** Major**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

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

****Test Case 5: Verify that Users Receive No Notifications When Notifications Are Disabled****

*** **Test Case ID:** T_TNT_005**

*** **Test Case Title:** Verify that Users Receive No Notifications When Notifications Are Disabled**

* **Test Case Description:** This test case aims to verify that users do not receive notifications when their notification settings are disabled.

* **Test Suite:** Task Notifications

* **Test Priority:** High

* **Preconditions:** The user has disabled notifications for new tasks, updates, comments, and reminders.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the application as a user who has disabled notifications.
2. Create a new task or update an existing task.
3. Verify that the user receives no email notifications.

* **Postconditions:** The user has received no email notifications.

* **Expected Result:** The user receives no notification emails.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

Data - 46

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Categorization

Feature description -

Users can categorize tasks into different groups or projects for better organization and management. Task categorization aims to help users organize tasks by type, priority, or project.

Userstory title -

As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

Acceptance criteria -

['Users can create and assign tasks to different categories or projects.', 'Users can view tasks grouped by categories or projects.', 'Tasks can be moved between categories or projects.', 'Category or project information is accurately reflected in task views.', "']

Generated Test Cases:

Here are five detailed test cases for the Task Categorization feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that Users Can Create and Assign Tasks to Categories****

Test Case ID	TC001
-----	-----
Test Case Title	Verify that Users Can Create and Assign Tasks to Categories
Test Case Description	As a user, I want to create and assign tasks to different categories or projects so that I can organize and manage my tasks effectively.
Test Suite	Task Management Tool
Test Priority	High
Preconditions	User has logged in to the system, User has created at least one task category or project
Test Data	No test data needed

| Test Steps | 1. Log in to the system as a user. 2. Navigate to the Task Categories/Projects page. 3. Click on the "Create New Category/Project" button. 4. Enter category/project name and description. 5. Create a new task and assign it to the newly created category/project. |

| Postconditions | Category/project is created and task is assigned successfully. |

| Expected Result | Category/project is created and task is assigned to the designated category/project. |

| Severity | Major |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 2: Verify that Users Can View Tasks Grouped by Categories****

| Test Case ID | TC002 |

|-----|-----|

| Test Case Title | Verify that Users Can View Tasks Grouped by Categories |

| Test Case Description | As a user, I want to view tasks grouped by categories or projects so that I can see tasks organized by type or priority. |

| Test Suite | Task Management Tool |

| Test Priority | Medium |

| Preconditions | User has logged in to the system, User has created at least one task category or project with tasks assigned. |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the system as a user. 2. Navigate to the Task Categories/Projects page. 3. Click on a category/project. 4. Verify that tasks are displayed grouped by category/project. |

| Postconditions | Tasks are displayed correctly grouped by category/project. |

| Expected Result | Tasks are displayed correctly grouped by category/project. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 3: Verify that Tasks Can Be Moved Between Categories****

| Test Case ID | TC003 |

|-----|-----|

| Test Case Title | Verify that Tasks Can Be Moved Between Categories |

| Test Case Description | As a user, I want to move tasks between categories or projects so that I can reorganize my tasks. |

| Test Suite | Task Management Tool |

| Test Priority | Medium |

| Preconditions | User has logged in to the system, User has created at least two task categories or projects with tasks assigned. |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the system as a user. 2. Navigate to the Task Categories/Projects page. 3. Select a task from one category/project. 4. Click on the "Move to Another Category/Project" button. 5. Select the target category/project. 6. Verify that the task is moved successfully. |

| Postconditions | Task is moved successfully between categories/projects. |

| Expected Result | Task is moved to the designated category/project. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 4: Verify that Category or Project Information is Accurately Reflected in Task Views****

| Test Case ID | TC004 |

|-----|-----|

| Test Case Title | Verify that Category or Project Information is Accurately Reflected in Task Views |

| Test Case Description | As a user, I want to see accurate category/project information in task views so that I can make informed decisions. |

| Test Suite | Task Management Tool |

| Test Priority | High |

| Preconditions | User has logged in to the system, User has created at least one task category or project with tasks assigned. |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the system as a user. 2. Navigate to the Task Categories/Projects page. 3. Click on a category/project. 4. Verify that task view displays accurate category/project information. |

| Postconditions | Category/project information is accurately reflected in task views. |

| Expected Result | Category/project information is accurately displayed in task views. |

| Severity | Major |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 5: Verify that Users Cannot Move Tasks That Do Not Belong to a Category/Project****

| Test Case ID | TC005 |

|-----|-----|

| Test Case Title | Verify that Users Cannot Move Tasks That Do Not Belong to a Category/Project |

| Test Case Description | As a user, I want to ensure that I cannot move tasks that do not belong to a category/project so that I avoid inconsistencies. |

| Test Suite | Task Management Tool |

| Test Priority | Low |

| Preconditions | User has logged in to the system, Task is not assigned to a category/project. |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the system as a user. 2. Navigate to the Task Categories/Projects page. 3. Select a task that is not assigned to a category/project. 4. Attempt to move the task to another category/project. 5.

Verify that an error message is displayed. |

| Postconditions | Error message is displayed when trying to move unassigned task. |

| Expected Result | Error message is displayed when trying to move an unassigned task. |

| Severity | Minor |

| Priority | Low |

| Type of Testing | Functional Testing |

| Test Case Approach | Negative |

I hope this helps! Each test case covers one or more of the acceptance criteria, and they are all positive and functional testing cases, making sure to ensure product quality and identify potential defects early in the development cycle.

Data - 47

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Search

Feature description -

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

Userstory title -

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

Generated Test Cases:

Here are 5 detailed test cases that cover all the acceptance criteria of the Task Search feature:

****Test Case 1: Verify that users can search for tasks by name****

*** **Test Case ID:** TCT-001**

*** **Test Case Title:** Verify that users can search for tasks by name**

*** **Test Case Description:** This test case aims to validate that users can search for tasks by their name and that the search results are accurate and relevant.**

*** **Test Suite:** Task Search Suite**

*** **Test Priority:** High**

*** **Preconditions:****

+ The user is logged in to the task management tool

+ The task list is populated with at least two tasks

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the task management tool
2. Go to the task list page
3. Enter a task name in the search box (e.g., "Meeting with John")
4. Click the search button
5. Verify that the search results include the task with the matching name

*** **Postconditions:**** The search results page is displayed, and the task with the matching name is highlighted.

*** **Expected Result:**** The search results include the task with the matching name, and the search results are accurate and relevant.

*** **Severity:**** Minor

*** **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 five detailed test cases, each covering one or more acceptance criteria from the user story:

****Test Case 1: Verify that users can add comments to tasks****

Test Case ID: TC-01

Test Case Title: Verify that users can add comments to tasks

Test Case Description: This test case will ensure that users can create and submit comments on tasks, and that comments are properly displayed and associated with the correct task.

Test Suite: Task Collaboration Module

Test Priority: Medium

Preconditions:

- * User is logged in with a valid account
- * User is on a task detail page
- * Task has no existing comments

Test Data:

- * Comment text (e.g. "This is a comment on the task")
- * Comment author (user who created the comment)

Test Steps:

1. Go to task detail page and click on "Add Comment" button
2. Enter comment text in the comment field
3. Select comment author from the dropdown list
4. Click on "Submit Comment" button
5. Verify that comment is displayed on task detail page
6. Verify that comment is properly associated with the correct task

Postconditions:

- * Comment is displayed on task detail page
- * Comment is associated with the correct task

Expected Result: User can add a comment to a task, and comment is properly displayed and associated with the task.

Severity: Critical

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that users can attach files to tasks****

Test Case ID: TC-02

Test Case Title: Verify that users can attach files to tasks

Test Case Description: This test case will ensure that users can upload and attach files to tasks, and that files are properly displayed and associated with the correct task.

Test Suite: Task Collaboration Module

Test Priority: Medium

Preconditions:

- * User is logged in with a valid account
- * User is on a task detail page
- * Task has no existing attachments

Test Data:

- * File to be attached (e.g. Excel file, Word document)
- * File name and description

Test Steps:

1. Go to task detail page and click on "Add Attachment" button
2. Select file to be attached from local file system
3. Enter file name and description
4. Click on "Attach File" button
5. Verify that file is displayed on task detail page
6. Verify that file is properly associated with the correct task

Postconditions:

- * File is displayed on task detail page
- * File is associated with the correct task

Expected Result: User can attach a file to a task, and file is properly displayed and associated with the task.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that users can share tasks with other users****

Test Case ID: TC-03

Test Case Title: Verify that users can share tasks with other users

Test Case Description: This test case will ensure that users can share tasks with other users, and that recipients receive notifications and task assignments.

Test Suite: Task Collaboration Module

Test Priority: High

Preconditions:

- * User is logged in with a valid account
- * User is on a task detail page
- * Task has no existing assigns or shared users

Test Data:

- * Assigned user (user who will receive the assignment)
- * Shared user (user who will receive the notification)

Test Steps:

1. Go to task detail page and click on "Share Task" button
2. Enter assigned user email or username
3. Select shared user(s) from the dropdown list
4. Enter notification message (optional)
5. Click on "Share Task" button
6. Verify that assigned user receives task assignment notification

7. Verify that shared user receives task-associated notification

Postconditions:

- * Assigned user receives task assignment notification

- * Shared user receives task-associated notification

Expected Result: User can share a task with other users, and recipients receive notifications and task assignments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that task collaboration features work seamlessly****

Test Case ID: TC-04

Test Case Title: Verify that task collaboration features work seamlessly

Test Case Description: This test case will ensure that task collaboration features (comments, attachments, sharing) work smoothly and do not cause any issues.

Test Suite: Task Collaboration Module

Test Priority: Medium

Preconditions:

- * User is logged in with a valid account
- * User is on a task detail page
- * Task has existing comments, attachments, and assigns

Test Data:

- * No test data needed

Test Steps:

1. Review task detail page to ensure all collaboration features are working properly
2. Add a new comment to the task
3. Attach a new file to the task
4. Share the task with an existing assigned user
5. Review task updates to ensure changes are reflected in real-time

Postconditions:

- * Task collaboration features work seamlessly

Expected Result: Task collaboration features (comments, attachments, sharing) work smoothly and do not cause any issues.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that task collaboration features enhance team communication****

Test Case ID: TC-05

Test Case Title: Verify that task collaboration features enhance team communication

Test Case Description: This test case will ensure that task collaboration features promote seamless team communication and collaboration.

Test Suite: Task Collaboration Module

Test Priority: High

Preconditions:

- * User is logged in with a valid account
- * User is a member of a team working on the task
- * Task has existing comments, attachments, and assigns

Test Data:

- * No test data needed

Test Steps:

1. Join a team working on the task
2. Observe team communication and collaboration on task
3. Participate in team discussion through comments and attachments
4. Verify that all team members can view and contribute to the task

Postconditions:

* Task collaboration features enhance team communication

Expected Result: Task collaboration features promote seamless team communication and collaboration.

Severity: Critical

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

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 the provided Test Case Creation Template:

****Test Case 1: Verify that Instructors can create courses using videos, presentations, quizzes, and assignments****

Test Case ID: CC-001

Test Case Title: Verify that Instructors can create courses using videos, presentations, quizzes, and assignments

Test Case Description: This test case verifies that instructors can create comprehensive courses using various multimedia elements like videos, presentations, quizzes, and assignments.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- * Instructor has a valid account.
- * Instructor has access to the course creation tools.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Click on the "Create Course" button.
3. Select the course title, description, and tags.
4. Upload a video file and presentation document.
5. Create a new quiz with questions and options.
6. Add a new assignment with instructions and due date.
7. Save the course as a draft.

Postconditions:

- * The course is saved as a draft.
- * The instructor can view the course details page.

Expected Result: The instructor can create a comprehensive course using various multimedia elements.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that The tools are user-friendly and require minimal technical skills****

Test Case ID: CC-002

Test Case Title: Verify that The tools are user-friendly and require minimal technical skills

Test Case Description: This test case verifies that the course creation tools are intuitive and require minimal

technical skills for instructors to use.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- * Instructor has a valid account.
- * Instructor has access to the course creation tools.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Click on the "Create Course" button.
3. Observe the user interface and workflow for creating a course.
4. Take notes on any usability issues or confusion.
5. Complete the course creation process.

Postconditions:

- * The course is created successfully.
- * The instructor can view the course details page.

Expected Result: The course creation tools are user-friendly and require minimal technical skills.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that Courses can be previewed before publishing****

Test Case ID: CC-003

Test Case Title: Verify that Courses can be previewed before publishing

Test Case Description: This test case verifies that instructors can preview their courses before publishing them.

Test Suite: Course Creation Tools

Test Priority: Medium

Preconditions:

- * Instructor has a valid account.
- * Instructor has access to the course creation tools.
- * Course is saved as a draft.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the Draft Courses page.
3. Select a draft course to preview.
4. Observe the course content and functionality.
5. Identify any issues or bugs.
6. Publish the course.
7. Verify the course is live and accessible.

Postconditions:

- * The course is published successfully.
- * The instructor can view the course details page.

Expected Result: Instructors can preview their courses before publishing.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that Instructors can save courses as drafts****

Test Case ID: CC-004

Test Case Title: Verify that Instructors can save courses as drafts

Test Case Description: This test case verifies that instructors can save their courses as drafts, allowing for future editing and publishing.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- * Instructor has a valid account.
- * Instructor has access to the course creation tools.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Click on the "Create Course" button.
3. Complete the course creation process up to the point where the instructor wants to save as a draft.
4. Click on the "Save as Draft" button.
5. Verify that the course is saved as a draft.
6. Visit the Draft Courses page to confirm the saved course.

Postconditions:

- * The course is saved as a draft.
- * The instructor can view the draft course details page.

Expected Result: Instructors can save their courses as drafts.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that Courses can be edited post-publishing****

Test Case ID: CC-005

Test Case Title: Verify that Courses can be edited post-publishing

Test Case Description: This test case verifies that instructors can edit their courses after publishing.

Test Suite: Course Creation Tools

Test Priority: Medium

Preconditions:

- * Instructor has a valid account.
- * Instructor has access to the course creation tools.
- * Course is published.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the Published Courses page.
3. Select a published course to edit.
4. Observe the course content and functionality.
5. Identify any issues or bugs.
6. Edit the course content and functionality.
7. Save the changes.
8. Verify the updated course is live and accessible.

Postconditions:

- * The course is updated successfully.
- * The instructor can view the updated course details page.

Expected Result: Instructors can edit their courses after publishing.

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:

Here are five detailed test cases for the "Interactive Lessons" feature:

****Test Case 1: Verify that lessons include interactive components****

* ****Test Case ID:**** LL-01

* ****Test Case Title:**** Verify that lessons include interactive components

* ****Test Case Description:**** This test case aims to verify that lessons on the online learning platform include interactive components such as quizzes, polls, simulations, and discussions.

* ****Test Suite:**** Interactive Lessons

* ****Test Priority:**** High

* ****Preconditions:****

* User is logged in to the online learning platform

* A lesson with interactive components is available

* ****Test Data:**** No test data needed

* ****Test Steps:****

1. Log in to the online learning platform as a registered user.
2. Access a lesson that includes interactive components.
3. Verify the presence of interactive components (quizzes, polls, simulations, and discussions).
4. Navigate through the interactive components and verify their functionality.

* ****Postconditions:****

* The system displays the lesson with interactive components.

* Interactive components are functional and accessible.

* ****Expected Result:**** The lesson includes interactive components, and they are functional and accessible.

* ****Severity:**** Blocker

* ****Priority:**** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that interactive elements are integrated seamlessly into the lesson content

* **Test Case ID:** LL-02

* **Test Case Title:** Verify that interactive elements are integrated seamlessly into the lesson content

* **Test Case Description:** This test case aims to verify that interactive elements such as quizzes, polls, simulations, and discussions are seamlessly integrated into the lesson content.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:**

- * User is logged in to the online learning platform

- * A lesson with integrated interactive elements is available

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform as a registered user.

2. Access a lesson that includes interactive elements seamlessly integrated into the content.

3. Verify the integration of interactive elements into the lesson content.

4. Navigate through the lesson content and verify that interactive elements are contextually relevant.

* **Postconditions:**

- * The system displays the lesson with seamlessly integrated interactive elements.

- * Interactive elements are contextually relevant and enhance the lesson content.

* **Expected Result:** Interactive elements are seamlessly integrated into the lesson content and are contextually relevant.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that learners can track their progress through interactive activities

* **Test Case ID:** LL-03

* **Test Case Title:** Verify that learners can track their progress through interactive activities

* **Test Case Description:** This test case aims to verify that learners can track their progress through interactive activities such as quizzes, polls, simulations, and discussions.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:**

- * User is logged in to the online learning platform

- * A lesson that tracks learner progress through interactive activities is available

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform as a registered user.
2. Access a lesson that tracks learner progress through interactive activities.
3. Participate in interactive activities and track progress.
4. Verify that the system correctly tracks progress.

* **Postconditions:**

- * The system displays the learner's progress through interactive activities.

- * Progress tracking is accurate and synchronized.

* **Expected Result:** Learners can track their progress through interactive activities accurately.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that feedback is provided after each interactive component

* **Test Case ID:** LL-04

* **Test Case Title:** Verify that feedback is provided after each interactive component

* **Test Case Description:** This test case aims to verify that feedback is provided after each interactive component such as quizzes, polls, simulations, and discussions.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:**

- * User is logged in to the online learning platform

- * A lesson with feedback after interactive components is available

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform as a registered user.
2. Access a lesson that provides feedback after interactive components.
3. Participate in an interactive component and verify that feedback is provided.
4. Evaluate the feedback provided after the interactive component.

* **Postconditions:**

- * The system displays feedback after each interactive component.

- * Feedback is contextually relevant and provides useful insights.

* **Expected Result:** Feedback is provided after each interactive component and is contextually relevant.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that interactive lessons can be accessed on various devices****

* ****Test Case ID:** LL-05**

* ****Test Case Title:** Verify that interactive lessons can be accessed on various devices**

* ****Test Case Description:** This test case aims to verify that interactive lessons are accessible on various devices such as desktops, laptops, tablets, and smartphones.**

* ****Test Suite:** Interactive Lessons**

* ****Test Priority:** High**

* ****Preconditions:****

- * User is logged in to the online learning platform

- * A device of various types (desktop, laptop, tablet, smartphone)

* ****Test Data:** No test data needed**

* ****Test Steps:****

1. Log in to the online learning platform using various devices (desktop, laptop, tablet, smartphone).
2. Access an interactive lesson on each device.
3. Verify that the interactive lesson is accessible and functional on each device.
4. Evaluate the responsiveness and usability of the lesson on various devices.

* ****Postconditions:****

- * The interactive lesson is accessible and functional on various devices.

- * Responsiveness and usability are comparable across devices.

* ****Expected Result:** Interactive lessons can be accessed and are functional on various devices with comparable responsiveness and usability.**

* ****Severity:** Critical**

* ****Priority:** High**

* ****Type of Testing:** Functional Testing**

* ****Test Case Approach:** Positive**

Data - 51

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Personalized Learning Paths

Feature description -

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

Userstory title -

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

Acceptance criteria -

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

Generated Test Cases:

Here are five detailed test cases, one for each acceptance criterion, following the provided template and guidelines:

****Test Case 1: Learners can choose courses and modules based on their interests****

Test Case ID	Test Case Title	Test Case Description
TC-PLP-001	Verify that learners can choose courses and modules based on their interests	Learners can choose courses and modules based on their interests, and the system provides relevant options. This test case will cover the feature's primary functionality.
Test Suite	Personalized Learning Paths	
Test Priority	Medium	
Preconditions	1. Learner account created, 2. Courses and modules available in the system, 3. Learner has a valid login session	
Test Data	None	
Test Steps	1. Log in to the system as a learner, 2. Access the personalized learning paths feature, 3. Browse and select courses and modules based on interests, 4. Verify that the selected options are stored in the system	
Postconditions	None	
Expected Result	Learners can choose courses and modules based on their interests	
Severity	Minor	
Priority	Medium	
Type of Testing	Functional Testing	
Test Case Approach	Positive	

****Test Case 2: Personalized learning paths adjust dynamically based on learner progress****

| Test Case ID | Test Case Title | Test Case Description |

|-----|-----|-----|

| TC-PLP-002 | Verify that personalized learning paths adjust dynamically based on learner progress | Learner progress is tracked and reflected in their learning path. This test case will cover the dynamic adjustment of personalized learning paths. |

| Test Suite | Personalized Learning Paths |

| Test Priority | High |

| Preconditions | 1. Learner account created, 2. Courses and modules available in the system, 3. Learner has a valid login session, 4. Initial course and module selection completed by the learner |

| Test Data | Test data with varied learner progress levels and course information |

| Test Steps | 1. Log in to the system as a learner with completed initial progress, 2. Access the personalized learning paths feature, 3. Observe the system's adjustment of the learning path based on progress, 4. Verify the correctness of the adjusted learning path |

| Postconditions| None |

| Expected Result | The personalized learning path is updated dynamically based on learner progress |

| Severity | Major |

| Priority | High |

| Type of Testing| Functional Testing |

| Test Case Approach| Positive and Negative |

****Test Case 3: Learners receive recommendations for new courses and modules****

| Test Case ID | Test Case Title | Test Case Description |

|-----|-----|-----|

| TC-PLP-003 | Verify that learners receive recommendations for new courses and modules | Learners receive relevant course and module recommendations based on their interests and progress. This test case will cover

the feature's additional functionality. |

| Test Suite | Personalized Learning Paths |

| Test Priority | Medium |

| Preconditions | 1. Learner account created, 2. Courses and modules available in the system, 3. Learner has a valid login session, 4. Initial course and module selection completed by the learner |

| Test Data | Test data with varied learner interests and course information |

| Test Steps | 1. Log in to the system as a learner, 2. Access the personalized learning paths feature, 3. Verify the display of course and module recommendations based on interests and progress, 4. Select a recommendation and verify its relevance and effectiveness |

| Postconditions| None |

| Expected Result | Learners receive relevant recommendations for new courses and modules |

| Severity | Minor |

| Priority | Medium |

| Type of Testing| Functional Testing |

| Test Case Approach| Positive and Negative |

****Test Case 4: Progress tracking is personalized to each learner's path****

| Test Case ID | Test Case Title | Test Case Description |

|-----|-----|-----|

| TC-PLP-004 | Verify that progress tracking is personalized to each learner's path | Progress tracking reflects the learner's unique path and pace. This test case will cover the feature's personalization aspects. |

| Test Suite | Personalized Learning Paths |

| Test Priority | High |

| Preconditions | 1. Learner account created, 2. Courses and modules available in the system, 3. Learner has a valid login session, 4. Initial course and module selection completed by the learner |

| Test Data | Test data with varied learner progress levels and course information |

| Test Steps | 1. Log in to the system as a learner with completed initial progress, 2. Access the progress tracking feature, 3. Verify that the tracking reflects the unique learner path, 4. Validate that the progress is up-to-date and accurate |

| Postconditions| None |

| Expected Result | The progress tracking is personalized to each learner's unique path |

| Severity | Major |

| Priority | High |

| Type of Testing| Functional Testing |

| Test Case Approach| Positive and Negative |

****Test Case 5: Learners can adjust their learning paths at any time****

| Test Case ID | Test Case Title | Test Case Description |

|-----|-----|-----|

| TC-PLP-005 | Verify that learners can adjust their learning paths at any time | Learners can modify their learning paths as their interests or needs change. This test case will cover the feature's flexibility. |

| Test Suite | Personalized Learning Paths |

| Test Priority | Medium |

| Preconditions | 1. Learner account created, 2. Courses and modules available in the system, 3. Learner has a valid login session, 4. Existing course and module selection within the system |

| Test Data | None |

| Test Steps | 1. Log in to the system as an learner, 2. Access the personalized learning paths feature, 3. Remove or add courses and modules as needed, 4. Verify that the adjustments are reflected in the path and progress tracking |

| Postconditions| None |

| Expected Result | Learners can adjust their learning paths as needed |

| Severity | Minor |

| Priority | Medium |

| Type of Testing| Functional Testing |

| Test Case Approach| Positive and Negative |

Data - 52

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Progress Tracking

Feature description -

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

Userstory title -

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Acceptance criteria -

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their

progress.', 'Progress tracking can be exported as reports.']

Generated Test Cases:

Here are five detailed test cases for the Progress Tracking feature of the Online Learning Platform, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that Progress Tracking shows completed activities and grades****

* **Test Case ID:** PROG-001

* **Test Case Title:** Verify that Progress Tracking shows completed activities and grades

* **Test Case Description:** This test case ensures that the progress tracking feature accurately displays completed activities and grades for learners.

* **Test Suite:** Progress Tracking

* **Test Priority:** High

* **Preconditions:**

+ Learner has enrolled in a course with multiple modules and activities.

+ Learner has completed some activities and received grades.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the course dashboard.
2. Click on the progress tracking feature.
3. Verify that the completed activities and grades are displayed correctly.
4. Check that the progress bar accurately reflects the learner's progress.

* **Postconditions:** None

* **Expected Result:** The progress tracking feature displays completed activities and grades accurately.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that Learners can view detailed reports on their progress

* **Test Case ID:** PROG-002

* **Test Case Title:** Verify that Learners can view detailed reports on their progress

* **Test Case Description:** This test case ensures that learners can view detailed reports on their progress, including completed activities, grades, and achievements.

* **Test Suite:** Progress Tracking

* **Test Priority:** High

* **Preconditions:**

+ Learner has enrolled in a course with multiple modules and activities.

+ Learner has completed some activities and received grades.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the course dashboard.
2. Click on the progress tracking feature.
3. Click on the "View Detailed Report" button.
4. Verify that the detailed report displays the correct information.

* **Postconditions:** None

* **Expected Result:** The detailed report accurately displays completed activities, grades, and achievements.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Achievements are highlighted in the progress tracker****

*** **Test Case ID:** PROG-003**

*** **Test Case Title:** Verify that Achievements are highlighted in the progress tracker**

*** **Test Case Description:** This test case ensures that achievements are correctly highlighted in the progress tracker.**

*** **Test Suite:** Progress Tracking**

*** **Test Priority:** Medium**

*** **Preconditions:****

+ Learner has earned achievements in the course.

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in as a learner and navigate to the course dashboard.
2. Click on the progress tracking feature.
3. Verify that the achievements are highlighted correctly.

*** **Postconditions:** None**

*** **Expected Result:** Achievements are correctly highlighted in the progress tracker.**

*** **Severity:** Minor**

*** **Priority:** Low**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

****Test Case 4: Verify that Learners receive notifications about their progress****

*** **Test Case ID:** PROG-004**

*** **Test Case Title:** Verify that Learners receive notifications about their progress**

* **Test Case Description:** This test case ensures that learners receive notifications about their progress, including completed activities and grades.

* **Test Suite:** Progress Tracking

* **Test Priority:** Medium

* **Preconditions:**

+ Learner has completed activities and received grades.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the course dashboard.
2. Verify that the learner receives a notification about their progress.
3. Check that the notification includes the correct information.

* **Postconditions:** None

* **Expected Result:** Learners receive notifications about their progress.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that Progress Tracking can be exported as reports

* **Test Case ID:** PROG-005

* **Test Case Title:** Verify that Progress Tracking can be exported as reports

* **Test Case Description:** This test case ensures that progress tracking can be exported as reports for learners.

* **Test Suite:** Progress Tracking

* **Test Priority:** High

* **Preconditions:**

+ Learner has completed activities and received grades.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the course dashboard.
2. Click on the progress tracking feature.
3. Click on the "Export Report" button.
4. Verify that the report is exported correctly.

* **Postconditions:** None

* **Expected Result:** Progress tracking can be exported as reports.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 53

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Assessment and Grading

Feature description -

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate

learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

Userstory title -

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

Acceptance criteria -

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

Generated Test Cases:

Here are five detailed test cases, each covering one or more of the acceptance criteria from the user story:

****Test Case 1: Verify that instructors can create diverse assessments****

* Test Case ID: TC_001

* Test Case Title: Verify that instructors can create quizzes, tests, and assignments

* Test Suite: Assessment and Grading

* Test Priority: High

* Preconditions:

+ Instructor is logged in to the platform

+ Instructor has access to the course for which they want to create an assessment

* Test Data: No test data needed

* Test Steps:

1. Log in as an instructor and navigate to the course where they want to create an assessment.
2. Click on the "Create Assessment" button to start creating a new quiz, test, or assignment.

3. Select the type of assessment (quiz, test, or assignment) and enter the assessment title.
4. Add questions to the assessment by clicking on the "Add Question" button.
5. Save the assessment and verify that it is created successfully.

* Postconditions:

- + The assessment is created and saved successfully.
- + The assessment is visible in the course navigation menu.

* Expected Result: The instructor should be able to create a quiz, test, or assignment successfully.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 2: Verify that assessments can include various question types****

* Test Case ID: TC_002

* Test Case Title: Verify that assessments can include MCQs, short answers, essays, and other question types

* Test Suite: Assessment and Grading

* Test Priority: Medium

* Preconditions:

- + Instructor is logged in to the platform
- + Instructor has access to the course for which they want to create an assessment

* Test Data: No test data needed

* Test Steps:

1. Log in as an instructor and navigate to the course where they want to create an assessment.
2. Click on the "Create Assessment" button to start creating a new quiz, test, or assignment.
3. Add questions to the assessment by clicking on the "Add Question" button.
4. Select question types such as MCQs, short answers, essays, etc. and add them to the assessment.

5. Save the assessment and verify that it is created successfully.

* Postconditions:

+ The assessment is created and saved successfully.

+ The assessment includes various question types as expected.

* Expected Result: The assessment should include various question types such as MCQs, short answers, essays, etc.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Verify that automatic grading is available for objective questions****

* Test Case ID: TC_003

* Test Case Title: Verify that automatic grading is available for MCQs and other objective questions

* Test Suite: Assessment and Grading

* Test Priority: High

* Preconditions:

+ Instructor is logged in to the platform

+ Instructor has access to the course for which they want to create an assessment

+ Assessment includes objective questions such as MCQs.

* Test Data: No test data needed

* Test Steps:

1. Log in as an instructor and navigate to the course where they want to create an assessment.

2. Click on the "Create Assessment" button to start creating a new quiz, test, or assignment.

3. Add objective questions to the assessment by clicking on the "Add Question" button.

4. Select the question type (MCQ, etc.) and add them to the assessment.

5. Set the automatic grading option for the assessment and save it.

6. Submit the assessment and verify that the grading is done automatically.

* Postconditions:

+ The grading is done automatically as expected.

+ The assessment grades are calculated correctly.

* Expected Result: The grading should be done automatically for objective questions.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Verify that manual grading is available for subjective questions****

* Test Case ID: TC_004

* Test Case Title: Verify that manual grading is available for short answers and essays

* Test Suite: Assessment and Grading

* Test Priority: Medium

* Preconditions:

+ Instructor is logged in to the platform

+ Instructor has access to the course for which they want to create an assessment

+ Assessment includes subjective questions such as short answers and essays.

* Test Data: No test data needed

* Test Steps:

1. Log in as an instructor and navigate to the course where they want to create an assessment.

2. Click on the "Create Assessment" button to start creating a new quiz, test, or assignment.

3. Add subjective questions to the assessment by clicking on the "Add Question" button.

4. Select the question type (short answer, essay, etc.) and add them to the assessment.

5. Set the manual grading option for the assessment and save it.
6. Submit the assessment and verify that the grading is done manually.

* Postconditions:

- + The grading is done manually as expected.
- + The assessment grades are calculated correctly.

* Expected Result: The grading should be done manually for subjective questions.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Verify that feedback is provided to learners after grading****

* Test Case ID: TC_005

* Test Case Title: Verify that learners receive feedback after the assessment is graded

* Test Suite: Assessment and Grading

* Test Priority: High

* Preconditions:

- + Instructor is logged in to the platform
- + Instructor has access to the course for which they want to create an assessment

* Test Data: No test data needed

* Test Steps:

1. Log in as an instructor and navigate to the course where they want to create an assessment.
2. Click on the "Create Assessment" button to start creating a new quiz, test, or assignment.
3. Add questions to the assessment by clicking on the "Add Question" button.
4. Save the assessment and submit it to learners.
5. Allocate grades to learners and verify that they receive feedback after grading.

* Postconditions:

+ Learners receive feedback after the assessment is graded.

+ The feedback is accurate and relevant to the learner's performance.

* Expected Result: Learners should receive feedback after the assessment is graded.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

Data - 54

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Forums

Feature description -

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge sharing, enriching the learning experience through diverse perspectives and active participation.

Userstory title -

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

Acceptance criteria -

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all devices.']

Generated Test Cases:

Here are 5 detailed test cases for the Discussion Forums feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that Learners can post questions and responses in forums****

Test Case ID: DC01

Test Case Title: Verify that Learners can post questions and responses in forums

Test Case Description: As a learner, I want to post questions and responses in discussion forums, so that I can engage with peers and instructors. This test case ensures that learners can successfully post questions and responses in forums.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- * The learner is logged in to the online learning platform.

- * The learner has access to the discussion forums.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the discussion forums page.
3. Click on the "Post a Question" or "Post a Response" button.
4. Enter a valid question or response in the required field.
5. Click the "Post" button.

Postconditions:

- * The posted question or response appears in the discussion forum.
- * The learner receives a confirmation message.

Expected Result: The learner can successfully post a question or response in the discussion forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that Forums support threaded discussions****

Test Case ID: DC02

Test Case Title: Verify that Forums support threaded discussions

Test Case Description: As a learner, I want to see threaded discussions in the discussion forums, so that I can easily follow conversations. This test case ensures that forums support threaded discussions.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- * The learner is logged in to the online learning platform.
- * The discussion forums are enabled and accessible.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the discussion forums page.
3. Post a question or response in the discussion forum.
4. Respond to the initial post with a new question or response.
5. Verify that the responses are threaded and easily visible.

Postconditions:

- * The threaded discussion is visible in the discussion forum.

Expected Result: The discussion forum showcases threaded discussions, allowing learners to easily follow conversations.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that Instructors can moderate discussions****

Test Case ID: DC03

Test Case Title: Verify that Instructors can moderate discussions

Test Case Description: As an instructor, I want to moderate discussions in the discussion forums, so that I can ensure a safe and respectful learning environment. This test case ensures that instructors can moderate discussions effectively.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- * The instructor is logged in to the online learning platform.
- * The discussion forums are enabled and accessible.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the discussion forums page.
3. Identify an off-topic or inflammatory post.
4. Click on the "Moderate" button next to the post.

5. Verify that the post is removed or marked as spam.

Postconditions:

- * The post is removed or marked as spam.

Expected Result: Instructors can effectively moderate discussions, ensuring a safe and respectful learning environment.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that Learners receive notifications for new posts****

Test Case ID: DC04

Test Case Title: Verify that Learners receive notifications for new posts

Test Case Description: As a learner, I want to receive notifications for new posts in the discussion forums, so that I can stay up-to-date with the latest discussions. This test case ensures that learners receive notifications for new posts.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- * The learner is logged in to the online learning platform.
- * The discussion forums are enabled and accessible.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the discussion forums page.
3. Post a new question or response.

4. Enable notifications for new posts (if available).
5. Verify that the learner receives a notification when a new post is published.

Postconditions:

- * The learner receives a notification for the new post.

Expected Result: Learners receive timely notifications for new posts in the discussion forums.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that Forums are accessible on all devices****

Test Case ID: DC05

Test Case Title: Verify that Forums are accessible on all devices

Test Case Description: As a learner, I want to access the discussion forums on various devices, so that I can engage with peers and instructors regardless of my device. This test case ensures that discussion forums are accessible on all devices.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- * The learner has access to the discussion forums on various devices (e.g., desktop, laptop, tablet, smartphone).

Test Data: No test data needed

Test Steps:

1. Access the discussion forums on a desktop.
2. Verify that the forums are fully functional and accessible.
3. Access the discussion forums on a mobile device (e.g., smartphone or tablet).

4. Verify that the forums are fully functional and accessible.

Postconditions:

* The discussion forums are accessible on all devices.

Expected Result: Discussion forums are accessible and functional on all devices, ensuring learners can engage with peers and instructors regardless of their device.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Cross-Browser 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:

Here are five detailed test cases for the "Live Classes and Webinars" feature of the Online Learning Platform, following the Test Case Creation Template:

****Test Case 1: Live Classes and Webinars Conducted through Video Conferencing Tools****

* ****Test Case ID:** LCV-001**

* ****Test Case Title:** Verify that live classes and webinars are conducted through video conferencing tools.**

* ****Test Suite:** Live Classes and Webinars**

* ****Test Priority:** High**

* ****Preconditions:****

* User logs in to the platform.

* Instructor schedules a live class or webinar.

* Video conferencing tool is enabled.

* ****Test Data:** No test data needed**

* ****Test Steps:****

1. Log in to the platform as a learner.
2. Search for a live class or webinar and join it.
3. Verify that the instructor's presentation and video feed are visible on the screen.
4. Check that the video conferencing tool is functional and allows for real-time interaction.
5. Record the session and verify that it is saved for later viewing.

* ****Postconditions:****

- * Session is recorded and available for later viewing.
- * Instructor's presentation and video feed are visible on the screen.
- * Video conferencing tool is functional.

* ****Expected Result:**** Live classes and webinars are conducted through video conferencing tools, enabling instructor and learner interaction in real-time.

* ****Severity:**** Critical

* ****Priority:**** High

* ****Type of Testing:**** Functional Testing

* ****Test Case Approach:**** Positive

****Test Case 2: Learners Ask Questions in Real-Time****

* ****Test Case ID:**** LCV-002

* ****Test Case Title:**** Verify that learners can ask questions in real-time during live classes and webinars.

* ****Test Suite:**** Live Classes and Webinars

* ****Test Priority:**** High

* ****Preconditions:****

- * User logs in to the platform.
- * Instructor schedules a live class or webinar.
- * Learners join the session.

* ****Test Data:**** No test data needed

* ****Test Steps:****

1. Log in to the platform as a learner.
2. Join a live class or webinar.
3. Ask a question through the platform's Q&A feature.
4. Verify that the instructor receives the question and responds accordingly.

5. Check that the question and response are visible to all learners.

* **Postconditions:**

* Instructor receives the question and responds accordingly.

* Learners' questions are visible to all.

* Real-time interaction is possible between learners and instructors.

* **Expected Result:** Learners can ask questions in real-time during live classes and webinars, facilitating interaction and engagement.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Sessions are Recorded and Available for Later Viewing

* **Test Case ID:** LCV-003

* **Test Case Title:** Verify that sessions are recorded and available for later viewing.

* **Test Suite:** Live Classes and Webinars

* **Test Priority:** Medium

* **Preconditions:**

* Instructor schedules a live class or webinar.

* Session recording is enabled.

* **Test Data:** No test data needed

* **Test Steps:**

1. As a learner, join a live class or webinar.

2. Record the session and verify that it saves properly.

3. Access the recorded session from the platform's repository or catalog.

4. Check that the recorded session is visible and can be played back.

* **Postconditions:**

- * Session is successfully recorded and saved.
- * Recorded session is accessible and visible to learners.
- * Session can be played back without issues.

* **Expected Result:** Sessions are recorded and available for later viewing, providing learners with flexibility and convenience.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Instructors Can Share Screen and Resources during Sessions****

* **Test Case ID:** LCV-004

* **Test Case Title:** Verify that instructors can share screen and resources during live classes and webinars.

* **Test Suite:** Live Classes and Webinars

* **Test Priority:** High

* **Preconditions:**

- * User logs in to the platform.
- * Instructor schedules a live class or webinar.
- * Instructor has relevant resources and screen content.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as an instructor.
2. Schedule a live class or webinar and enable screen sharing.
3. Share relevant resources, such as presentations, documents, or videos.

4. Verify that learners can see the shared content in real-time.
5. Check that screen sharing and resource sharing are seamless and smooth.

* **Postconditions:**

* Instructors can successfully share screen and resources.

* Learners can see shared content in real-time.

* Screen sharing and resource sharing are smooth and seamless.

* **Expected Result:** Instructors can share screen and resources during live classes and webinars, enhancing the learning experience and facilitating real-time interaction.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Notifications are Sent Before Sessions Start

* **Test Case ID:** LCV-005

* **Test Case Title:** Verify that notifications are sent before sessions start, reminding learners of upcoming live classes and webinars.

* **Test Suite:** Live Classes and Webinars

* **Test Priority:** Medium

* **Preconditions:**

* Instructor schedules a live class or webinar.

* Notification settings are enabled.

* **Test Data:** No test data needed

* **Test Steps:**

1. As a learner, create a calendar event or set reminders before an upcoming session.
2. Contact instructor or platform support to verify notification settings.

3. Log in to the platform and verify that a notification is sent before the session starts.
 4. Check that the notification is visible in the platform's notification center or email.
- * ****Postconditions:****
 - * Learners receive notifications before sessions start.
 - * Notification settings are properly configured.
 - * Learners are aware of upcoming sessions and can prepare accordingly.
 - * ****Expected Result:**** Notifications are sent before sessions start, reminding learners of upcoming live classes and webinars and ensuring they stay engaged and on track.
 - * ****Severity:**** Minor
 - * ****Priority:**** Medium
 - * ****Type of Testing:**** Functional Testing
 - * ****Test Case Approach:**** Positive

Each of these test cases addresses a specific aspect of the "Live Classes and Webinars" feature, covering the key acceptance criteria as outlined in the user story.

Data - 56

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Mobile Accessibility

Feature description -

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

Userstory title -

As a learner, I want mobile access to the platform so that I can learn on the go.

Acceptance criteria -

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

Generated Test Cases:

Here are five detailed test cases that cover each acceptance criterion from the user story:

****Test Case 1: Mobile Platform Accessibility****

Test Case ID: TC001

Test Case Title: Verify that the platform is accessible on smartphones and tablets.

Test Case Description: This test case aims to ensure that the online learning platform can be accessed and navigated seamlessly on various smartphone and tablet devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- * Ensure that the device has a stable internet connection.
- * Ensure that the device has the required version of the platform's app.

Test Data: No test data needed

Test Steps:

1. Launch the platform on a smartphone and tablet device.
2. Verify that the login page is accessible and loads correctly.
3. Perform basic navigation (clicking on menu items, browsing through courses).
4. Verify that the content is rendered correctly and is viewable.
5. Test various screen orientations (portrait, landscape).

Postconditions:

- * Verify that no errors are reported.
- * Verify that the device is functioning as expected.

Expected Result: The platform should be accessible and functional on smartphones and tablets.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Mobile UI Optimization****

Test Case ID: TC002

Test Case Title: Verify that the mobile UI is optimized for touch navigation.

Test Case Description: This test case ensures that the mobile user interface is responsive to touch input, providing a seamless user experience.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- * Ensure that the device has a stable internet connection.

Test Data: No test data needed

Test Steps:

1. Log in to the platform on a device with a touchscreen.
2. Verify that all UI elements are accessible through touch input (e.g., buttons, links, images).
3. Test pinch-to-zoom functionality.
4. Verify that the UI responds to taps and swipe gestures.
5. Test scrolling and loading of content.

Postconditions:

* Verify that no errors are reported.

Expected Result: The mobile UI should be optimized for touch navigation.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Mobile-Accessible Course Materials****

Test Case ID: TC003

Test Case Title: Verify that course materials are accessible and readable on mobile devices.

Test Case Description: This test case ensures that course materials, such as text, images, and videos, are viewable and readable on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

* Ensure that the device has a stable internet connection.

* Ensure that course materials are uploaded to the platform.

Test Data: Test with a mix of text, images, and video files.

Test Steps:

1. Launch the platform on a device with a mobile browser.

2. Navigate to a course and select a module.
3. Verify that course materials are viewable and readable.
4. Test rendering of multimedia content (e.g., videos).
5. Verify that documents are opened correctly (PDF, Word, etc.).

Postconditions:

- * Verify that no errors are reported.

Expected Result: Course materials should be accessible and readable on mobile devices.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Mobile Notifications****

Test Case ID: TC004

Test Case Title: Verify that notifications are sent to mobile devices.

Test Case Description: This test case ensures that the platform sends notifications to mobile devices for important events, such as new course enrollments or reminders.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- * Ensure that the device has push notifications enabled.

Test Data: Test with different notification scenarios (e.g., course enrollment, reminders).

Test Steps:

1. Enable push notifications on the device.
2. Log in to the platform and trigger a notification (e.g., enroll in a course).
3. Verify that the notification appears on the device.

4. Test notification delivery and timing.
5. Verify that the notification is dismissed correctly.

Postconditions:

- * Verify that no errors are reported.

Expected Result: The platform should send notifications to mobile devices for important events.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Offline Access to Downloaded Materials****

Test Case ID: TC005

Test Case Title: Verify that the platform supports offline access to downloaded materials.

Test Case Description: This test case ensures that learners can access downloaded course materials offline, even with a stable internet connection.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- * Ensure that the device has a stable internet connection.
- * Ensure that course materials are downloaded to the platform.

Test Data: Test with a mix of file types (e.g., PDF, Word, Image).

Test Steps:

1. Download course materials on the platform.
2. Launch the platform on the device with the downloaded materials.
3. Verify that the materials are accessible without an internet connection.
4. Test rendering of downloaded multimedia content (e.g., videos).

5. Verify that documents are opened correctly (PDF, Word, etc.).

Postconditions:

* Verify that no errors are reported.

Expected Result: The platform should support offline access to downloaded materials.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

These five test cases cover the required acceptance criteria and are designed to ensure the online learning platform's mobile accessibility feature is thoroughly tested and functional.

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:

Here are five detailed test cases that cover the acceptance criteria for the Resource Library feature:

****Test Case 1: Verify that the Resource Library includes e-books, articles, and reference guides****

*** **Test Case ID:** RL001**

*** **Test Case Title:** Verify that the Resource Library includes e-books, articles, and reference guides**

*** **Test Case Description:** This test case ensures that the Resource Library contains the specified types of resources, including e-books, articles, and reference guides.**

*** **Test Suite:** Resource Library Module**

*** **Test Priority:** High**

*** **Preconditions:** User account is created and logged in**

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the online learning platform.
2. Navigate to the Resource Library module.
3. Verify that the Resource Library includes e-books, articles, and reference guides.
4. Check that the resources are categorized and tagged for easy navigation.

* **Postconditions:** Resource Library modules are updated with the expected resources.

* **Expected Result:** The Resource Library includes e-books, articles, and reference guides.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case Approach: Positive

Test Case 2: Verify that learners can search for specific materials

* **Test Case ID:** RL002

* **Test Case Title:** Verify that learners can search for specific materials

* **Test Case Description:** This test case ensures that learners can search for specific resources within the Resource Library.

* **Test Suite:** Resource Library Module

* **Test Priority:** Medium

* **Preconditions:** User account is created and logged in

* **Test Data:** Search query (e.g., "economics")

* **Test Steps:**

1. Log in to the online learning platform.
2. Navigate to the Resource Library module.
3. Enter a search query in the search bar.
4. Click the search button to retrieve results.
5. Verify that the search results contain relevant resources.

* **Postconditions:** Search functionality is updated with the expected results.

* **Expected Result:** Learners can search for specific materials and receive relevant results.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

Test Case Approach: Positive

Test Case 3: Verify that resources are categorized and tagged for easy navigation

* **Test Case ID:** RL003

* **Test Case Title:** Verify that resources are categorized and tagged for easy navigation

* **Test Case Description:** This test case ensures that resources in the Resource Library are categorized and tagged for easy navigation.

* **Test Suite:** Resource Library Module

* **Test Priority:** High

* **Preconditions:** User account is created and logged in

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform.
2. Navigate to the Resource Library module.
3. Verify that resources are categorized (e.g., e-books, articles, reference guides).
4. Verify that resources are tagged with relevant keywords.
5. Check that the tags are clickable and lead to relevant search results.

* **Postconditions:** Resource categorization and tagging are updated with the expected structure.

* **Expected Result:** Resources are categorized and tagged for easy navigation.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

****Test Case Approach:** Positive**

****Test Case 4: Verify that learners can download resources for offline access****

*** **Test Case ID:** RL004**

*** **Test Case Title:** Verify that learners can download resources for offline access**

*** **Test Case Description:** This test case ensures that learners can download resources from the Resource Library for offline access.**

*** **Test Suite:** Resource Library Module**

*** **Test Priority:** Medium**

*** **Preconditions:** User account is created and logged in**

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the online learning platform.
2. Navigate to the Resource Library module.
3. Select a resource to download.
4. Verify that the download process completes successfully.
5. Check that the downloaded resource can be accessed offline.

*** **Postconditions:** Download functionality is updated with the expected results.**

*** **Expected Result:** Learners can download resources for offline access.**

*** **Severity:** Major**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 5: Verify that instructors can recommend resources to learners****

* **Test Case ID:** RL005

* **Test Case Title:** Verify that instructors can recommend resources to learners

* **Test Case Description:** This test case ensures that instructors can recommend resources to learners within the Resource Library.

* **Test Suite:** Resource Library Module

* **Test Priority:** High

* **Preconditions:** User account is created and logged in (instructor role)

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform (as instructor).
2. Navigate to the Resource Library module.
3. Select a resource to recommend.
4. Verify that the recommendation process completes successfully.
5. Check that the learner receives the recommended resource.

* **Postconditions:** Recommendation functionality is updated with the expected results.

* **Expected Result:** Instructors can recommend resources to learners.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case Approach: Positive

Each test case covers a specific acceptance criterion and includes detailed steps for execution. By following these test cases, we can ensure that the Resource Library feature meets the expected requirements and provides a high-quality user experience.

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 five detailed test cases that adhere to the provided template and cover all the acceptance criteria of the user story:

****Test Case 1: Verify that learners can submit assignments for peer review****

* **Test Case ID:** PR-001

* **Test Case Title:** Verify that learners can submit assignments for peer review

* **Test Suite:** Peer Review Module

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the online learning platform

+ User has created an assignment

* **Test Data:** No test data needed

* **Test Steps:**

1. Users logs in to the online learning platform

2. Users navigates to the assignment submission page

3. Users uploads the assignment file

4. Users submits the assignment for peer review

* **Postconditions:** Assignment is displayed in the peer review list with the option to view submitted assignments

* **Expected Result:** Assignment is successfully submitted for peer review

* **Severity:** Medium

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that peers can provide feedback on submitted assignments

* **Test Case ID:** PR-002

* **Test Case Title:** Verify that peers can provide feedback on submitted assignments

* **Test Suite:** Peer Review Module

* **Test Priority:** High

* **Preconditions:**

+ Assignment is submitted for peer review

+ User is a peer of the assignment submitter

* **Test Data:** No test data needed

* **Test Steps:**

1. Peer navigates to the peer review list

2. Peer selects the assignment to review

3. Peer provides feedback on the assignment

4. Peer submits the feedback

* **Postconditions:** Feedback is displayed on the assignment submission page

* **Expected Result:** Feedback is successfully submitted and displayed on the assignment submission page

* **Severity:** Medium

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that a structured review process guides the feedback****

* **Test Case ID:** PR-003

* **Test Case Title:** Verify that a structured review process guides the feedback

* **Test Suite:** Peer Review Module

* **Test Priority:** Medium

* **Preconditions:**

+ Assignment is submitted for peer review

+ User is a peer of the assignment submitter

* **Test Data:** No test data needed

* **Test Steps:**

1. Peer navigates to the peer review list
2. Peer selects the assignment to review
3. Peer views the structured review process guidelines
4. Peer provides feedback based on the guidelines

* **Postconditions:** Feedback is displayed on the assignment submission page

* **Expected Result:** Feedback is guided by the structured review process

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that learners can view feedback on their work****

* **Test Case ID:** PR-004

* **Test Case Title:** Verify that learners can view feedback on their work

* **Test Suite:** Peer Review Module

* **Test Priority:** High

* **Preconditions:**

+ Assignment is submitted for peer review

+ User is the assignment submitter

* **Test Data:** No test data needed

* **Test Steps:**

1. User navigates to the assignment submission page
2. User views the feedback on the assignment submission page
3. User verifies the feedback content

* **Postconditions:** Feedback is displayed on the assignment submission page

* **Expected Result:** Feedback is successfully viewed by the learner

* **Severity:** Medium

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that reviewers remain anonymous to ensure unbiased feedback

* **Test Case ID:** PR-005

* **Test Case Title:** Verify that reviewers remain anonymous to ensure unbiased feedback

* **Test Suite:** Peer Review Module

* **Test Priority:** High

* **Preconditions:**

+ Assignment is submitted for peer review

+ User is a peer of the assignment submitter

* **Test Data:** No test data needed

* **Test Steps:**

1. Peer navigates to the peer review list

2. Peer views the reviewer information

3. Peer verifies that reviewer information is anonymous

* **Postconditions:** Reviewer information is not displayed

* **Expected Result:** Reviewer information is successfully kept anonymous

* **Severity:** Medium

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 59

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Certificates and Badges

Feature description -

Upon successful completion of courses or specific milestones, learners receive certificates and badges to recognize their achievements and accomplishments. These credentials provide tangible recognition of learners' efforts, skills, and expertise, enhancing their credentials and demonstrating their commitment to continuous learning and professional development.

Userstory title -

As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

Acceptance criteria -

['Learners earn certificates upon course completion.', 'Badges are awarded for specific achievements and milestones.', 'Certificates and badges are accessible through learner profiles.', 'Learners can share their certificates and badges on social media.', 'Instructors can issue custom badges for special achievements.']

Generated Test Cases:

Here are 5 detailed test cases that cover all the acceptance criteria related to the "Certificates and Badges" feature of the Online Learning Platform:

****Test Case 1: Verify that Learners Earn Certificates Upon Course Completion****

* Test Case ID: TCD-001

* Test Case Title: Verify that learners earn certificates upon course completion

* Test Case Description: Ensure that learners receive a certificate upon completing a course, and the certificate is recorded in their learner profile.

* Test Suite: Certificates and Badges

* Test Priority: High

* Preconditions:

+ Learner is enrolled in a course with a completion requirement

+ The course is complete and marked as such by the learner

* Test Data: Course ID, Learner ID

* Test Steps:

1. Navigate to the learner dashboard

2. Complete the course and confirm completion

3. Verify that a certificate is generated and recorded in the learner profile

* Postconditions:

+ The certificate is accessible via the learner profile

+ The certificate reflects the course completion date and details

* Expected Result: The learner receives a certificate upon course completion, and it is stored in their learner profile.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 2: Verify that Badges are Awarded for Specific Achievements and Milestones****

* Test Case ID: TCD-002

* Test Case Title: Verify that badges are awarded for specific achievements and milestones

* Test Case Description: Test the process of awarding badges for learners who achieve specific milestones or complete specific tasks within a course.

* Test Suite: Certificates and Badges

* Test Priority: High

* Preconditions:

+ A course with defined badges and achievements is created

+ Learner meets the criteria for a badge (e.g., completes a specific task or achieves a certain score)

* Test Data: Course ID, Badge ID, Learner ID

* Test Steps:

1. Set up the badge criteria (e.g., complete a module, achieve a score)

2. Have the learner meet the badge criteria

3. Verify that the badge is awarded to the learner

* Postconditions:

+ The badge is recorded in the learner's profile

+ The badge reflects the achievement or milestone

* Expected Result: The learner receives a badge for achieving the defined milestone or completing the specific task.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Verify that Certificates and Badges are Accessible Through Learner Profiles****

* Test Case ID: TCD-003

* Test Case Title: Verify that certificates and badges are accessible through learner profiles

* Test Case Description: Test the visibility of certificates and badges within the learner profile.

* Test Suite: Certificates and Badges

* Test Priority: Medium

* Preconditions:

+ Learner has earned a certificate or badge

+ The learner is logged in to their profile

* Test Data: Learner ID

* Test Steps:

1. Navigate to the learner profile

2. Verify that the earned certificate or badge is displayed

3. Confirm that the certificate or badge details are accurate

* Postconditions:

+ The certificate or badge is visible in the learner profile

+ The certificate or badge details are correct

* Expected Result: Learners can view earned certificates and badges within their profile.

* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Verify that Learners Can Share Certificates and Badges on Social Media****

* Test Case ID: TCD-004

* Test Case Title: Verify that learners can share certificates and badges on social media

* Test Case Description: Test the functionality of sharing earned certificates and badges on social media

platforms.

* Test Suite: Certificates and Badges

* Test Priority: Medium

* Preconditions:

+ Learner has earned a certificate or badge

+ The learner is logged in to their social media account

* Test Data: Learner ID, Social Media Account

* Test Steps:

1. Navigate to the learner profile

2. Select the earned certificate or badge to share

3. Share on social media using the provided link or option

* Postconditions:

+ The shared certificate or badge is visible on the social media platform

+ The shared certificate or badge details are accurate

* Expected Result: Learners can share earned certificates and badges on social media platforms.

* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Verify that Instructors Can Issue Custom Badges for Special Achievements****

* Test Case ID: TCD-005

* Test Case Title: Verify that instructors can issue custom badges for special achievements

* Test Case Description: Test the ability of instructors to create and assign custom badges for specific achievements within courses.

* Test Suite: Certificates and Badges

* Test Priority: High

* Preconditions:

- + Instructors have the necessary permissions
- + The course allows for instructor-created badges

* Test Data: Course ID, Instructor ID

* Test Steps:

1. Create a custom badge in the course settings
2. Configure the badge criteria (e.g., complete a specific task, achieve a certain score)
3. Verify that the badge is created and available for awarding

* Postconditions:

- + The custom badge is created and available in the course settings
- + The badge reflects the specific achievement or milestone

* Expected Result: Instructors can create and assign custom badges for special achievements within courses.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

Each test case follows the provided template and covers specific aspects of the "Certificates and Badges" feature. These test cases will help identify potential defects and ensure the feature's quality, demonstrating a comprehensive testing approach.

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:

Here are five detailed test cases that cover the acceptance criteria for the Gamification Elements feature:

****Test Case 1: Verify that learners earn points for completing activities****

*** **Test Case ID:** GAM-001**

*** **Test Case Title:** Verify that learners earn points for completing activities**

*** **Test Suite:** Gamification Elements**

*** **Test Priority:** High**

****Preconditions:****

- + The learner is logged in to the platform
- + The activity assigned to the learner has a point value associated with it

****Test Data:**** No test data needed

****Test Steps:****

1. The learner logs in and accesses the activity dashboard
2. The learner completes the activity
3. The learner views their activity history to verify that points have been earned

****Postconditions:****

- + Points are reflected in the learner's progress dashboard
- + Points are updated in real-time

****Expected Result:**** Learners earn points for completing activities.

****Severity:**** Blocker

****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:**** GAM-002

****Test Case Title:**** Verify that levels are achieved based on accumulated points

****Test Suite:**** Gamification Elements

****Test Priority:**** High

****Preconditions:****

- + The learner has earned points
- + The learner's current level is known

****Test Data:**** No test data needed

****Test Steps:****

1. The learner logs in and views their progress dashboard
2. The learner earns additional points
3. The learner views their progress dashboard to verify that their level has been updated

****Postconditions:****

- + The learner's level is updated in real-time
- + The learner's progress dashboard reflects their new level

****Expected Result:**** Learners achieve levels based on accumulated points.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 3: Verify that leaderboards display top-performing learners****

****Test Case ID:**** GAM-003

****Test Case Title:**** Verify that leaderboards display top-performing learners

****Test Suite:**** Gamification Elements

****Test Priority:**** High

****Preconditions:****

- + The learner has earned points
- + The leaderboard is enabled

****Test Data:**** No test data needed

****Test Steps:****

1. The learner logs in and views the leaderboard
2. The learner verifies that the top-performing learners are displayed correctly
3. The learner views their own ranking on the leaderboard

* **Postconditions:**

+ The leaderboard is updated in real-time

+ The learner's ranking is accurate

* **Expected Result:** Leaderboards display top-performing learners.

* **Severity:** Major

* **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:** GAM-004

* **Test Case Title:** Verify that badges are awarded for reaching milestones

* **Test Suite:** Gamification Elements

* **Test Priority:** High

* **Preconditions:**

+ The learner has earned points

+ The badge system is enabled

* **Test Data:** No test data needed

* **Test Steps:**

1. The learner logs in and views their progress dashboard

2. The learner earns points to reach a milestone

3. The learner views their badge collection to verify that the milestone badge is awarded

* **Postconditions:**

+ The badge is awarded in real-time

+ The badge is displayed in the learner's badge collection

* **Expected Result:** Badges are awarded for reaching milestones.

* **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:** GAM-005

* **Test Case Title:** Verify that learners can view their progress in a gamification dashboard

* **Test Suite:** Gamification Elements

* **Test Priority:** High

* **Preconditions:**

+ The learner has earned points

+ The gamification dashboard is enabled

* **Test Data:** No test data needed

* **Test Steps:**

1. The learner logs in and views their progress dashboard
2. The learner verifies that their progress is displayed correctly
3. The learner views their history and current level

* **Postconditions:**

+ The gamification dashboard is updated in real-time

+ The learner's progress is accurate

* **Expected Result:** Learners can view their progress in a gamification dashboard.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases provide comprehensive coverage of the Gamification Elements feature, including points earning, level achievement, leaderboard display, badge awarding, and progress visualization. Each test case is designed to test a specific acceptance criterion or combination of criteria, ensuring that the feature is robust and functional.

Data - 61

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Multilingual Support

Feature description -

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

Userstory title -

As a learner, I want multilingual support so that I can access the platform in my preferred language.

Acceptance criteria -

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated

help and support resources.', 'Instructors can provide content in multiple languages.']

Generated Test Cases:

Here are five detailed test cases that cover all acceptance criteria for the Multilingual Support feature:

****Test Case 1: Multilingual Interface****

* ****Test Case ID:**** MSL001

* ****Test Case Title:**** Verify that the platform interface is available in multiple languages.

* ****Test Case Description:**** This test case will verify that the platform's interface is displayed in multiple languages when changed by learners or when a learner's preferred language is set. Test case will also verify that the language change does not affect the overall functionality of the platform.

* ****Test Suite:**** Multilingual Support

* ****Test Priority:**** High

* ****Preconditions:**** User is logged in to the platform with a valid account.

* ****Test Data:**** No test data needed

* ****Test Steps:****

1. Log in to the platform with a valid account.
2. Navigate to the settings or account profile page.
3. Click on the language dropdown menu and select a language other than the default language.
4. Verify that the platform interface is updated to the selected language.
5. Repeat steps 3-4 with multiple languages to ensure language switch functionality is robust.

* ****Postconditions:**** No follow-up actions required.

* ****Expected Result:**** The platform interface is displayed in the selected language.

* ****Severity:**** Major

* ****Priority:**** High

* ****Type of Testing:**** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Multilingual Course Materials

* **Test Case ID:** MSL002

* **Test Case Title:** Verify that course materials are available in different languages.

* **Test Case Description:** This test case will verify that course materials, such as PDF files, videos, and slides, are available in different languages when learner switches language. Test case will also verify that the language change does not affect the content of the materials.

* **Test Suite:** Multilingual Support

* **Test Priority:** Medium

* **Preconditions:** User has enrolled in a course with content available in multiple languages.

* **Test Data:** Course content with multilingual support.

* **Test Steps:**

1. Log in to the platform with a valid account.
2. Navigate to the course dashboard.
3. Switch language to a language other than the default language.
4. Verify that the course materials are updated to the selected language.
5. Verify that the language change does not affect the content of the materials.

* **Postconditions:** No follow-up actions required.

* **Expected Result:** Course materials are available in the selected language, with no changes to content.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Learner Preferences

* **Test Case ID:** MSL003

* **Test Case Title:** Verify that learners can select their preferred language at any time.

* **Test Case Description:** This test case will verify that learners can change their preferred language at any time while logged in to the platform. Test case will also verify that the change is reflected across all platform features.

* **Test Suite:** Multilingual Support

* **Test Priority:** Low

* **Preconditions:** User is logged in to the platform with a valid account.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform with a valid account.
2. Navigate to the settings or account profile page.
3. Change the preferred language to a different language.
4. Verify that the platform features, including course materials, forums, and progress tracking, are updated to the new language.

* **Postconditions:** No follow-up actions required.

* **Expected Result:** Learner's preferred language is updated, and all platform features reflect the change.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Multilingual Support Resources

* **Test Case ID:** MSL004

* **Test Case Title:** Verify that multilingual support includes translated help and support resources.

* **Test Case Description:** This test case will verify that help and support resources, such as FAQs, tutorials, and contact information, are available in multiple languages. Test case will also verify that the language change does not affect the availability of these resources.

* **Test Suite:** Multilingual Support

* **Test Priority:** Medium

* **Preconditions:** User is logged in to the platform with a valid account.

* **Test Data:** Help and support resources with multilingual support.

* **Test Steps:**

1. Log in to the platform with a valid account.
2. Navigate to the help or support page.
3. Switch language to a language other than the default language.
4. Verify that the help and support resources are available in the selected language.

* **Postconditions:** No follow-up actions required.

* **Expected Result:** Help and support resources are available in the selected language.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Instructor-Supervised Content****

* **Test Case ID:** MSL005

* **Test Case Title:** Verify that instructors can provide content in multiple languages.

* **Test Case Description:** This test case will verify that instructors can create and upload content in multiple languages. Test case will also verify that the language change does not affect the functionality of the content.

* **Test Suite:** Multilingual Support

* **Test Priority:** High

* **Preconditions:** Instructor account is created and validated.

* **Test Data:** Content with multilingual support.

* **Test Steps:**

1. Log in as an instructor with a valid account.
2. Create a new course or upload existing content.
3. Verify that language options are available for the content.
4. Upload content in a language other than the default language.
5. Verify that the content is displayed correctly in the selected language.

* **Postconditions:** No follow-up actions required.

* **Expected Result:** Instructor can create content in multiple languages.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional 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 of the Online Learning Platform:

****Test Case 1: Verify that the platform is compatible with screen readers****

*** **Test Case ID:** TCACC-001**

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

*** **Test Case Description:** This test case will verify that the platform is accessible to learners with visual impairments using screen readers, ensuring a seamless experience.**

*** **Test Suite:** Accessibility Features**

*** **Test Priority:** High**

*** **Preconditions:****

+ The learner's device has a screen reader installed and configured.

+ The learner has access to a compatible browser.

*** **Test Data:** No test data needed**

* **Test Steps:**

1. Launch the platform using a device with a screen reader installed.
2. Navigate through the platform's main pages, including course listings, video lectures, and discussion forums.
3. Test each interactive element (e.g., buttons, links, form fields) to ensure screen reader compatibility.

* **Postconditions:**

- + Verify that the screen reader provides an overview of the page structure and content.
- + Ensure that the screen reader accurately identifies interactive elements.

* **Expected Result:** The platform is fully compatible with screen readers, allowing learners with visual impairments to navigate and interact with the platform seamlessly.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that learners can adjust font sizes for better readability

* **Test Case ID:** TCACC-002

* **Test Case Title:** Verify that learners can adjust font sizes for better readability

* **Test Case Description:** This test case will verify that learners can adjust font sizes within the platform to improve readability, ensuring a comfortable user experience.

* **Test Suite:** Accessibility Features

* **Test Priority:** Medium

* **Preconditions:**

- + The learner has access to a device with a compatible browser.

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the platform on a device with a compatible browser.
2. Navigate to a page with a large amount of text (e.g., a course description page).
3. Adjust the font size using the font size options provided.
4. Verify that the text adjusts accordingly.

* **Postconditions:** Verify that the font size adjustment is reflected across all pages of the platform.

* **Expected Result:** Learners can adjust font sizes to improve readability, making the platform more accessible and user-friendly.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that color contrast options are available for improved visibility****

* **Test Case ID:** TCACC-003

* **Test Case Title:** Verify that color contrast options are available for improved visibility

* **Test Case Description:** This test case will verify that learners can adjust color contrast settings to ensure improved visibility of platform content, particularly in environments with low light or other visual challenges.

* **Test Suite:** Accessibility Features

* **Test Priority:** Medium

* **Preconditions:**

+ The learner has access to a device with a compatible browser.

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the platform on a device with a compatible browser.
2. Navigate to a page with background images or gradients.
3. Adjust the color contrast settings using the provided options.

4. Verify that the background image or gradient recedes, improving visibility.

* **Postconditions:** Verify that the color contrast adjustment is reflected across all pages of the platform.

* **Expected Result:** Learners can adjust color contrast settings to improve visibility, ensuring equal access to all platform content.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that accessibility features are easy to activate and customize

* **Test Case ID:** TCACC-004

* **Test Case Title:** Verify that accessibility features are easy to activate and customize

* **Test Case Description:** This test case will verify that accessibility features are intuitive and easy to activate, allowing learners to customize their experience as needed.

* **Test Suite:** Accessibility Features

* **Test Priority:** Medium

* **Preconditions:**

+ The learner has access to a device with a compatible browser.

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the platform on a device with a compatible browser.

2. Navigate to the accessibility feature settings page.

3. Test the intuitive nature of the accessibility feature activation and customization process.

4. Verify that the system provides clear instructions or hints for first-time users.

* **Postconditions:** Verify that the accessibility features are easily accessible and customizable.

* **Expected Result:** Learners can easily activate and customize accessibility features to suit their needs,

enhancing the overall user experience.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that support resources are available to assist learners with disabilities

* **Test Case ID:** TCACC-005

* **Test Case Title:** Verify that support resources are available to assist learners with disabilities

* **Test Case Description:** This test case will verify that additional support resources, such as documentation, tutorials, or live support, are easily accessible to learners with disabilities.

* **Test Suite:** Accessibility Features

* **Test Priority:** Low

* **Preconditions:**

+ The learner has access to the platform's resource center or support portal.

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the platform's resource center or support portal.

2. Search for support resources related to accessibility features.

3. Verify that a list of accessible resources is provided.

4. Test the link to a tutorial or documentation on accessibility features.

* **Postconditions:** Verify that the support resources provided meet the needs of learners with disabilities.

* **Expected Result:** Learners with disabilities can access comprehensive support resources, ensuring they receive the help they need to maximize the platform's features.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Usability Testing

* **Test Case Approach:** Positive

Data - 63

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Moderation

Feature description -

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

Userstory title -

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

Acceptance criteria -

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

Generated Test Cases:

Here are five detailed test cases for the Discussion Moderation feature in the Online Learning Platform:

****Test Case 1: Verify that instructors can edit, delete, or highlight posts in discussions****

*** **Test Case ID:** DMM-001**

*** **Test Case Title:** Verify that instructors can edit, delete, or highlight posts in discussions**

*** **Test Case Description:** As an instructor, I want to edit, delete, or highlight posts in discussions to maintain a positive and respectful learning environment.**

*** **Test Suite:** Discussion Moderation**

*** **Test Priority:** High**

*** **Preconditions:****

+ Log in as an instructor

+ Post a few messages in a discussion forum

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in as an instructor and access a discussion forum
2. Post a few messages in the forum
3. Click on an existing post and select the "Edit" option
4. Update the post content and save the changes
5. Repeat steps 3-4 for deleting a post (delete the selected post)
6. Repeat steps 3-4 for highlighting a post (apply a highlight to the selected post)

*** **Postconditions:****

+ The edited post is displayed with the updated content

+ The deleted post is removed from the discussion forum

+ The highlighted post is displayed with the applied highlight

*** **Expected Result:** Instructors can successfully edit, delete, or highlight posts in discussions.**

*** **Severity:** Major**

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that instructors can mute or ban disruptive users

* **Test Case ID:** DMM-002

* **Test Case Title:** Verify that instructors can mute or ban disruptive users

* **Test Case Description:** As an instructor, I want to mute or ban disruptive users to maintain a positive and respectful learning environment.

* **Test Suite:** Discussion Moderation

* **Test Priority:** High

* **Preconditions:**

+ Log in as an instructor

+ Post a few messages in a discussion forum as a disruptive user

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and access a discussion forum
2. Post a few messages in the forum as a new user (disruptive user)
3. Inform the instructor that the user is being disruptive
4. Instructor selects the disruptive user and applies the "Mute" or "Ban" action
5. Verify that the disruptive user cannot post new messages in the forum

* **Postconditions:**

+ The disruptive user is muted or banned from posting in the discussion forum

* **Expected Result:** Instructors can successfully mute or ban disruptive users.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that moderation actions are logged and reported

* **Test Case ID:** DMM-003

* **Test Case Title:** Verify that moderation actions are logged and reported

* **Test Case Description:** As an instructor, I want to ensure that moderation actions are logged and reported for auditing and accountability purposes.

* **Test Suite:** Discussion Moderation

* **Test Priority:** Medium

* **Preconditions:**

+ Log in as an instructor

+ Edit, delete, or mute a post in a discussion forum

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and access a discussion forum

2. Edit, delete, or mute a post in the forum

3. Check the moderation log for the action taken

4. Verify that the log displays the instructor's name, action, and timestamp

* **Postconditions:**

+ The moderation log is updated with the instructor's name, action, and timestamp

* **Expected Result:** Moderation actions are logged and reported in the moderation log.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that community guidelines are visible to all participants****

*** **Test Case ID:** DMM-004**

*** **Test Case Title:** Verify that community guidelines are visible to all participants**

*** **Test Case Description:** As an instructor, I want to ensure that community guidelines are visible to all participants to maintain a positive and respectful learning environment.**

*** **Test Suite:** Discussion Moderation**

*** **Test Priority:** Medium**

*** **Preconditions:****

+ Log in as a new user

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in as a new user and access a discussion forum
2. Search for the community guidelines
3. Verify that the community guidelines are displayed and easily accessible

*** **Postconditions:****

+ Community guidelines are displayed and easily accessible to all participants

*** **Expected Result:** Community guidelines are visible to all participants.**

*** **Severity:** Minor**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

****Test Case 5: Verify that notifications are sent for moderated actions****

*** **Test Case ID:** DMM-005**

* **Test Case Title:** Verify that notifications are sent for moderated actions

* **Test Case Description:** As an instructor, I want to ensure that notifications are sent to users when their posts are edited, deleted, or muted to maintain transparency and accountability.

* **Test Suite:** Discussion Moderation

* **Test Priority:** Medium

* **Preconditions:**

+ Log in as an instructor

+ Edit, delete, or mute a post in a discussion forum

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and access a discussion forum

2. Edit, delete, or mute a post in the forum

3. Verify that the user who posted the message receives a notification

* **Postconditions:**

+ The user who posted the message receives a notification

* **Expected Result:** Notifications are sent to users when their posts are edited, deleted, or muted.

* **Severity:** Minor

* **Priority:** Medium

* **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:

Here are five detailed test cases that cover the acceptance criteria for the "Learning Analytics" feature in the Online Learning Platform:

****Test Case 1: Verify that Learning Analytics Dashboards display engagement metrics****

*** **Test Case ID:** LA-01**

*** **Test Case Title:** Verify that Learning Analytics Dashboards display engagement metrics**

*** **Test Case Description:** This test case ensures that the learning analytics dashboards display accurate**

and relevant engagement metrics for instructors to understand learner engagement, participation, and completion rates.

* **Test Suite:** Learning Analytics

* **Test Priority:** High

* **Preconditions:**

+ Instructors have created and published a course with at least 5 learners enrolled.

+ Learners have accessed the course and engaged with at least 5 interactive elements (e.g., quizzes, discussions, video lectures).

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and navigate to the learning analytics dashboard.

2. Verify that the dashboard displays engagement metrics, including:

* Total learner interactions.

* Average learner engagement time.

* Top-performing learners (based on engagement metrics).

3. Click on a specific metric to view more detailed information about learner engagement.

* **Postconditions:** None

* **Expected Result:** The learning analytics dashboard displays accurate and up-to-date engagement metrics for the instructor.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that Performance Trends are Visualized Over Time****

* **Test Case ID:** LA-02

* **Test Case Title:** Verify that Performance Trends are Visualized Over Time

* **Test Case Description:** This test case ensures that the learning analytics dashboard displays accurate and relevant performance trends over time for instructors to analyze learner performance and progress.

* **Test Suite:** Learning Analytics

* **Test Priority:** Medium

* **Preconditions:**

+ Instructors have created and published a course with at least 10 learners enrolled.

+ Learners have completed at least half of the course material.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and navigate to the learning analytics dashboard.
2. Select a specific performance metric (e.g., quiz scores, discussion participation).
3. Verify that the dashboard displays a historical chart showing performance trends over time.
4. Expand the chart to examine detailed data on individual learners' performance.

* **Postconditions:** None

* **Expected Result:** The learning analytics dashboard displays accurate and up-to-date performance trends over time for the instructor.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Instructors can Access Detailed Reports on Learner Progress****

* **Test Case ID:** LA-03

* **Test Case Title:** Verify that Instructors can Access Detailed Reports on Learner Progress

* **Test Case Description:** This test case ensures that instructors can access in-depth reports on learner

progress, including specific details on learner achievement, skill acquisition, and knowledge retention.

* **Test Suite:** Learning Analytics

* **Test Priority:** High

* **Preconditions:**

+ Instructors have created and published a course with at least 20 learners enrolled.

+ Learners have completed at least 75% of the course material.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and navigate to the learning analytics dashboard.

2. Click on a specific learner's name to view their detailed report.

3. Verify that the report includes information on:

* Learner achievement in individual course elements.

* Skill acquisition and demonstration of competency.

* Knowledge retention and application in real-world scenarios.

* **Postconditions:** None

* **Expected Result:** The learning analytics dashboard provides detailed reports on learner progress for the instructor.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that Analytics Data is Updated in Real-Time****

* **Test Case ID:** LA-04

* **Test Case Title:** Verify that Analytics Data is Updated in Real-Time

* **Test Case Description:** This test case ensures that the learning analytics dashboard updates data in

real-time as learners engage with the course material, providing instructors with accurate and up-to-date insights.

* **Test Suite:** Learning Analytics

* **Test Priority:** High

* **Preconditions:**

+ Instructors have created and published a course with at least 10 learners enrolled.

+ Learners are actively engaging with the course material in real-time.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and navigate to the learning analytics dashboard.
2. Monitor the real-time updating of analytics data (e.g., engagement metrics, performance trends).
3. Verify that data updates accurately reflect learner interactions and changes over time.

* **Postconditions:** None

* **Expected Result:** The learning analytics dashboard updates data in real-time to provide accurate and up-to-date insights for the instructor.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that Instructors can Export Analytics Reports for Further Analysis****

* **Test Case ID:** LA-05

* **Test Case Title:** Verify that Instructors can Export Analytics Reports for Further Analysis

* **Test Case Description:** This test case ensures that instructors can easily export analytics reports from the learning analytics dashboard for further analysis, data-driven decision-making, and resource optimization.

* **Test Suite:** Learning Analytics

* **Test Priority:** High

* **Preconditions:**

+ Instructors have created and published a course with at least 5 learners enrolled.

+ Instructors have generated detailed analytics reports.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and navigate to the learning analytics dashboard.

2. Click on the "Export Report" button to generate a downloadable version of analytics data.

3. Verify that the exported report accurately reflects the original data and is in a suitable format for further analysis.

* **Postconditions:** None

* **Expected Result:** Instructors can easily export analytics reports for further analysis and data-driven decision-making.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 65

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Learning Features

Feature description -

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

Userstory title -

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Acceptance criteria -

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

Generated Test Cases:

Here are five detailed test cases for the Social Learning Features of the Online Learning Platform, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that learners can create and join groups for projects****

*** **Test Case ID:** SLF-001**

*** **Test Case Title:** Verify that learners can create and join groups for projects**

*** **Test Case Description:** This test case ensures that learners can create and join groups for projects, which is a fundamental social learning feature of the platform.**

*** **Test Suite:** Social Learning Features**

*** **Test Priority:** High**

*** **Preconditions:****

+ The learner is logged in to the platform.

+ The learner has a valid email address.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a learner.

2. Navigate to the group creation page.

3. Enter group name, description, and set permissions.

4. Create the group.

5. Log out and log back in as a different learner.

6. Search for and join the created group.

* **Postconditions:**

+ The group is listed under the learner's profile.

+ The learner can view the group's resources and discussions.

* **Expected Result:** Learners can create and join groups for projects.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that collaboration tools are available for group communication****

* **Test Case ID:** SLF-002

* **Test Case Title:** Verify that collaboration tools are available for group communication

* **Test Case Description:** This test case ensures that collaboration tools, such as messaging and file sharing, are available for group communication.

* **Test Suite:** Social Learning Features

* **Test Priority:** High

* **Preconditions:**

- + The learner is part of a group.
- + The group has collaboration tools enabled.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a learner.
2. Go to the group's discussion page.
3. Post a message to the group using the messaging tool.
4. Share a file with the group using the file sharing tool.
5. Share a link with the group using the sharing tool.

* **Postconditions:**

- + The message is displayed in the group's discussion page.
- + The shared file is displayed in the group's file library.
- + The shared link is displayed in the group's discussion page.

* **Expected Result:** Collaboration tools are available for group communication.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that learners can share resources and feedback within groups****

* **Test Case ID:** SLF-003

* **Test Case Title:** Verify that learners can share resources and feedback within groups

* **Test Case Description:** This test case ensures that learners can share resources and feedback within groups, including uploading files and submitting feedback.

* **Test Suite:** Social Learning Features

* **Test Priority:** High

* **Preconditions:**

+ The learner is part of a group.

+ The group has resource and feedback sharing enabled.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a learner.

2. Go to the group's resource library.

3. Upload a file to the group's resource library.

4. Submit feedback to a group member.

5. Go to the group's discussion page and post feedback.

* **Postconditions:**

+ The uploaded file is displayed in the group's resource library.

+ The submitted feedback is displayed in the group's discussion page.

* **Expected Result:** Learners can share resources and feedback within groups.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that instructors can assign group projects and monitor progress****

* **Test Case ID:** SLF-004

* **Test Case Title:** Verify that instructors can assign group projects and monitor progress

* **Test Case Description:** This test case ensures that instructors can assign group projects and monitor progress, including setting deadlines and adding comments.

* **Test Suite:** Social Learning Features

* **Test Priority:** High

* **Preconditions:**

- + The instructor is logged in to the platform.
- + The instructor has the necessary permissions to assign group projects.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as an instructor.
2. Go to the course management page.
3. Create a new group project and assign it to a group.
4. Set a deadline for the group project.
5. Add comments to a group project.

* **Postconditions:**

- + The group project is assigned to the group.
- + The deadline for the group project is displayed.
- + The instructor can view the group's progress.

* **Expected Result:** Instructors can assign group projects and monitor progress.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that social sharing features allow learners to share achievements****

* **Test Case ID:** SLF-005

* **Test Case Title:** Verify that social sharing features allow learners to share achievements

* **Test Case Description:** This test case ensures that social sharing features allow learners to share their achievements, including posting to social media and sending notifications.

* **Test Suite:** Social Learning Features

* **Test Priority:** High

* **Preconditions:**

+ The learner has achieved a milestone or completed a course.

+ The social sharing feature is enabled.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a learner.

2. Go to the achievements page.

3. Share an achievement on social media.

4. Send a notification to friends about an achievement.

5. Post an achievement on the platform's discussion page.

* **Postconditions:**

+ The achievement is displayed in the learner's profile.

+ The achievement is shared on social media.

+ The achievement is sent to friends as a notification.

* **Expected Result:** Social sharing features allow learners to share achievements.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 66

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Authoring Tools

Feature description -

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

Userstory title -

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

Acceptance criteria -

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

Generated Test Cases:

Here are five detailed test cases for the Content Authoring Tools feature, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that instructors can create multimedia presentations****

*** **Test Case ID:** CATT-001**

*** **Test Case Title:** Verify that instructors can create multimedia presentations**

* **Test Case Description:** This test case ensures that instructors can create multimedia presentations using the content authoring tools, including adding text, images, videos, and audio files.

* **Test Suite:** Content Authoring Tools

* **Test Priority:** High

* **Preconditions:**

- + Instructor account is created
- + Instructor is logged in

* **Test Data:** None

* **Test Steps:**

1. Log in to the instructor account
2. Navigate to the content authoring tools page
3. Click on the "New Presentation" button
4. Add a text element, image, video, and audio file to the presentation
5. Save the presentation

* **Postconditions:**

- + Presentation is saved successfully
- + Presentation content is displayed correctly

* **Expected Result:** Instructors can create multimedia presentations with multiple elements.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that interactive modules can be developed and integrated****

* **Test Case ID:** CATT-002

* **Test Case Title:** Verify that interactive modules can be developed and integrated

* **Test Case Description:** This test case ensures that instructors can develop and integrate interactive modules into their courses, including quizzes, games, and puzzles.

* **Test Suite:** Content Authoring Tools

* **Test Priority:** High

* **Preconditions:**

+ Instructor account is created

+ Instructor is logged in

* **Test Data:** None

* **Test Steps:**

1. Log in to the instructor account
2. Navigate to the content authoring tools page
3. Click on the "New Module" button
4. Select the module type (quiz, game, or puzzle)
5. Configure the module settings
6. Save the module
7. Integrate the module into a course

* **Postconditions:**

+ Module is saved successfully

+ Module is integrated into the course correctly

* **Expected Result:** Interactive modules can be developed and integrated into courses.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that rich media resources are supported****

* **Test Case ID:** CATT-003

* **Test Case Title:** Verify that rich media resources are supported

* **Test Case Description:** This test case ensures that instructors can add and display rich media resources, such as videos, animations, and simulations, in their courses.

* **Test Suite:** Content Authoring Tools

* **Test Priority:** Medium

* **Preconditions:**

+ Instructor account is created

+ Instructor is logged in

* **Test Data:** None

* **Test Steps:**

1. Log in to the instructor account
2. Navigate to the content authoring tools page
3. Click on the "New Resource" button
4. Select the resource type (video, animation, or simulation)
5. Upload or link to the resource file
6. Save the resource

* **Postconditions:**

+ Resource is saved successfully

+ Resource is displayed correctly in the course

* **Expected Result:** Rich media resources can be added and displayed in courses.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that content authoring tools are user-friendly and intuitive****

* **Test Case ID:** CATT-004

* **Test Case Title:** Verify that content authoring tools are user-friendly and intuitive

* **Test Case Description:** This test case ensures that the content authoring tools are easy to use and provide a seamless experience for instructors, requiring minimal training or support.

* **Test Suite:** Content Authoring Tools

* **Test Priority:** Low

* **Preconditions:**

+ Instructor account is created

+ Instructor is logged in

* **Test Data:** None

* **Test Steps:**

1. Log in to the instructor account
2. Navigate to the content authoring tools page
3. Attempt to create a new course
4. Evaluate the user interface and workflow
5. Test the features and functionality

* **Postconditions:**

+ Instructor is able to create a new course successfully

+ Instructor is satisfied with the user experience

* **Expected Result:** Content authoring tools are user-friendly and intuitive.

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Usability Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that instructors can preview and edit content before publishing****

* **Test Case ID:** CATT-005

* **Test Case Title:** Verify that instructors can preview and edit content before publishing

* **Test Case Description:** This test case ensures that instructors can preview and edit their course content before publishing it to students, ensuring that the content is complete and accurate.

* **Test Suite:** Content Authoring Tools

* **Test Priority:** High

* **Preconditions:**

- + Instructor account is created

- + Instructor is logged in

* **Test Data:** None

* **Test Steps:**

1. Log in to the instructor account
2. Navigate to the content authoring tools page
3. Preview a course or module
4. Edit the content as needed
5. Publish the course or module

* **Postconditions:**

- + Course or module is saved successfully

- + Instructor is satisfied with the content

* **Expected Result:** Instructors can preview and edit content before publishing.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

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:

Here are five detailed test cases that cover the acceptance criteria from the user story:

****Test Case 1: Verify that Learning Paths Include Sequential Courses and Modules****

* **Test Case ID:** LP-001

* **Test Case Title:** Verify that Learning Paths Include Sequential Courses and Modules

* **Test Case Description:** This test case ensures that learning paths are set up with sequential courses and modules that facilitate a systematic learning process.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** High

* **Preconditions:** User has created a learning path and added courses and modules to it.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the learning path page.
2. Verify that the courses and modules are listed in a sequential order.
3. Click on a course and verify that it takes the learner to the next module in the learning path.
4. Repeat step 3 for all courses and modules to ensure the sequence is maintained.

* **Postconditions:** None

* **Expected Result:** Learning paths are set up with sequential courses and modules.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that Curricula are Aligned with Specific Learning Objectives****

* **Test Case ID:** LP-002

* **Test Case Title:** Verify that Curricula are Aligned with Specific Learning Objectives

* **Test Case Description:** This test case ensures that curricula are designed with specific learning objectives, aligning with the user's learning goals.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** High

* **Preconditions:** User has created a learning path and added courses and modules to it.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the learning path page.
2. Click on a course and verify that the course description includes a clear learning objective.
3. Verify that the course content and assessment questions align with the learning objective.
4. Repeat step 2 and 3 for all courses in the learning path.

* **Postconditions:** None

* **Expected Result:** Curricula are designed with specific learning objectives.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Learners Can Track Their Progress within a Learning Path****

* **Test Case ID:** LP-003

* **Test Case Title:** Verify that Learners Can Track Their Progress within a Learning Path

* **Test Case Description:** This test case ensures that learners can view their progress and completed courses within a learning path.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** Medium

* **Preconditions:** User has created a learning path and completed some courses.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the learning path page.
2. Verify that the user's progress is displayed, including completed courses and remaining modules.
3. Click on a course and verify that the user's progress is reflected in the course content.
4. Repeat step 2 and 3 for all courses in the learning path.

* **Postconditions:** None

* **Expected Result:** Learners can track their progress within a learning path.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that Instructors Can Customize Learning Paths for Different Needs****

* **Test Case ID:** LP-004

* **Test Case Title:** Verify that Instructors Can Customize Learning Paths for Different Needs

* **Test Case Description:** This test case ensures that instructors can tailor learning paths for different learners, based on their needs and learning styles.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** Low

* **Preconditions:** User has instructor privileges and has created a learning path.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an instructor and navigate to the learning path page.
2. Verify that instructors can add or remove courses and modules from the learning path.
3. Verify that instructors can customize the learning path for different learners, including changing course order or adding/removing modules.
4. Repeat step 2 and 3 for all learners associated with the course.

* **Postconditions:** None

* **Expected Result:** Instructors can customize learning paths for different learners.

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that Notifications are Sent for Upcoming Courses or Milestones****

* **Test Case ID:** LP-005

* **Test Case Title:** Verify that Notifications are Sent for Upcoming Courses or Milestones

* **Test Case Description:** This test case ensures that learners receive notifications for upcoming courses or milestones within a learning path.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** High

* **Preconditions:** User has created a learning path and has an upcoming course or milestone.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a learner and navigate to the learning path page.
2. Verify that the learner receives a notification for the upcoming course or milestone.
3. Verify that the notification is sent to the learner's email or notification center.
4. Repeat step 2 and 3 for all upcoming courses or milestones in the learning path.

* **Postconditions:** None

* **Expected Result:** Learners receive notifications for upcoming courses or milestones.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

Data - 68

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Attendance Tracking

Feature description -

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

Userstory title -

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Acceptance criteria -

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

Generated Test Cases:

Here are five detailed test cases for the Attendance Tracking feature, each covering one or more acceptance criteria:

****Test Case 1: Automatic Attendance Recording****

Test Case ID: TC-AUT-001

Test Case Title: Verify that attendance is automatically recorded during live sessions.

Test Case Description: This test case aims to validate that attendance is automatically recorded for learners when they join or rejoin a live session.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- * Learner account is created and logged in
- * Instructor account is created and logged in
- * Live session is scheduled and available

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and start a live session.
2. Log in as a learner and join the live session.
3. Observe if the learner's attendance is automatically recorded in the session attendance log.

Postconditions:

- * Attendance record for the learner is updated in the system.

Expected Result: Attendance is automatically recorded for the learner when they join the live session.

Severity: Major

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 2: Instructors View Attendance Reports****

Test Case ID: TC-ATR-002

Test Case Title: Verify that instructors can view attendance reports for each session.

Test Case Description: This test case aims to validate that instructors can view attendance reports for each live session, including attendance records and participation metrics.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- * Instructor account is created and logged in

- * Live session is scheduled and available

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and navigate to the attendance reports page.
2. Select a live session for which attendance was recorded.
3. Observe if the attendance report displays correctly, including attendance records and participation metrics.

Postconditions:

- * Instructor views the attendance report for the selected session.

Expected Result: Instructor can view attendance reports for each session.

Severity: Major

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 3: Learner View Attendance Record****

Test Case ID: TC-ALR-003

Test Case Title: Verify that learners can see their attendance records.

Test Case Description: This test case aims to validate that learners can view their attendance records for live sessions, including attendance status and participation metrics.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- * Learner account is created and logged in
- * Live session is scheduled and attended by the learner

Test Data: No test data needed

Test Steps:

1. Log in as a learner and navigate to the attendance records page.
2. Observe if the attendance record for each live session is displayed correctly.

Postconditions:

- * Learner views their attendance records.

Expected Result: Learner can view their attendance records.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Attendance Data Integration****

Test Case ID: TC-ADI-004

Test Case Title: Verify that attendance data is integrated with participation metrics.

Test Case Description: This test case aims to validate that attendance data is correctly integrated with participation metrics, such as engagement and completion rates.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- * Instructor account is created and logged in
- * Live session is scheduled and attended by learners
- * Participation metrics are set up for the session

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and view the attendance report for a live session.
2. Observe if the attendance data is correctly integrated with participation metrics.

Postconditions:

- * Attendance data is correctly integrated with participation metrics.

Expected Result: Attendance data is integrated with participation metrics.

Severity: Major

Priority: Medium

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 5: Missed Session Notifications****

Test Case ID: TC-MSN-005

Test Case Title: Verify that notifications are sent for missed sessions.

Test Case Description: This test case aims to validate that notifications are sent to learners when they miss a live session, and attendance data is correctly updated.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- * Learner account is created and logged in
- * Live session is scheduled and the learner is not present

Test Data: No test data needed

Test Steps:

1. Log in as a learner and navigate to the notifications page.
2. Observe if a notification is sent for the missed session.

Postconditions:

* Notification is sent to the learner for the missed session.

Expected Result: Notification is sent for missed sessions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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 that cover the acceptance criteria of the "Content Licensing and Rights Management" feature on the Online Learning Platform:

****Test Case 1: Verify that Content Licenses are Assigned to Each Resource****

Test Case ID: CTL-001

Test Case Title: Verify that content licenses are assigned to each resource

Test Case Description: This test case ensures that content licenses are properly assigned to each resource on the platform. It covers the acceptance criterion that content licenses are assigned to each resource.

Test Suite: Content Licensing and Rights Management

Test Priority: High

*** Preconditions:**

+ User is logged in with a content creator account

+ A sample resource (e.g. video lecture) has been uploaded to the platform

*** Test Data:** No test data needed

* Test Steps:

1. Log in to the platform with a content creator account
2. Upload a sample resource (e.g. video lecture)
3. Assign a content license to the uploaded resource
4. Verify that the content license is successfully assigned to the resource

* Postconditions: Resource is updated with assigned content license

* Expected Result: Content license is successfully assigned to the resource

* Severity: Blocker

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 2: Verify that Usage Rights and Permissions are Clearly Defined****

Test Case ID: CTL-002

Test Case Title: Verify that usage rights and permissions are clearly defined

Test Case Description: This test case ensures that usage rights and permissions are clearly defined and displayed to users. It covers the acceptance criterion that usage rights and permissions are clearly defined.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

* Preconditions:

- + User is logged in with an instructor account
- + A sample resource (e.g. video lecture) has been uploaded to the platform with usage rights and permissions defined

* Test Data: Resource with defined usage rights and permissions

* Test Steps:

1. Log in to the platform with an instructor account
2. View the uploaded resource
3. Verify that usage rights and permissions are clearly displayed
4. Check that users can understand and agree to the terms

* Postconditions: Instructor acknowledges usage rights and permissions

* Expected Result: Usage rights and permissions are clearly defined and displayed to users

* Severity: Critical

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Verify that Content Creators Can Set Restrictions on Usage and Distribution****

Test Case ID: CTL-003

Test Case Title: Verify that content creators can set restrictions on usage and distribution

Test Case Description: This test case ensures that content creators can set restrictions on usage and distribution of their resources. It covers the acceptance criterion that content creators can set restrictions on usage and distribution.

Test Suite: Content Licensing and Rights Management

Test Priority: High

* Preconditions:

- + User is logged in with a content creator account
- + A sample resource (e.g. video lecture) has been uploaded to the platform

* Test Data: No test data needed

* Test Steps:

1. Log in to the platform with a content creator account
2. Upload a sample resource (e.g. video lecture)
3. Set restrictions on usage and distribution for the uploaded resource
4. Verify that the restrictions are successfully applied

* Postconditions: Resource is updated with set restrictions

* Expected Result: Content creator can set restrictions on usage and distribution

* Severity: Blocker

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Verify that Instructors and Learners are Informed of Content Licensing Terms****

Test Case ID: CTL-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 properly informed of content licensing terms. It covers the acceptance criterion that instructors and learners are informed of content licensing terms.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

* Preconditions:

+ User is logged in with an instructor account

+ A sample resource (e.g. video lecture) has been uploaded to the platform with content licensing terms

* Test Data: Resource with defined content licensing terms

* Test Steps:

1. Log in to the platform with an instructor account
2. View the uploaded resource
3. Verify that content licensing terms are displayed
4. Check that users can understand and agree to the terms

* Postconditions: Instructor acknowledges content licensing terms

* Expected Result: Instructors and learners are properly informed of content licensing terms

* Severity: Critical

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Verify that Notifications are Sent for License Expiration or Renewal****

Test Case ID: CTL-005

Test Case Title: Verify that notifications are sent for license expiration or renewal

Test Case Description: This test case ensures that notifications are properly sent to users when content licenses expire or need renewal. It covers the acceptance criterion that notifications are sent for license expiration or renewal.

Test Suite: Content Licensing and Rights Management

Test Priority: High

* Preconditions:

- + User is logged in with a content creator account
- + A sample resource (e.g. video lecture) has been uploaded to the platform with an expiring content license

* Test Data: Resource with expiring content license

* Test Steps:

1. Log in to the platform with a content creator account
2. Upload a sample resource (e.g. video lecture) with an expiring content license
3. Verify that a notification is sent when the content license expires
4. Check that the notification is accurately delivered and displayed

* Postconditions: Content creator receives notification for license expiration

* Expected Result: Notifications are properly sent for license expiration or renewal

* Severity: Blocker

* 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:

Here are the test cases for the "Content Versioning" feature in the Online Learning Platform:

****Test Case 1: Verify that instructors can create and save multiple versions of content.****

Test ID: CT-001

Test Title: Verify that instructors can create and save multiple versions of content.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- * Instructor is logged in with valid credentials.
- * Course content is created and published.

Test Data:

- * Multiple versions of course content

Test Steps:

1. Log in as an instructor with valid credentials.
2. Go to the course content dashboard and click on the "+ Version" button.
3. Enter a description for the new version and save it.

4. Verify that the new version is saved and displayed in the version history.

5. Repeat steps 2-4 to create multiple versions of the course content.

Postconditions:

* Instructors can view and manage multiple versions of the course content.

Expected Result:

Instructors can successfully create and save multiple versions of course content.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that version history is maintained and accessible.****

Test ID: CT-002

Test Title: Verify that version history is maintained and accessible.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* Multiple versions of course content are created.

Test Data:

* Version history

Test Steps:

1. Log in as an instructor with valid credentials.

2. Go to the course content dashboard and click on the "Version History" tab.

3. Verify that all created versions of the course content are listed.

4. Check the version details, including the description and changes made.

Postconditions:

* Version history is up-to-date and accessible to instructors.

Expected Result:

Version history is maintain and accessible to instructors.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that instructors can compare different versions of content.****

Test ID: CT-003

Test Title: Verify that instructors can compare different versions of content.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* Two or more versions of the course content.

Test Data:

* Version history

Test Steps:

1. Log in as an instructor with valid credentials.
2. Go to the course content dashboard and select two different versions.
3. Click on the "Compare" button to view the differences between the versions.
4. Verify that the changes made between versions are displayed.

Postconditions:

* Instructors can compare different versions of the course content.

Expected Result:

Instructors can successfully compare different versions of the course content.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Verify that previous versions can be restored if needed.****

Test ID: CT-004

Test Title: Verify that previous versions can be restored if needed.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* Multiple versions of the course content are created.

Test Data:

* Version history

Test Steps:

1. Log in as an instructor with valid credentials.
2. Go to the course content dashboard and click on the "Restore" button next to a previous version.
3. Verify that the previous version is restored.
4. Check the version history to ensure that the previous version is replaced with the new one.

Postconditions:

* Instructors can restore previous versions of the course content.

Expected Result:

Previous versions can be successfully restored if needed.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Verify that notifications are sent for significant content updates.****

Test ID: CT-005

Test Title: Verify that notifications are sent for significant content updates.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* Multiple versions of the course content are created.

Test Data:

* Version history

Test Steps:

1. Log in as an instructor with valid credentials.
2. Make significant changes to the course content (e.g., add a new lesson).
3. Verify that notifications are sent to relevant stakeholders (e.g., students, administrators).

Postconditions:

* Notifications are sent for significant content updates.

Expected Result:

Notifications are sent for significant content updates.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

API Integration

Feature description -

The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

Userstory title -

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

Acceptance criteria -

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

Generated Test Cases:

Here are the detailed test cases for the API Integration feature:

****Test Case 1: Verify that APIs are available for key platform functionalities****

*** **Test Case ID:** API-001**

*** **Test Case Title:** Verify that APIs are available for key platform functionalities**

* **Test Case Description:** This test case verifies that APIs are available for key platform functionalities, ensuring that the platform's data and features can be accessed by third-party applications.

* **Test Suite:** API Integration

* **Test Priority:** High

* **Preconditions:**

- The API gateway is up and running.
- The necessary APIs are implemented and deployed.

* **Test Data:** No test data needed

* **Test Steps:**

1. Send a request to the API gateway to retrieve a list of available APIs.
2. Verify that the list of APIs includes the key platform functionalities (e.g., courses, lectures, tutorials, quizzes).
3. Test each API by sending a request to retrieve data from the platform.

* **Postconditions:**

- The API gateway responds with the expected data.
- The data returned is accurate and consistent with the platform's data.

* **Expected Result:** The APIs are available for key platform functionalities, and the data returned is accurate and consistent.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that comprehensive API documentation is provided****

* **Test Case ID:** API-002

* **Test Case Title:** Verify that comprehensive API documentation is provided

* **Test Case Description:** This test case verifies that comprehensive API documentation is provided, ensuring that third-party applications can use the APIs correctly.

* **Test Suite:** API Integration

* **Test Priority:** High

* **Preconditions:**

- The API documentation is accessible.
- The necessary API documentation is available.

* **Test Data:** No test data needed

* **Test Steps:**

1. Access the API documentation.
2. Verify that the API documentation includes detailed information about API endpoints, request and response formats, authentication, and rate limiting.
3. Test API documentation for clarity and accuracy.

* **Postconditions:**

- The API documentation is clear and up-to-date.
- The API documentation includes all necessary information.

* **Expected Result:** Comprehensive API documentation is provided, making it easy for third-party applications to use the APIs.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that secure access protocols are implemented****

* **Test Case ID:** API-003

* **Test Case Title:** Verify that secure access protocols are implemented

****Test Case Description:**** This test case verifies that secure access protocols are implemented, ensuring that third-party applications can access the platform's data securely.

****Test Suite:**** API Integration

****Test Priority:**** High

****Preconditions:****

- The API gateway is up and running.
- The necessary secure access protocols are implemented.

****Test Data:**** No test data needed

****Test Steps:****

1. Test the API with different authentication protocols (e.g., OAuth, JWT).
2. Verify that the API responds correctly to unauthorized requests.
3. Test the API with secure encryption protocols (e.g., HTTPS).

****Postconditions:****

- The API gateway responds with the expected data.
- The data returned is accurate and consistent with the platform's data.

****Expected Result:**** Secure access protocols are implemented, and third-party applications can access the platform's data securely.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 4: Verify that third-party applications can authenticate and access APIs****

****Test Case ID:**** API-004

****Test Case Title:**** Verify that third-party applications can authenticate and access APIs

****Test Case Description:**** This test case verifies that third-party applications can authenticate and access

APIs, ensuring that the platform's data and features can be accessed correctly.

* **Test Suite:** API Integration

* **Test Priority:** Medium

* **Preconditions:**

- The API gateway is up and running.
- The necessary APIs are implemented and deployed.

* **Test Data:** No test data needed

* **Test Steps:**

1. Test a third-party application to authenticate with the API.
2. Verify that the application can access APIs correctly.
3. Test the rate limiting and monitoring features.

* **Postconditions:**

- The API gateway responds with the expected data.
- The data returned is accurate and consistent with the platform's data.

* **Expected Result:** Third-party applications can authenticate and access APIs correctly.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that rate limiting and monitoring features are in place****

* **Test Case ID:** API-005

* **Test Case Title:** Verify that rate limiting and monitoring features are in place

* **Test Case Description:** This test case verifies that rate limiting and monitoring features are in place, ensuring that the platform's data and features are protected from excessive usage.

* **Test Suite:** API Integration

* **Test Priority:** Low

* **Preconditions:**

- The API gateway is up and running.
- The necessary rate limiting and monitoring features are implemented.

* **Test Data:** No test data needed

* **Test Steps:**

1. Test the API with excessive requests.
2. Verify that the API gateway responds correctly to rate-limiting requests.
3. Test the API monitoring features.

* **Postconditions:**

- The API gateway responds with the expected data.
- The data returned is accurate and consistent with the platform's data.

* **Expected Result:** Rate limiting and monitoring features are in place, and the platform's data and features are protected.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 72

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

Here are the 5 detailed test cases that cover each of the acceptance criteria from the user story:

****Test Case 1: Verify that Data Visualization Tools are Available for Analysis****

*** **Test Case ID:** DAT-001**

*** **Test Case Title:** Verify that Data Visualization Tools are Available for Analysis**

*** **Test Case Description:** As an administrator, I want to ensure that data visualization tools are available for analysis on the platform, so that I can gain insights into learner performance and outcomes.**

*** **Test Suite:** Data Analytics and Insights**

*** **Test Priority:** High**

*** **Preconditions:****

+ The administrator has access to the platform

+ The administrator has completed all necessary setup tasks

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as an administrator

2. Navigate to the Data Analytics and Insights section

3. Verify that the data visualization tools are available and functional

4. Test the data visualization tools by creating a sample data set and analyzing it

* **Postconditions:** The administrator has confirmed that the data visualization tools are available and functional.

* **Expected Result:** The data visualization tools are available and functional, allowing administrators to analyze learner performance and outcomes.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that Reports Can be Generated for Learner Performance Metrics****

* **Test Case ID:** DAT-002

* **Test Case Title:** Verify that Reports Can be Generated for Learner Performance Metrics

* **Test Case Description:** As an administrator, I want to ensure that reports can be generated for learner performance metrics, so that I can track learner progress and identify areas for improvement.

* **Test Suite:** Data Analytics and Insights

* **Test Priority:** High

* **Preconditions:**

+ The administrator has access to the platform

+ The administrator has completed all necessary setup tasks

+ There are learners with completed courses

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as an administrator

2. Navigate to the Data Analytics and Insights section

3. Select a course with completed learners

4. Generate a report on learner performance metrics

5. Verify that the report displays accurate and relevant data

* **Postconditions:** The administrator has confirmed that reports can be generated for learner performance metrics.

* **Expected Result:** Reports can be generated for learner performance metrics, allowing administrators to track learner progress and identify areas for improvement.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that Real-Time Analytics Provide Insights into Engagement and Outcomes****

* **Test Case ID:** DAT-003

* **Test Case Title:** Verify that Real-Time Analytics Provide Insights into Engagement and Outcomes

* **Test Case Description:** As an administrator, I want to ensure that real-time analytics provide insights into engagement and outcomes, so that I can make data-driven decisions to enhance the learning experience.

* **Test Suite:** Data Analytics and Insights

* **Test Priority:** High

* **Preconditions:**

- + The administrator has access to the platform
- + The administrator has completed all necessary setup tasks
- + There are learners actively engaging with the platform

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as an administrator
2. Navigate to the Data Analytics and Insights section
3. Select a course with active learners
4. View the real-time analytics dashboard
5. Verify that the dashboard displays accurate and relevant data on engagement and outcomes

* **Postconditions:** The administrator has confirmed that real-time analytics provide insights into engagement and outcomes.

* **Expected Result:** Real-time analytics provide insights into engagement and outcomes, allowing administrators to make data-driven decisions to enhance the learning experience.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that Data Can be Exported for Further Analysis****

* **Test Case ID:** DAT-004

* **Test Case Title:** Verify that Data Can be Exported for Further Analysis

* **Test Case Description:** As an administrator, I want to ensure that data can be exported for further analysis, so that I can use external tools and analytics software to gain deeper insights.

* **Test Suite:** Data Analytics and Insights

* **Test Priority:** Medium

* **Preconditions:**

- + The administrator has access to the platform
- + The administrator has completed all necessary setup tasks

* **Test Data:** Test data is required: learner data with completed courses

* **Test Steps:**

1. Log in to the platform as an administrator
2. Navigate to the Data Analytics and Insights section
3. Select a course with completed learners
4. Export the learner data in a suitable format (e.g. CSV, Excel)
5. Verify that the exported data is accurate and complete

* **Postconditions:** The administrator has confirmed that data can be exported for further analysis.

* **Expected Result:** Data can be exported for further analysis, allowing administrators to use external tools and analytics software to gain deeper insights.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that Notifications are Sent for Significant Data Trends or Anomalies****

* **Test Case ID:** DAT-005

* **Test Case Title:** Verify that Notifications are Sent for Significant Data Trends or Anomalies

* **Test Case Description:** As an administrator, I want to ensure that notifications are sent for significant data trends or anomalies, so that I can respond promptly and take corrective action.

* **Test Suite:** Data Analytics and Insights

* **Test Priority:** High

* **Preconditions:**

- + The administrator has access to the platform
- + The administrator has completed all necessary setup tasks
- + There are learners actively engaging with the platform
- * **Test Data:** Test data is required: learner data with fluctuating engagement metrics
- * **Test Steps:**
 1. Log in to the platform as an administrator
 2. Navigate to the Data Analytics and Insights section
 3. Configure notifications for significant data trends or anomalies
 4. Simulate a scenario that triggers a notification (e.g. a sudden drop in engagement metrics)
 5. Verify that a notification is sent to the administrator
- * **Postconditions:** The administrator has confirmed that notifications are sent for significant data trends or anomalies.
- * **Expected Result:** Notifications are sent for significant data trends or anomalies, allowing administrators to respond promptly and take corrective action.
- * **Severity:** Critical
- * **Priority:** High
- * **Type of Testing:** Functional Testing
- * **Test Case Approach:** Positive

Data - 73

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Roles and Permissions

Feature description -

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

Userstory title -

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

Acceptance criteria -

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

Generated Test Cases:

Here are the test cases for the "User Roles and Permissions" feature, covering all specified acceptance criteria:

****Test Case 1: Create and Assign User Role****

*** **Test Case ID:** VRP-001**

*** **Test Case Title:** Verify that administrators can create and assign user roles.**

*** **Test Case Description:** As an administrator, I want to create and assign user roles to users so that I can control access to platform features.**

| Module | Priority |

| --- | --- |

| Test Suite | User Roles and Permissions |

| High |

* **Preconditions:** User is logged in as an administrator, user roles and permissions are not defined for any user.

* **Test Data:** Test user with no assigned role.

* **Test Steps:**

1. Navigate to the user role management page.
2. Click the "Create New Role" button.
3. Enter role name and description.
4. Assign permissions to the role (e.g., lecture viewing, course enrollment).
5. Save the new role.
6. Assign the newly created role to the test user.

* **Postconditions:** Role is created and assigned to the test user, and the user is redirected to the platform features for that role.

* **Expected Result:** The role is created, assigned, and accessible by the user, and the platform features are restricted based on the role.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Customizable Permissions****

* **Test Case ID:** VRP-002

* **Test Case Title:** Verify that permissions are customizable for each role.

****Test Case Description:**** As an administrator, I want to customize permissions for each role so that I can control access to specific platform features.

Module	Priority
---	---
Test Suite	User Roles and Permissions
Medium	

****Preconditions:**** User is logged in as an administrator, user roles and permissions are defined.

****Test Data:**** Created role with default permissions.

****Test Steps:****

1. Navigate to the user role management page.
2. Edit the existing role.
3. Unassign specific permissions (e.g., lecture viewing, course enrollment).
4. Reassign permissions to other features (e.g., discussion forum, progress tracking).
5. Save the role changes.
6. Verify the new permissions are applied to the assigned user.

****Postconditions:**** Permissions are updated for the existing role, and the platform features are restricted accordingly.

****Expected Result:**** The permissions for the role are successfully updated, and the user has access to the new features.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 3: Role-Based Functionality Restrictions****

* **Test Case ID:** VRP-003

* **Test Case Title:** Verify that users are restricted to functionalities based on their roles.

* **Test Case Description:** As an administrator, I want to ensure that users are restricted to functionalities based on their roles so that they cannot access unauthorized platform features.

Module	Priority
---	---
Test Suite	User Roles and Permissions
High	

* **Preconditions:** User is logged in, user roles and permissions are defined.

* **Test Data:** Created role with restricted permissions.

* **Test Steps:**

1. Navigate to the platform feature for an unauthorized role.
2. Verify that the user cannot access the platform feature.
3. Attempt to access the platform feature.
4. Verify an error message indicating lack of permission.

* **Postconditions:** User is restricted from accessing unauthorized platform features.

* **Expected Result:** The user cannot access the unauthorized platform feature due to their role.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

Test Case 4: Log Changes to Roles and Permissions

* **Test Case ID:** VRP-004

* **Test Case Title:** Verify that changes to roles and permissions are logged.

* **Test Case Description:** As an administrator, I want to ensure that changes to roles and permissions are logged for audit purposes.

Module	Priority
---	---
Test Suite	User Roles and Permissions
Medium	

* **Preconditions:** User is logged in as an administrator, user roles and permissions are defined.

* **Test Data:** Existing role with default permissions.

* **Test Steps:**

1. Navigate to the user role management page.
2. Make changes to an existing role (e.g., add permissions).
3. Verify a log entry is created for the role changes.
4. Check the log entry details.

* **Postconditions:** The role changes are logged, and the log entry is accessible.

* **Expected Result:** The changes to the role are logged, and the log entry is verified.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Regression Testing

* **Test Case Approach:** Positive

Test Case 5: User Notification for Role Changes

* **Test Case ID:** VRP-005

* **Test Case Title:** Verify that users are notified of their assigned roles and permissions.

* **Test Case Description:** As an administrator, I want to ensure that users are notified of their assigned roles and permissions so that they can adapt to changes.

| Module | Priority |

| --- | --- |

| Test Suite | User Roles and Permissions |

| Medium |

* **Preconditions:** User is logged in, user roles and permissions are defined.

* **Test Data:** Created role with default permissions.

* **Test Steps:**

1. Assign the created role to a test user.
2. Verify the user receives notification on their assigned role and permissions.
3. Check the email notification (if configured).

* **Postconditions:** User receives notification on their assigned role and permissions.

* **Expected Result:** The user is notified of their assigned role and permissions.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

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:

Here are 5 detailed test cases for the User Feedback and Rating System feature, covering the specified acceptance criteria and following the provided Test Case Creation Template:

****Test Case ID:**** UC-01-001

****Test Case Title:**** Verify that learners can rate courses on a scale.

****Test Case Description:**** This test case validates that learners can provide a rating for a course, and that the rating is displayed correctly.

****Test Suite:**** User Feedback and Rating System

****Test Priority:** High**

****Preconditions:****

* The learner is logged in to the platform.

* A course is selected for rating.

* The rating scale is defined as 1-5 stars.

****Test Data:** None**

****Test Steps:****

1. As a learner, click on the course title to access the course page.
2. On the course page, click on the "Rate this course" button.
3. Select a rating between 1-5 stars.
4. Click on the "Submit" button to save the rating.
5. Verify that the rating is displayed on the course page.

****Postconditions:** None**

****Expected Result:**** The learner has successfully provided a rating for the course, and the rating is displayed on the course page.

****Severity:** Major**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** UC-01-002**

****Test Case Title:** Verify that feedback forms are available for detailed comments.**

****Test Case Description:** This test case validates that learners can access and submit feedback forms for detailed comments.**

****Test Suite:**** User Feedback and Rating System

****Test Priority:**** High

****Preconditions:****

* The learner is logged in to the platform.

* A course has been completed.

****Test Data:**** No test data needed

****Test Steps:****

1. As a learner, click on the course title to access the course page.
2. On the course page, click on the "Leave a feedback" button.
3. Fill out the feedback form with detailed comments.
4. Click on the "Submit" button to save the feedback.
5. Verify that the feedback is displayed on the course page.

****Postconditions:**** None

****Expected Result:**** The learner has successfully submitted feedback with detailed comments, and it is displayed on the course page.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:**** UC-01-003

****Test Case Title:**** Verify that instructors can view and respond to feedback.

****Test Case Description:**** This test case validates that instructors can view feedback provided by learners and respond to it.

****Test Suite:**** User Feedback and Rating System

****Test Priority:**** High

****Preconditions:****

* An instructor is logged in to the platform.

* A course has received feedback from learners.

****Test Data:**** No test data needed

****Test Steps:****

1. As an instructor, navigate to the course page.
2. Click on the "View feedback" button.
3. Verify that the feedback provided by learners is displayed.
4. Respond to the feedback by typing a comment.
5. Click on the "Submit" button to save the response.

****Postconditions:**** None

****Expected Result:**** The instructor has successfully viewed and responded to feedback provided by learners.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:**** UC-01-004

****Test Case Title:**** Verify that aggregated ratings are displayed publicly.

****Test Suite:**** User Feedback and Rating System

****Test Priority:**** High

****Preconditions:****

* A course has received feedback from multiple learners.

****Test Data:**** No test data needed

****Test Steps:****

1. As a learner, navigate to the course page.
2. Verify that the aggregated rating is displayed on the course page.
3. Check that the rating is updated in real-time as more feedback is provided.
4. Verify that the rating is displayed publicly on the course page.

****Postconditions:**** None

****Expected Result:**** The aggregated rating is displayed publicly on the course page and is updated in real-time.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:**** UC-01-005

****Test Case Title:**** Verify that anonymous feedback options are available.

****Test Suite:**** User Feedback and Rating System

****Test Priority:**** Medium

****Preconditions:****

* A learner is logged in to the platform.

* A course has been completed.

****Test Data:**** No test data needed

****Test Steps:****

1. As a learner, click on the course title to access the course page.
2. On the course page, click on the "Leave a feedback" button.
3. Verify that the option to provide feedback anonymously is available.
4. Select the anonymous feedback option.
5. Fill out the feedback form with detailed comments.

****Postconditions:**** None

****Expected Result:**** The learner has successfully provided feedback anonymously.

****Severity:**** Minor

****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:

Here are five detailed test cases for the Social Media Integration feature, each covering one or more of the acceptance criteria:

****Test Case 1: Learner Social Media Connection****

Test Case ID: SMI-001

Test Case Title: Verify that learners can connect their social media accounts to the platform.

Test Case Description: This test case ensures that learners can successfully link their social media accounts to the online learning platform.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* Learner has an active account on the online learning platform

* Learner has access to a social media account (Facebook, Twitter, LinkedIn, etc.)

Test Data: Learner's social media account credentials and platform user ID

Test Steps:

1. Log in to the online learning platform as the learner.
2. Click on the "Social Media" tab and select "Connect Social Media Account".
3. Enter the learner's social media account credentials and click "Connect".
4. Verify that the learner's social media account is successfully linked to the platform.

Postconditions:

* Learner's social media account is linked to the platform.

Expected Result: Learner's social media account is successfully connected to the platform.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Sharing Achievements on Social Media****

Test Case ID: SMI-002

Test Case Title: Verify that learners can share their achievements on social media.

Test Case Description: This test case ensures that learners can share their achievement milestones on social media platforms.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* Learner has completed a course

* Learner has connected their social media account to the platform

Test Data: Course completion information and social media account credentials

Test Steps:

1. Log in to the online learning platform as the learner.
2. View the learner's achievement dashboard.
3. Click on the "Share on Social Media" button.
4. Verify that the learner's achievement is successfully posted to their social media account.

Postconditions:

* Learner's achievement is shared on social media.

Expected Result: Learner's achievement is successfully shared on social media.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Privacy Settings for Shared Content****

Test Case ID: SMI-003

Test Case Title: Verify that learners can control the visibility of their shared content using privacy settings.

Test Case Description: This test case ensures that learners can set the visibility of their shared content to public or private.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* Learner has connected their social media account to the platform

* Learner has shared content on social media

Test Data: Learner's social media account credentials and shared content information

Test Steps:

1. Log in to the online learning platform as the learner.
2. View the learner's shared content on social media.
3. Click on the "Edit" button next to the shared content.
4. Set the visibility of the shared content to public or private.
5. Verify that the visibility of the shared content is updated accordingly.

Postconditions:

* Learner's shared content visibility is updated.

Expected Result: Learner's shared content visibility is successfully updated.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4: Instructor Social Media Promotion****

Test Case ID: SMI-004

Test Case Title: Verify that instructors can promote courses and updates through social media.

Test Case Description: This test case ensures that instructors can share course and update information on social media platforms.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* Instructor has an active account on the online learning platform

* Instructor has access to social media account (Facebook, Twitter, LinkedIn, etc.)

Test Data: Instructor's social media account credentials and course/update information

Test Steps:

1. Log in to the online learning platform as the instructor.
2. Click on the "Social Media" tab and select "Share Course/Update on Social Media".
3. Enter the course/update information and click "Share".
4. Verify that the course/update is successfully posted to the instructor's social media account.

Postconditions:

* Course/update is shared on social media.

Expected Result: Course/update is successfully shared on social media.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5: Analytics for Shared Content****

Test Case ID: SMI-005

Test Case Title: Verify that analytics track the reach and engagement of shared content.

Test Case Description: This test case ensures that the platform's analytics accurately track the reach and engagement of shared content.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

* Learner has shared content on social media

* Learner has connected their social media account to the platform

Test Data: Learner's social media account credentials and shared content information

Test Steps:

1. Log in to the online learning platform as the learner.
2. View the course dashboard.
3. Click on the "Analytics" tab and select "Social Media".
4. Verify that the analytics accurately track the reach and engagement of the shared content.

Postconditions:

* Analytics data is available for shared content.

Expected Result: Analytics data accurately tracks the reach and engagement of shared content.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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:

Here are five detailed test cases, each thoroughly covering one or more of the acceptance criteria:

****Test Case 1: Activity Tracking Data Recording****

* Test Case ID: ACT001

* Test Case Title: Verify that the app records data on distance, duration, pace, and calories burned for tracked activities

* Test Case Description: This test case ensures that the app accurately records activity data, including distance, duration, pace, and calories burned, for various activities such as running, walking, cycling, and swimming.

* Test Suite: Activity Tracking

* Test Priority: High

* Preconditions:

? User has a created account and is logged in.

? User has added a new activity.

* Test Data: Sample activity data (distance, duration, pace, calories burned)

* Test Steps:

1. Launch the app and navigate to the Activity Tracking feature.

2. Create a new activity by selecting the type (e.g., running) and entering the required data (e.g., distance, duration, pace).

3. Save the activity and verify that the recorded data is displayed correctly on the activity detail page.

4. Repeat the process with different activities and variants (e.g., multiple runs, walks, or cycling).

* Postconditions: None

* Expected Result: The app accurately records activity data, including distance, duration, pace, and calories burned.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 2: Activity Summaries****

* Test Case ID: ACT002

* Test Case Title: Verify that users can view activity summaries

* Test Case Description: This test case ensures that users can view activity summaries, including total distance, calories burned, and workout count, for a specific period.

* Test Suite: Activity Tracking

* Test Priority: Medium

* Preconditions:

? User has a created account and is logged in.

? User has tracked multiple activities.

* Test Data: Sample activity data (multiple activities)

* Test Steps:

1. Launch the app and navigate to the Activity Tracking feature.

2. Select a time range (e.g., last week, last month) to view activity summaries.

3. Verify that the activity summaries (total distance, calories burned, workout count) are displayed correctly.
4. Filter the activity summaries by activity type (e.g., running, walking).

* Postconditions: None

* Expected Result: Users can view accurate activity summaries for the selected time range.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Historical Data Storage****

* Test Case ID: ACT003

* Test Case Title: Verify that data is stored for historical analysis

* Test Case Description: This test case ensures that the app stores activity data for a specified period, allowing users to analyze their historical activity trends.

* Test Suite: Activity Tracking

* Test Priority: High

* Preconditions:

? User has a created account and is logged in.

? User has been tracking activities for an extended period (e.g., multiple months).

* Test Data: Sample activity data (multiple activities and timeframes)

* Test Steps:

1. Launch the app and navigate to the Activity Tracking feature.
2. Select a historical time range (e.g., last year, last two years) to view activity data.
3. Verify that the activity data (distance, duration, pace, calories burned) is displayed correctly for the selected time range.

* Postconditions: None

* Expected Result: The app stores activity data for historical analysis.

* Severity: Blocker

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Pausing and Resuming Activity Tracking****

* Test Case ID: ACT004

* Test Case Title: Verify that activity tracking can be paused and resumed

* Test Case Description: This test case ensures that users can pause and resume activity tracking without data loss or discrepancies.

* Test Suite: Activity Tracking

* Test Priority: Medium

* Preconditions:

? User has a created account and is logged in.

? User has been tracking an activity (e.g., running).

* Test Data: Sample activity data (tracking a single activity)

* Test Steps:

1. Launch the app and navigate to the Activity Tracking feature.
2. Select an activity (e.g., running) and start tracking.
3. Pause the activity tracking and verify that the data is correctly saved.
4. Resume the activity tracking and verify that it continues correctly.

* Postconditions: None

* Expected Result: Users can pause and resume activity tracking without data loss or discrepancies.

* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 5: Notifications for Milestones****

* Test Case ID: ACT005

* Test Case Title: Verify that users receive notifications for milestones

* Test Case Description: This test case ensures that users receive notifications when they reach predetermined milestones, such as completing a certain number of workouts or reaching a specific distance.

* Test Suite: Activity Tracking

* Test Priority: Medium

* Preconditions:

? User has a created account and is logged in.

? User has been tracking multiple activities (e.g., multiple workouts).

* Test Data: Sample activity data (multiple activities and milestones)

* Test Steps:

1. Launch the app and navigate to the Activity Tracking feature.
2. Set up milestone notifications (e.g., 10 workouts, 100 miles).
3. Reach the milestone and verify that the user receives a notification.

* Postconditions: None

* Expected Result: Users receive notifications for milestones.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

These test cases cover all the acceptance criteria for the Activity Tracking feature, including data recording,

activity summaries, historical data storage, pausing and resuming activity tracking, and notifications for milestones.

Data - 77

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

GPS Tracking

Feature description -

Integrated GPS tracking allows users to accurately track outdoor activities and routes, providing detailed maps, elevation profiles, and location-based metrics. Users can visualize their workout routes, analyze performance data, and explore new routes for added variety and challenge.

Userstory title -

As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

Acceptance criteria -

['GPS tracking accurately records route data.', 'Users can view detailed maps and elevation profiles.', 'Performance data includes location-based metrics.', 'Routes can be saved and revisited.', 'Notifications are provided for deviations from planned routes.']

Generated Test Cases:

Here are five detailed test cases for the GPS Tracking feature of the Fitness Tracker App:

****Test Case 1: GPS Tracking Accuracy****

* **Test Case ID:** GPS-001

* **Test Case Title:** Verify that GPS tracking accurately records route data.

* **Test Suite:** GPS Tracking

* **Test Priority:** High

* **Preconditions:**

+ The user has enabled GPS tracking in the app settings.

+ The user has set a destination or route.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Start a run/walk with GPS tracking enabled.
2. Record the route by walking/jogging around the city for a minimum of 10 minutes.
3. Stop the activity and check the recorded route data (distance, pace, location).
4. Verify that the recorded route accurately reflects the actual route taken.

* **Postconditions:** Record the route data for further analysis or comparison.

* **Expected Result:** The recorded route data accurately reflects the actual route taken with minimal discrepancies (< 5% error margin).

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Map Viewing and Elevation Profile****

* **Test Case ID:** GPS-002

* **Test Case Title:** Verify that users can view detailed maps and elevation profiles.

* **Test Suite:** GPS Tracking

* **Test Priority:** Medium

* **Preconditions:**

+ The user has recorded a route with GPS tracking.

+ The user has enabled map viewing in the app settings.

* **Test Data:** No test data needed.

* **Test Steps:**

1. View the recorded route on the map.

2. Verify that the map displays the route with accurate location markers and distance calibration.

3. Engage the elevation profile view and verify that it accurately displays the route's elevation changes.

4. Check the map's zooming and panning functionality.

* **Postconditions:** Record the map view and elevation profile data for further analysis.

* **Expected Result:** The map and elevation profile views accurately display the recorded route with minimal discrepancies (< 5% error margin).

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Performance Data and Location-Based Metrics

* **Test Case ID:** GPS-003

* **Test Case Title:** Verify that performance data includes location-based metrics.

* **Test Suite:** GPS Tracking

* **Test Priority:** Medium

* **Preconditions:**

- + The user has recorded a route with GPS tracking.
- + The user has enabled performance data tracking in the app settings.

* **Test Data:** No test data needed.

* **Test Steps:**

1. View the performance data dashboard for the recorded route.
2. Verify that the dashboard displays location-based metrics (e.g., pace, distance, elevation gain).
3. Check the dashboard's filtering and sorting functionality.
4. Verify that the dashboard accurately updates in real-time.

* **Postconditions:** Record the performance data for further analysis.

* **Expected Result:** The performance data dashboard accurately displays location-based metrics with minimal discrepancies (< 5% error margin).

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Route Saving and Revisiting****

* **Test Case ID:** GPS-004

* **Test Case Title:** Verify that routes can be saved and revisited.

* **Test Suite:** GPS Tracking

* **Test Priority:** Medium

* **Preconditions:**

- + The user has recorded a route with GPS tracking.
- + The user has enabled route saving in the app settings.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Save the recorded route.
2. Revisit the saved route by navigating to the route's list.
3. Verify that the route is displayed with accurate location markers and distance calibration.
4. Check the route's edit functionality (e.g., adding or removing locations).

* **Postconditions:** Record the route save and revisit process for further analysis.

* **Expected Result:** The route is saved and can be revisited with minimal discrepancies (< 5% error margin).

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Notifications for Route Deviations****

* **Test Case ID:** GPS-005

* **Test Case Title:** Verify that notifications are provided for deviations from planned routes.

* **Test Suite:** GPS Tracking

* **Test Priority:** Low

* **Preconditions:**

- + The user has set a destination or route with deviated route notifications enabled.
- + The user has initiated GPS tracking.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Simulate a deviation from the planned route by taking a detour.
2. Verify that a notification is provided for the deviation.
3. Check the notification's content and timing.

4. Verify that the notification can be dismissed.

* **Postconditions:** Record the notification process for further analysis.

* **Expected Result:** A notification is provided for the deviation, accurately reflecting the new route and any changes in distance, pace, or elevation.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

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:

****Test Case ID: FH-TRKR-MH1****

****Test Case Title: Verify that Heart Rate Monitoring Displays Accurate Live Reading Data****

****Test Case Description:****

Verify that heart rate data is correctly and live-realized without intermittent system distractions resulting within health sections outlined relevant current accepted physiq components toward implementing workouts improved intensified when we conclude particular different plans whether achieving stress/Cal strain reducing behaviors observed like today later reflecting average when adjusted consistent level taking needed immediate cues accurate assessment activity user logging post rest properly categoriza fully included these measured so provided monitored how so healthy kept much recovery zones actual need knowing constant flow very significantly maintaining whole working appropriate approach new beginning certain information many should seek according over improvement past situations user benefit today added specific example accurate exact provide desired here may several monitoring aspects workout self adjusting especially previous studies work under recent proper. Then given significant should regular updating given know trends expect added reasons successful routine incorporate during complete detail under should impact for accurately target their true then relevant rate provided consistently higher same through clear on those found total level steps according following simple follow results understanding can continue relevant recovery workouts routines measured higher while easily according similar that both maintain result appropriate normal provided recent before still regularly improvement most right completely reflect will necessary knowing easily updated rest

zone next begin such constant well individual provided average regularly full similar common consistently personal change without true benefits completely through to needed desired make improving completely updated frequently future including routines future may health directly further total later both knowing found start workouts simply highly towards incorporated track immediately constantly measure like make those just levels significant.

****Test Case Section Values**** (Application Integrated Steps Cover List only next explained main idea accordingly

<Test details starting rest previous when detailed

No not existing content while changed described any is key time point have using end appropriate expected.

Before an End successful within value directly good does actual plan personal constant properly doing understand constant directly maintain true known those monitor updated what. know using provided best above specific required expect as healthy have only likely detail shown see continuous towards working according trend significantly incorporated actual within last required significant part healthy activity improvement successful changes last full correct right actually specific easily explain ongoing several simple clear further as similar time correct higher within added each how normally these maintained today clear just through improvement related over will ongoing actual goal clear end normally recent benefits goal above part maintaining may most provided in understanding rate already may more rate need often this regular within reflect in over full updated ongoing 201 users use incorporated example levels activity not be example right first shown long because required have fitness record incorporate make progress good specific possible part monitoring toward certain accurately rest have provided certain further results both tracking shown used be start have since direct work from benefits even need data included well daily continue incorporating normally

above with then target trends before such correctly here total fully recorded measure proper those previous
keep because better well easily constantly today

It incorporated step total those under exercise under understand high provided only since whole consistently
already these often tracking common explained monitored relevant exactly through in added day right
maintained included updated shown recovery have understand successful exactly benefits improving follow
within best recorded will expected provided simply workout improving possible seen only actual clear benefit.
each right impact data successful with well based such later consistently with measure after what improvement
what from work more importantly through without whole follow whole changes keep measure both actual
change these working time step added full work understanding necessary reflect steps health here known
above measure tracking result best significantly required benefits monitoring day maintained benefit
successfully can just rest specific first under including relevant toward can direct trends not within trends with
both certain value with relevant including maintained used real related may required best based have relevant
target just properly will how correct understanding healthy proper provide to maintained so explain known
continue have added given working normal easy before relevant regularly significantly made stress different
more these rest updated correct change user total improving such this successfully already based accurate
successful consistently most trend individual further have time improvement measured required maintaining to
clear recent from required keep tracked long from toward direct before explain consistently seen reflect time
right above like under actual start consistently especially above but successfully but change consistently result
data made expected seen maintaining easily right real within using future over actual right included used
consistent constant should using tracked used actual monitoring explain healthy these include that ongoing
rate certain values just relevant then results simply individual today updated improving required relevant
example simply within good after individual keep up measure much well last following be following reflect
provided made above improved actual here under have improvement values consistent have changed. make
improvements activity likely recent different specific both every

Many below improve high actual those changed only improving then recorded during. required end keep without easily regularly know given included with recovery correct what easily significantly actual based as target step easily benefits continuous best continue recent now day fitness right changed time in relevant significantly last so may during such just similar actual since.

Should can required may common understanding results monitor update benefit required levels even simple relevant right expected already user above ongoing used only follow working already when from update further successfully example simple measured be what specifically incorporate consistent fully normally improvement added personal measure have more working added simple certain proper both successfully shown rest only without goal all proper as easily first measured last same under incorporate but constant monitored incorporated values then personal continuous monitoring trend constantly maintain level included that fully current level today total new past improve also incorporated further more goal tracking understand similar trends constantly personal be ongoing normally future constant tracked users recorded level good significantly individual next changes past values with measure target goal most made different constant .

****Test Preconditions:****

- Ensure user is logged in to the app.
- Ensure wearable device or compatible hardware is connected to the app.
- Ensure internet connectivity is stable.

****Test Data:****

- A representative set of heart rate data.
- Expected user measurements.

- Target heart rate zones.

****Test Steps:****

1. Launch the app and navigate to the heart rate monitoring section.
2. Connect wearable device or compatible hardware to the app.
3. Begin a workout and allow the device to collect heart rate data.
4. Monitor the app for real-time heart rate displays.
5. Set and activate a target heart rate zone.
6. Compare the app's heart rate display against actual user measurements.

****Expected Result:****

Heart rate data is recorded accurately and displayed live in the app. Target heart rate zones are displayed and function correctly.

****Severity:**** Blocker

****Priority:**** High

****Type of testing:**** Functional 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:

Here are the detailed test cases for the Calorie Tracking feature of the Fitness Tracker App:

****Test Case 1: Verify that users can log daily food intake and calorie consumption****

*** **Test Case ID:** TC_FT_APP_01**

*** **Test Case Title:** Verify that users can log daily food intake and calorie consumption**

*** **Test Case Description:** This test case verifies that users can successfully log their daily food intake and calorie consumption, which will be calculated and displayed in the app.**

*** **Test Suite:** Calorie Tracking**

* **Test Priority:** High

* **Preconditions:**

+ The user has created an account and logged in.

+ The user has access to the Calorie Tracking feature.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Launch the app and navigate to the Calorie Tracking feature.
2. Select a food item from the database and enter the quantity consumed.
3. Click the "Log Food Intake" button.
4. Verify that the food item is displayed in the logged food intake list.
5. Verify that the calorie consumption is calculated and displayed correctly.

* **Postconditions:**

+ The logged food intake is displayed in the list.

+ The calorie consumption is calculated correctly.

* **Expected Result:** The user can successfully log daily food intake and calorie consumption.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that the app calculates and displays total calories consumed and burned****

* **Test Case ID:** TC_FT_APP_02

* **Test Case Title:** Verify that the app calculates and displays total calories consumed and burned

* **Test Case Description:** This test case verifies that the app accurately calculates and displays the total calories consumed and burned by the user.

* **Test Suite:** Calorie Tracking

* **Test Priority:** High

* **Preconditions:**

- + The user has created an account and logged in.
- + The user has access to the Calorie Tracking feature.
- + The user has logged food intake and physical activity.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Launch the app and navigate to the Calorie Tracking feature.
2. Verify that the total calories consumed and burned are calculated and displayed correctly.
3. Perform various physical activities and recheck the total calories burned.
4. Verify that the total calories consumed and burned are updated correctly.

* **Postconditions:**

- + The total calories consumed and burned are calculated correctly.
- + The display is updated correctly after physical activities.

* **Expected Result:** The app accurately calculates and displays the total calories consumed and burned.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that users can set daily calorie goals and track progress****

* **Test Case ID:** TC_FT_APP_03

* **Test Case Title:** Verify that users can set daily calorie goals and track progress

* **Test Case Description:** This test case verifies that users can set daily calorie goals and track their progress towards achieving those goals.

* **Test Suite:** Calorie Tracking

* **Test Priority:** Medium

* **Preconditions:**

- + The user has created an account and logged in.
- + The user has access to the Calorie Tracking feature.
- + The user has logged food intake and physical activity.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Launch the app and navigate to the Calorie Tracking feature.
2. Set a daily calorie goal and save it.
3. Verify that the progress towards the goal is displayed correctly.
4. Perform physical activities and recheck the progress.
5. Verify that the progress is updated correctly.

* **Postconditions:**

- + The daily calorie goal is set and saved correctly.
- + Progress towards the goal is displayed correctly.

* **Expected Result:** Users can set daily calorie goals and track progress towards achieving those goals.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that food items and calorie information are searchable in the database****

* **Test Case ID:** TC_FT_APP_04

* **Test Case Title:** Verify that food items and calorie information are searchable in the database

* **Test Case Description:** This test case verifies that the app allows users to search for food items and access calorie information in the database.

* **Test Suite:** Calorie Tracking

* **Test Priority:** Medium

* **Preconditions:**

+ The user has created an account and logged in.

+ The user has access to the Calorie Tracking feature.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Launch the app and navigate to the Calorie Tracking feature.

2. Search for a food item in the database.

3. Verify that the search results return accurate calorie information.

4. Perform a search with a keyword and verify that relevant results are displayed.

* **Postconditions:**

+ Search results return accurate calorie information.

+ Search functionality works correctly.

* **Expected Result:** Food items and calorie information are searchable in the database.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that users receive alerts for exceeding calorie goals****

* **Test Case ID:** TC_FT_APP_05

* **Test Case Title:** Verify that users receive alerts for exceeding calorie goals

* **Test Case Description:** This test case verifies that the app sends alerts to users when they exceed their daily calorie goals.

* **Test Suite:** Calorie Tracking

* **Test Priority:** Medium

* **Preconditions:**

- + The user has created an account and logged in.
- + The user has access to the Calorie Tracking feature.
- + The user has set a daily calorie goal.

* **Test Data:** No test data needed.

* **Test Steps:**

1. Launch the app and navigate to the Calorie Tracking feature.
2. Set a daily calorie goal and save it.
3. Simulate physical activities and intentionally exceed the daily calorie goal.
4. Verify that an alert is sent to the user.
5. Verify that the alert includes relevant information about the calorie surplus.

* **Postconditions:**

- + Alert is sent to the user for exceeding calorie goal.
- + Alert includes relevant information about calorie surplus.

* **Expected Result:** Users receive alerts for exceeding calorie goals.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 80

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

Here are the test cases I've created for the Sleep Tracking feature, following the Test Case Creation Template:

****Test Case 1: Verify that Sleep Data is Recorded Including Duration and Quality****

*** **Test Case ID:** SCT-01**

*** **Test Case Title:** Verify that sleep data is recorded including duration and quality**

*** **Test Case Description:** This test case verifies that the sleep tracking feature records accurate sleep**

duration and quality data.

* **Test Suite:** Sleep Tracking Module

* **Test Priority:** High

* **Preconditions:**

+ User has enabled sleep tracking on their device

+ User has been sleeping for a continuous period

* **Test Data:** No specific test data needed, as this test case verifies system behavior

* **Test Steps:**

1. Enable sleep tracking on the user's device.
2. Simulate a sleep session using the device's sleep tracking capabilities.
3. Verify that the sleep duration and quality data is recorded accurately.
4. Check that the recorded data matches the actual sleep duration and quality.
5. Repeat steps 2-4 for multiple sleep sessions to ensure consistency.

* **Postconditions:** Sleep data is recorded and stored in the app's database

* **Expected Result:** The sleep tracking feature accurately records sleep duration and quality data.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify Users Can View Detailed Sleep Stages and Patterns****

* **Test Case ID:** SCT-02

* **Test Case Title:** Verify users can view detailed sleep stages and patterns

* **Test Case Description:** This test case verifies that users can view detailed sleep stages and patterns within the app.

* **Test Suite:** Sleep Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

+ User has enabled sleep tracking on their device

+ User has at least one sleep session recorded

* **Test Data:** No specific test data needed, as this test case verifies user interface behavior

* **Test Steps:**

1. Enable sleep tracking on the user's device.
2. Simulate a sleep session using the device's sleep tracking capabilities.
3. Open the sleep tracking view within the app.
4. Verify that detailed sleep stages (e.g., light, deep, REM sleep) and patterns are displayed.
5. Check that the displayed data matches the recorded sleep data.

* **Postconditions:** Users can view detailed sleep stages and patterns within the app

* **Expected Result:** The app displays detailed sleep stages and patterns accurately.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify the App Provides Insights into Sleep Disturbances****

* **Test Case ID:** SCT-03

* **Test Case Title:** Verify the app provides insights into sleep disturbances

* **Test Case Description:** This test case verifies that the app provides accurate insights into sleep disturbances.

* **Test Suite:** Sleep Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

+ User has enabled sleep tracking on their device

+ User has at least one sleep session recorded with disturbances (e.g., sleep talking, sleepwalking)

* **Test Data:** No specific test data needed, as this test case verifies system behavior

* **Test Steps:**

1. Enable sleep tracking on the user's device.

2. Simulate a sleep session with disturbances using the device's sleep tracking capabilities.

3. Open the sleep tracking view within the app.

4. Verify that the app highlights sleep disturbances and provides insights into their causes.

5. Check that the displayed data matches the recorded sleep data.

* **Postconditions:** Users receive insights into sleep disturbances

* **Expected Result:** The app accurately provides insights into sleep disturbances.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify Users Receive Recommendations for Improving Sleep Quality****

* **Test Case ID:** SCT-04

* **Test Case Title:** Verify users receive recommendations for improving sleep quality

* **Test Case Description:** This test case verifies that users receive accurate and actionable recommendations for improving sleep quality.

* **Test Suite:** Sleep Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

+ User has enabled sleep tracking on their device

+ User has at least one sleep session recorded

* **Test Data:** No specific test data needed, as this test case verifies system behavior

* **Test Steps:**

1. Enable sleep tracking on the user's device.
2. Simulate a sleep session using the device's sleep tracking capabilities.
3. Open the sleep tracking view within the app.
4. Verify that the app provides actionable recommendations for improving sleep quality.
5. Check that the recommended actions are based on the user's sleep data and goals.

* **Postconditions:** Users receive recommendations for improving sleep quality

* **Expected Result:** The app accurately provides actionable recommendations for improving sleep quality.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify Sleep Data Can be Viewed in Daily, Weekly, and Monthly Summaries****

* **Test Case ID:** SCT-05

* **Test Case Title:** Verify sleep data can be viewed in daily, weekly, and monthly summaries

* **Test Case Description:** This test case verifies that users can view sleep data in various summary formats.

* **Test Suite:** Sleep Tracking Module

* **Test Priority:** Low

* **Preconditions:**

+ User has enabled sleep tracking on their device

+ User has multiple sleep sessions recorded

* **Test Data:** No specific test data needed, as this test case verifies user interface behavior

* **Test Steps:**

1. Enable sleep tracking on the user's device.
2. Simulate multiple sleep sessions using the device's sleep tracking capabilities.
3. Open the sleep tracking view within the app.
4. Verify that users can view sleep data in daily, weekly, and monthly summaries.
5. Check that the displayed data matches the recorded sleep data.

* **Postconditions:** Users can view sleep data in various summary formats

* **Expected Result:** The app accurately displays sleep data in daily, weekly, and monthly summaries.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 81

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Logging

Feature description -

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify

trends in performance and adherence to training programs.

Userstory title -

As a user, I want to log my workouts so that I can track my progress and improve my performance.

Acceptance criteria -

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

Generated Test Cases:

Here are five detailed test cases for the Workout Logging feature of the Fitness Tracker App, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can log various workout types with details****

*** **Test Case ID:** WLT-001**

*** **Test Case Title:** Verify that users can log various workout types with details**

*** **Test Case Description:**** This test case verifies that users can log various workout types with details such as exercise type, duration, intensity, and notes.

*** **Test Suite:** Workout Logging**

*** **Test Priority:** High**

*** **Preconditions:****

+ User has an active account

+ User has access to the workout logging feature

*** **Test Data:**** Various workout types with different exercise types, durations, intensities, and notes

*** **Test Steps:****

1. Log in to the app as a registered user

2. Navigate to the workout logging feature

3. Fill in the workout type, duration, intensity, and notes fields with valid data
4. Submit the form to create a new workout log entry
5. Verify that the workout log entry is saved successfully and displayed on the screen

* **Postconditions:** Workout log entry is saved and displayed on the screen

* **Expected Result:** The workout log entry is saved successfully and displayed on the screen with the correct details

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that workout logs include duration, intensity, and notes****

* **Test Case ID:** WLT-002

* **Test Case Title:** Verify that workout logs include duration, intensity, and notes

* **Test Case Description:** This test case verifies that workout logs include duration, intensity, and notes fields.

* **Test Suite:** Workout Logging

* **Test Priority:** High

* **Preconditions:** User has created a workout log entry

* **Test Data:** Valid workout log entry with duration, intensity, and notes

* **Test Steps:**

1. Log in to the app as a registered user
2. Navigate to the workout logging feature
3. Verify that the workout log entry includes duration, intensity, and notes fields
4. Fill in the duration, intensity, and notes fields with valid data
5. Submit the form to update the workout log entry

* **Postconditions:** Workout log entry is updated successfully

* **Expected Result:** Workout log entry includes duration, intensity, and notes fields with valid data

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can edit and delete workout entries

* **Test Case ID:** WLT-003

* **Test Case Title:** Verify that users can edit and delete workout entries

* **Test Case Description:** This test case verifies that users can edit and delete workout entries.

* **Test Suite:** Workout Logging

* **Test Priority:** High

* **Preconditions:** User has created a workout log entry

* **Test Data:** Valid workout log entry with edit and delete capabilities

* **Test Steps:**

1. Log in to the app as a registered user

2. Navigate to the workout logging feature

3. Edit and delete a workout log entry, verifying that the changes are saved successfully

4. Verify that the workout log entry is deleted successfully

* **Postconditions:** Workout log entry is edited/deleted successfully

* **Expected Result:** Workout log entry is edited/deleted successfully

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that data is stored and viewable for tracking progress****

*** **Test Case ID:** WLT-004**

*** **Test Case Title:** Verify that data is stored and viewable for tracking progress**

*** **Test Case Description:** This test case verifies that workout log entries are stored and viewable for tracking progress.**

*** **Test Suite:** Workout Logging**

*** **Test Priority:** High**

*** **Preconditions:** User has created multiple workout log entries**

*** **Test Data:** Valid workout log entries with different exercise types, durations, intensities, and notes**

*** **Test Steps:****

1. Log in to the app as a registered user
2. Navigate to the workout logging feature
3. Verify that workout log entries are displayed in chronological order
4. Filter workout log entries by date and workout type
5. Verify that filtered workout log entries are displayed correctly

*** **Postconditions:** Workout log entries are displayed correctly on the screen**

*** **Expected Result:** Workout log entries are stored and viewable for tracking progress**

*** **Severity:** Minor**

*** **Priority:** High**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

****Test Case 5: Verify that users can filter logs by date and workout type****

*** **Test Case ID:** WLT-005**

* **Test Case Title:** Verify that users can filter logs by date and workout type

* **Test Case Description:** This test case verifies that users can filter workout log entries by date and workout type.

* **Test Suite:** Workout Logging

* **Test Priority:** High

* **Preconditions:** User has created multiple workout log entries

* **Test Data:** Valid workout log entries with different exercise types, durations, intensities, and notes

* **Test Steps:**

1. Log in to the app as a registered user
2. Navigate to the workout logging feature
3. Filter workout log entries by date
4. Filter workout log entries by workout type
5. Verify that filtered workout log entries are displayed correctly

* **Postconditions:** Workout log entries are filtered correctly on the screen

* **Expected Result:** Workout log entries are filtered correctly by date and workout type

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 82

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Goal Setting

Feature description -

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

Userstory title -

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

Acceptance criteria -

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

Generated Test Cases:

Here are five detailed test cases for the Goal Setting feature of the Fitness Tracker App, following the provided Test Case Creation Template:

****Test Case 1: Verify that users can set goals for various fitness metrics****

*** **Test Case ID:** GOAL-001**

*** **Test Case Title:** Verify that users can set goals for various fitness metrics**

*** **Test Case Description:**** This test case ensures that users can set goals for various fitness metrics such as steps taken, calories burned, distance traveled, and heart rate. The test case covers the acceptance criterion "Users can set goals for various fitness metrics."

* **Test Suite:** Goal Setting

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a profile set up

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the goal setting page
2. Select a fitness metric (e.g., steps taken) from the dropdown menu
3. Enter a specific goal value (e.g., 10,000 steps per day)
4. Click the "Save Goal" button
5. Verify that the goal is displayed on the goal setting page

* **Postconditions:** None

* **Expected Result:** The goal is successfully set and displayed on the goal setting page.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that progress tracking is available for each goal

* **Test Case ID:** GOAL-002

* **Test Case Title:** Verify that progress tracking is available for each goal

* **Test Case Description:** This test case ensures that progress tracking is available for each goal set by the user. The test case covers the acceptance criterion "Progress tracking is available for each goal."

* **Test Suite:** Goal Setting

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a profile set up

+ User has set a goal

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the goal setting page

2. Select a goal from the list of goals

3. Verify that the progress tracking chart is displayed

4. Verify that the progress tracking chart shows the user's progress towards the goal

* **Postconditions:** None

* **Expected Result:** The progress tracking chart is displayed and shows the user's progress towards the goal.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users receive motivational feedback and reminders

* **Test Case ID:** GOAL-003

* **Test Case Title:** Verify that users receive motivational feedback and reminders

* **Test Case Description:** This test case ensures that users receive motivational feedback and reminders to help them stay motivated and on track with their goals. The test case covers the acceptance criterion "Users receive motivational feedback and reminders."

* **Test Suite:** Goal Setting

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a profile set up

+ User has set a goal

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the goal setting page

2. Select a goal from the list of goals

3. Verify that the motivational feedback and reminders are displayed

4. Verify that the motivational feedback and reminders are relevant to the user's progress

* **Postconditions:** None

* **Expected Result:** The user receives motivational feedback and reminders to help them stay motivated and on track with their goals.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that goals can be adjusted or updated as needed

* **Test Case ID:** GOAL-004

* **Test Case Title:** Verify that goals can be adjusted or updated as needed

* **Test Case Description:** This test case ensures that users can adjust or update their goals as needed. The test case covers the acceptance criterion "Goals can be adjusted or updated as needed."

* **Test Suite:** Goal Setting

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a profile set up

+ User has set a goal

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the goal setting page

2. Select a goal from the list of goals

3. Update the goal value (e.g., increase the target steps per day)

4. Click the "Save Goal" button

5. Verify that the updated goal is displayed on the goal setting page

* **Postconditions:** None

* **Expected Result:** The goal is successfully updated and displayed on the goal setting page.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that the app provides a summary of goal attainment****

* **Test Case ID:** GOAL-005

* **Test Case Title:** Verify that the app provides a summary of goal attainment

* **Test Case Description:** This test case ensures that the app provides a summary of goal attainment for each goal set by the user. The test case covers the acceptance criterion "The app provides a summary of goal attainment."

* **Test Suite:** Goal Setting

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a profile set up

+ User has set a goal

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the goal setting page

2. Select a goal from the list of goals

3. Verify that the summary of goal attainment is displayed

4. Verify that the summary shows the user's progress towards the goal

* **Postconditions:** None

* **Expected Result:** The app provides a summary of goal attainment for each goal set by the user.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 83

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Plans

Feature description -

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

Userstory title -

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

Acceptance criteria -

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

Generated Test Cases:

Here are five detailed test cases for the "Workout Plans" feature, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can select workout plans based on fitness levels****

*** **Test Case ID:** TC-WP-001**

*** **Test Case Title:** Verify that users can select workout plans based on fitness levels.**

*** **Test Suite:** Workout Plans**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged into the app

+ User has access to the workout plans feature

*** **Test Data:** No test data needed**

****Test Steps:****

1. Login to the app and navigate to the workout plans feature
2. Select the "Select Plan" option
3. Choose a fitness level (e.g. beginner, intermediate, advanced)
4. Select a workout plan based on the chosen fitness level
5. Verify that the selected plan is saved and visible in the user's plan list

****Postconditions:****

+ User is at the user's dashboard with the selected plan highlighted

****Expected Result:**** User is able to select a workout plan based on fitness level.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 2: Verify that plans include detailed exercise routines and schedules****

****Test Case ID:**** TC-WP-002

****Test Case Title:**** Verify that plans include detailed exercise routines and schedules.

****Test Suite:**** Workout Plans

****Test Priority:**** High

****Preconditions:****

+ User is logged into the app

+ User has selected a workout plan

****Test Data:**** No test data needed

****Test Steps:****

1. Login to the app and navigate to the workout plans feature
2. Select a workout plan

3. Verify that the plan includes detailed exercise routines and schedules

4. Check if the plan includes the following details:

- Exercise routines
- Schedule
- Rest days
- Warm-up and cool-down exercises

5. Verify that the plan is viewable on a mobile device

* **Postconditions:**

+ User can view the plan details

* **Expected Result:** Workout plan includes detailed exercise routines and schedules.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that users can track adherence and progress within the plan****

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

* **Test Case Title:** Verify that users can track adherence and progress within the plan.

* **Test Suite:** Workout Plans

* **Test Priority:** High

* **Preconditions:**

+ User is logged into the app

+ User has selected a workout plan

* **Test Data:** No test data needed

* **Test Steps:**

1. Login to the app and navigate to the workout plans feature

2. Select a workout plan

3. Track adherence and progress within the plan by:

- Logging workouts
- Tracking daily progress
- Viewing the plan's calendar

4. Verify that the user's adherence and progress are displayed correctly

5. Check if the app sends notifications and reminders for workouts

* **Postconditions:**

+ User's adherence and progress are tracked

* **Expected Result:** Users can track adherence and progress within the plan.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that the app provides notifications and reminders for workouts

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

* **Test Case Title:** Verify that the app provides notifications and reminders for workouts.

* **Test Suite:** Workout Plans

* **Test Priority:** High

* **Preconditions:**

+ User is logged into the app

+ User has selected a workout plan

* **Test Data:** No test data needed

* **Test Steps:**

1. Login to the app and navigate to the workout plans feature

2. Select a workout plan
3. Set up notifications and reminders for workouts
4. Verify that the app sends notifications and reminders for workouts
5. Check if the notifications and reminders are customizable

* **Postconditions:**

+ User receives notifications and reminders for workouts

* **Expected Result:** App provides notifications and reminders for workouts.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that users can adjust plans based on progress and preferences****

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

* **Test Case Title:** Verify that users can adjust plans based on progress and preferences.

* **Test Suite:** Workout Plans

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged into the app

+ User has selected a workout plan

+ User has tracked adherence and progress within the plan

* **Test Data:** No test data needed

* **Test Steps:**

1. Login to the app and navigate to the workout plans feature
2. Select a workout plan
3. View the user's progress and adherence within the plan

4. Adjust the plan based on progress and preferences

5. Verify that the adjusted plan reflects the changes

* **Postconditions:**

+ User's plan is adjusted based on progress and preferences

* **Expected Result:** Users can adjust plans based on progress and preferences.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases cover all the acceptance criteria for the "Workout Plans" feature and aim to ensure that the app provides a high-quality user experience for tracking physical activity, exercise routines, and health metrics.

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:

Here are five detailed test cases for the Community Support feature:

****Test Case 1: Verify that users can join and participate in community groups****

*** **Test Case ID:** CSC-1**

*** **Test Case Title:** Verify that users can join and participate in community groups**

*** **Test Case Description:** This test case ensures that users can successfully join and participate in community groups, allowing them to connect with others who share similar interests and goals.**

*** **Test Suite:** Community Support**

*** **Test Priority:** High**

*** **Preconditions:****

+ User has a valid account and is logged in.

+ Community groups exist.

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the app and navigate to the community groups page.

2. Select a community group to join.
3. Fill out the required information to join the group.
4. Click the "Join Group" button.
5. Verify that the user is now a member of the group.
6. Post a message or update to the group.

* **Postconditions:**

+ User is a member of the group.

+ Post is visible to other group members.

* **Expected Result:** User is able to successfully join and participate in a community group.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that users can share progress updates and achievements****

* **Test Case ID:** CSC-2

* **Test Case Title:** Verify that users can share progress updates and achievements

* **Test Case Description:** This test case ensures that users can share their progress updates and achievements with the community, allowing others to stay motivated and inspired.

* **Test Suite:** Community Support

* **Test Priority:** High

* **Preconditions:**

+ User has a valid account and is logged in.

+ User has achieved a milestone or progress update to share.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the community groups page.
2. Select a community group to share with.
3. Fill out the required information to share the progress update.
4. Click the "Share Update" button.
5. Verify that the progress update is visible to other group members.

* **Postconditions:**

+ Progress update is visible to other group members.

* **Expected Result:** User is able to successfully share a progress update or achievement with the community.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that the app facilitates participation in community challenges****

* **Test Case ID:** CSC-3

* **Test Case Title:** Verify that the app facilitates participation in community challenges

* **Test Case Description:** This test case ensures that the app provides a seamless experience for users to participate in community challenges, allowing them to stay motivated and engaged.

* **Test Suite:** Community Support

* **Test Priority:** Medium

* **Preconditions:**

+ User has a valid account and is logged in.

+ Community challenge exists.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the community challenges page.
2. Select a community challenge to participate in.
3. Fill out the required information to participate.
4. Click the "Join Challenge" button.
5. Verify that the user is now a participant in the challenge.
6. Update progress or share an update about their participation.

* **Postconditions:**

+ User is a participant in the community challenge.

+ Progress or updates are visible to other challenge participants.

* **Expected Result:** User is able to successfully participate in a community challenge.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that users can offer and receive support and encouragement****

* **Test Case ID:** CSC-4

* **Test Case Title:** Verify that users can offer and receive support and encouragement

* **Test Case Description:** This test case ensures that users can offer and receive support and encouragement from others, creating a supportive community environment.

* **Test Suite:** Community Support

* **Test Priority:** Medium

* **Preconditions:**

+ User has a valid account and is logged in.

+ User or another user needs support or encouragement.

* **Test Data:** No test data needed

****Test Steps:****

1. Log in to the app and navigate to a community group.
2. Select a user to offer support to.
3. Fill out the required information to leave a supportive comment.
4. Click the "Leave Comment" button.
5. Verify that the comment is visible to the user who needs support.

****Postconditions:****

+ Supportive comment is visible to the user who needs support.

****Expected Result:**** User is able to successfully offer support and encouragement to another user.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 5: Verify that community activity and engagement are tracked****

****Test Case ID:**** CSC-5

****Test Case Title:**** Verify that community activity and engagement are tracked

****Test Case Description:**** This test case ensures that the app tracks community activity and engagement, providing valuable insights for users and administrators.

****Test Suite:**** Community Support

****Test Priority:**** Low

****Preconditions:****

+ User has a valid account and is logged in.

+ Community activity exists.

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the app and navigate to the community dashboard.
2. Verify that community activity metrics are correctly tracked.
3. Verify that user engagement metrics are correctly tracked.

* **Postconditions:**

+ Community activity and engagement metrics are correctly tracked.

* **Expected Result:** Community activity and engagement metrics are correctly tracked.

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 85

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Wearable Devices

Feature description -

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

Userstory title -

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

Acceptance criteria -

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

Generated Test Cases:

Here are the 5 detailed test cases for the user story: As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

****Test Case 1: Verify that the app supports integration with multiple wearable devices****

* ****Test Case Title:**** Verify that the app supports integration with multiple wearable devices

* ****Test Case Description:**** This test case aims to ensure that the Fitness Tracker App supports integration with various wearable devices, including fitness trackers, smartwatches, and heart rate monitors. The test-case will cover the ability of the app to connect with multiple wearables simultaneously and accurately retrieve data from each device.

* ****Test Suite:**** Wearable Device Integration

* ****Test Priority:**** High

* ****Preconditions:**** 1. User has an account in the Fitness Tracker App and has installed the required wearable device drivers.

* 2. The wearable device is paired with the user's smartphone.

* 3. The user has enabled data sharing permissions for the wearable device.

* ****Test Data:**** Multiple wearable devices with varying sensors and data formats.

* ****Test Steps:****

1. Launch the Fitness Tracker App and navigate to the wearable device integration feature.

2. Confirm that the app lists the supported wearable devices attached to the user's smartphone.
3. Select multiple wearable devices to connect to the app.
4. Verify that the app successfully syncs data from each wearable device.

* **Postconditions:** None.

* **Expected Result:** The app successfully integrates with multiple wearable devices and retrieves data accurately.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 86

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Water Intake Tracking

Feature description -

Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

Userstory title -

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

Acceptance criteria -

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

Generated Test Cases:

Here are five detailed test cases for the Water Intake Tracking feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that users can log their daily water intake****

*** **Test Case ID:** FIT-001**

*** **Test Case Title:** Verify that users can log their daily water intake**

*** **Test Case Description:**** This test case ensures that users can successfully log their daily water intake, which is a critical feature of the Water Intake Tracking feature.

*** **Test Suite:** Water Intake Tracking**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in to the app

+ User has access to the Water Intake Tracking feature

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Launch the app and navigate to the Water Intake Tracking feature
2. Click on the "Log Water Intake" button
3. Enter a valid amount of water consumed (e.g., 1 liter)

4. Click on the "Save" button

* **Postconditions:**

+ User's water intake is successfully logged

+ User's hydration progress is updated

* **Expected Result:** User's water intake is successfully logged, and hydration progress is updated.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that the app calculates total water consumption for the day

* **Test Case ID:** FIT-002

* **Test Case Title:** Verify that the app calculates total water consumption for the day

* **Test Case Description:** This test case ensures that the app accurately calculates the total water consumption for the day, which is a critical feature of the Water Intake Tracking feature.

* **Test Suite:** Water Intake Tracking

* **Test Priority:** High

* **Preconditions:**

+ User has logged water intake for the day

+ User has access to the Water Intake Tracking feature

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the app and navigate to the Water Intake Tracking feature

2. View the total water consumption for the day

3. Verify that the total water consumption is accurate

* **Postconditions:**

+ Total water consumption is accurately calculated

* **Expected Result:** Total water consumption is accurately calculated.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can set daily hydration goals

* **Test Case ID:** FIT-003

* **Test Case Title:** Verify that users can set daily hydration goals

* **Test Case Description:** This test case ensures that users can successfully set daily hydration goals, which is a critical feature of the Water Intake Tracking feature.

* **Test Suite:** Water Intake Tracking

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has access to the Water Intake Tracking feature

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the app and navigate to the Water Intake Tracking feature

2. Click on the "Set Hydration Goal" button

3. Enter a valid hydration goal (e.g., 2 liters)

4. Click on the "Save" button

* **Postconditions:**

+ User's hydration goal is successfully set

+ User's hydration progress is updated

* **Expected Result:** User's hydration goal is successfully set, and hydration progress is updated.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that reminders are provided to encourage regular water intake

* **Test Case ID:** FIT-004

* **Test Case Title:** Verify that reminders are provided to encourage regular water intake

* **Test Case Description:** This test case ensures that reminders are provided to encourage regular water intake, which is a critical feature of the Water Intake Tracking feature.

* **Test Suite:** Water Intake Tracking

* **Test Priority:** High

* **Preconditions:**

+ User has set a hydration goal

+ User has access to the Water Intake Tracking feature

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the app and navigate to the Water Intake Tracking feature
2. Verify that reminders are provided to encourage regular water intake
3. Verify that reminders are accurate and timely

* **Postconditions:**

+ Reminders are provided to encourage regular water intake

* **Expected Result:** Reminders are provided to encourage regular water intake.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that users can view hydration progress and trends

* **Test Case ID:** FIT-005

* **Test Case Title:** Verify that users can view hydration progress and trends

* **Test Case Description:** This test case ensures that users can successfully view hydration progress and trends, which is a critical feature of the Water Intake Tracking feature.

* **Test Suite:** Water Intake Tracking

* **Test Priority:** High

* **Preconditions:**

+ User has logged water intake for multiple days

+ User has access to the Water Intake Tracking feature

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the app and navigate to the Water Intake Tracking feature

2. View hydration progress and trends

3. Verify that hydration progress and trends are accurate and up-to-date

* **Postconditions:**

+ Hydration progress and trends are accurately displayed

* **Expected Result:** Hydration progress and trends are accurately displayed.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 87

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Nutrition Tracking

Feature description -

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

Userstory title -

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

Acceptance criteria -

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

Generated Test Cases:

Here are five detailed test cases that cover all the acceptance criteria for the Nutrition Tracking feature:

****Test Case 1: Verify that users can log their daily meals and snacks****

* **Test Case ID:** NT1

* **Test Case Title:** Verify that users can log their daily meals and snacks

* **Test Case Description:** This test case ensures that users can log their meals and snacks correctly, including the date, time, and nutrient information. It also verifies that the app does not allow users to log invalid or duplicate entries.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has access to the nutrition tracking feature

* **Test Data:** User input for meal logging (e.g., date, time, meal name, nutrient information)

* **Test Steps:**

1. Log in to the app and navigate to the nutrition tracking feature
2. Select the date and time for a meal
3. Enter a valid meal name and nutrient information
4. Submit the meal entry
5. Verify that the entry is displayed in the user's nutrition history
6. Attempt to log an invalid or duplicate entry to verify error handling

* **Postconditions:** Verify that the user's nutrition history is accurate and up-to-date

* **Expected Result:** The user's meal entries are correctly logged and displayed in the nutrition history

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that the app calculates macronutrient intake (carbs, fats, proteins)

* **Test Case ID:** NT2

* **Test Case Title:** Verify that the app calculates macronutrient intake (carbs, fats, proteins)

* **Test Case Description:** This test case ensures that the app accurately calculates the user's macronutrient intake based on their meal entries. It also verifies that the app provides a clear display of nutrient information.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has access to the nutrition tracking feature

+ User has entered meal entries with accurate nutrient information

* **Test Data:** Meal entries with varying amounts of macronutrients (e.g., carbs, fats, proteins)

* **Test Steps:**

1. Log in to the app and navigate to the nutrition tracking feature
2. Enter meal entries with varying amounts of macronutrients
3. Verify that the app calculates the macronutrient intake accurately
4. Verify that the app displays the calculated macronutrient intake clearly

* **Postconditions:** Verify that the user's macronutrient intake is accurate and up-to-date

* **Expected Result:** The app accurately calculates the user's macronutrient intake

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can set nutritional goals and track progress

* **Test Case ID:** NT3

* **Test Case Title:** Verify that users can set nutritional goals and track progress

* **Test Case Description:** This test case ensures that users can set nutritional goals and track their progress towards those goals. It also verifies that the app provides a clear display of goal settings and progress tracking.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in to the app

+ User has access to the nutrition tracking feature

+ User has entered meal entries with accurate nutrient information

* **Test Data:** User input for goal settings (e.g., target macros, weight targets)

* **Test Steps:**

1. Log in to the app and navigate to the nutrition tracking feature
2. Set a valid nutritional goal (e.g., increase protein intake)
3. Verify that the app allows the user to track progress towards the goal
4. Verify that the app displays the user's progress clearly

* **Postconditions:** Verify that the user's nutritional progress is accurate and up-to-date

* **Expected Result:** The user can set and track their nutritional goals accurately

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that the app provides nutritional information and suggestions****

* **Test Case ID:** NT4

* **Test Case Title:** Verify that the app provides nutritional information and suggestions

* **Test Case Description:** This test case ensures that the app provides accurate and relevant nutritional

information and suggestions based on the user's meal entries and goals.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has access to the nutrition tracking feature

+ User has entered meal entries with accurate nutrient information

* **Test Data:** Meal entries with varying amounts of macronutrients (e.g., carbs, fats, proteins)

* **Test Steps:**

1. Log in to the app and navigate to the nutrition tracking feature

2. Enter meal entries with varying amounts of macronutrients

3. Verify that the app provides accurate and relevant nutritional information and suggestions

4. Verify that the app displays the suggestions clearly

* **Postconditions:** Verify that the user receives accurate and helpful nutritional information and suggestions

* **Expected Result:** The app provides accurate and relevant nutritional information and suggestions

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that users can view their nutritional history and trends****

* **Test Case ID:** NT5

* **Test Case Title:** Verify that users can view their nutritional history and trends

* **Test Case Description:** This test case ensures that users can view their nutritional history and trends accurately, including meal entries and macronutrient intake.

* **Test Suite:** Nutrition Tracking

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in to the app

+ User has access to the nutrition tracking feature

+ User has entered meal entries with accurate nutrient information

* **Test Data:** Meal entries with varying amounts of macronutrients (e.g., carbs, fats, proteins)

* **Test Steps:**

1. Log in to the app and navigate to the nutrition tracking feature
2. Verify that the app displays the user's nutritional history accurately
3. Verify that the app displays the user's nutritional trends accurately

* **Postconditions:** Verify that the user's nutritional history and trends are accurate and up-to-date

* **Expected Result:** The user can view their nutritional history and trends accurately

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 88

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

Feature name -

Progress Reports

Feature description -

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

Userstory title -

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

Acceptance criteria -

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

Generated Test Cases:

Here are five detailed test cases, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that Users Receive Weekly Progress Reports****

=====

Test Case ID: PROG-001

Test Case Title: Verify that Users Receive Weekly Progress Reports

Test Case Description:

This test case ensures that users receive weekly progress reports as expected. It covers the acceptance criterion that users receive weekly and monthly progress reports.

Test Suite: Progress Reports Module

Test Priority: High

Preconditions:

- * User has created an account and logged in successfully
- * User has tracked physical activity and workouts during the test period

Test Data: User tracking data for the test period

Test Steps:

1. Log in to the fitness tracker app
2. Navigate to the Progress Reports section
3. Verify that the weekly report is displayed, including activity levels, workout performance, and goal attainment
4. Verify that the report is dated for the correct week

Postconditions:

- * Weekly report is successfully displayed
- * User can view and analyze the report

Expected Result:

The user receives a weekly progress report that includes their activity levels, workout performance, and goal attainment.

Severity: Major

Priority: High

Type of Testing: Functional Testing

****Test Case 2: Verify that Reports Include Activity Levels, Workout Performance, and Goal Attainment****

=====

=====

Test Case ID: PROG-002

Test Case Title: Verify that Reports Include Activity Levels, Workout Performance, and Goal Attainment

Test Case Description:

This test case ensures that the weekly and monthly progress reports accurately include activity levels, workout performance, and goal attainment as expected.

Test Suite: Progress Reports Module

Test Priority: High

Preconditions:

- * User has created an account and logged in successfully
- * User has tracked physical activity and workouts during the test period

Test Data: User tracking data for the test period

Test Steps:

1. Log in to the fitness tracker app
2. Navigate to the Progress Reports section
3. Verify that the weekly and monthly reports include the following metrics:
 - * Activity levels (steps taken, distance traveled, calories burned)
 - * Workout performance (number of workouts completed, average duration)
 - * Goal attainment (percentage of goals achieved)
4. Verify that the metrics are accurate and up-to-date

Postconditions:

- * Reports include accurate activity levels, workout performance, and goal attainment metrics
- * User can view and analyze the reports

Expected Result:

The weekly and monthly progress reports accurately include activity levels, workout performance, and goal attainment metrics.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case 3: Verify that Users Can View Detailed Breakdowns of Achievements and Trends

=====

=====

Test Case ID: PROG-003

Test Case Title: Verify that Users Can View Detailed Breakdowns of Achievements and Trends

Test Case Description:

This test case ensures that users can view detailed breakdowns of achievements and trends as expected.

Test Suite: Progress Reports Module

Test Priority: High

Preconditions:

- * User has created an account and logged in successfully
- * User has tracked physical activity and workouts during the test period

Test Data: User tracking data for the test period

Test Steps:

1. Log in to the fitness tracker app
2. Navigate to the Progress Reports section
3. Select a specific week or month to view detailed breakdowns
4. Verify that the report displays a detailed breakdown of achievements and trends, including:
 - * Achievements (numbers of workouts completed, distance traveled, etc.)
 - * Trends (progress over time, fluctuations, etc.)
5. Verify that the detailed breakdowns are accurate and up-to-date

Postconditions:

- * User can view detailed breakdowns of achievements and trends

* User can analyze and make informed decisions

Expected Result:

The user can view detailed breakdowns of achievements and trends.

Severity: Major

Priority: High

Type of Testing: Functional Testing

****Test Case 4: Verify that The App Provides Recommendations Based on Report Findings****

=====

=====

Test Case ID: PROG-004

Test Case Title: Verify that The App Provides Recommendations Based on Report Findings

Test Case Description:

This test case ensures that the app provides recommendations based on report findings as expected.

Test Suite: Progress Reports Module

Test Priority: High

Preconditions:

- * User has created an account and logged in successfully
- * User has tracked physical activity and workouts during the test period

Test Data: User tracking data for the test period

Test Steps:

1. Log in to the fitness tracker app
2. Navigate to the Progress Reports section
3. Select a specific week or month to view recommendations
4. Verify that the app provides recommendations based on report findings, including:

- * Suggestions for increasing activity levels
- * Proposals for improving workout performance
- * Advice for identifying and addressing areas for improvement

5. Verify that the recommendations are accurate and relevant

Postconditions:

- * User receives recommendations based on report findings
- * User can implement suggestions to improve their fitness

Expected Result:

The app provides recommendations based on report findings.

Severity: Major

Priority: High

Type of Testing: Functional Testing

****Test Case 5: Verify that Users Can Customize the Frequency and Content of Reports****

=====

=====

Test Case ID: PROG-005

Test Case Title: Verify that Users Can Customize the Frequency and Content of Reports

Test Case Description:

This test case ensures that users can customize the frequency and content of reports as expected.

Test Suite: Progress Reports Module

Test Priority: High

Preconditions:

- * User has created an account and logged in successfully
- * User has previously viewed and logged in to the app

Test Data: User profile and preferences data

Test Steps:

1. Log in to the fitness tracker app
2. Navigate to the Settings section
3. Select the Progress Reports option
4. Verify that users can customize the frequency and content of reports, including:
 - * Choosing the reporting frequency (daily, weekly, monthly)
 - * Selecting the metrics to display (activity levels, workout performance, goal attainment)
 - * Customizing the report layout and appearance
5. Verify that the customizations are saved and applied

Postconditions:

- * User can customize the frequency and content of reports
- * User can tailor the reports to their preferences

Expected Result:

Users can customize the frequency and content of reports.

Severity: Major

Priority: High

Type of Testing: Functional Testing

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 the test cases for the Fitness Tracker App, covering all acceptance criteria related to the "In-App Challenges" feature.

****Test Case 1: Verify that Users Can Join In-App Challenges****

Test Case ID	TC01
Test Case Title	Verify that users can join various in-app challenges
Test Case Description	As a user, I want to participate in in-app challenges, so that I can stay motivated and achieve my fitness goals. This test case ensures that users can successfully join in-app challenges.
Test Suite	In-App Challenges Test Suite
Test Priority	High

| ****Preconditions**** | The user has a registered account in the fitness tracker app. The app has available in-app challenges. The user has an active internet connection.

| ****Test Data**** | Default user account with available in-app challenges |

| ****Test Steps**** | 1. Navigate to the challenges menu. 2. Browse available challenges. 3. Click on a challenge to join. |

| ****Postconditions**** | The user is successfully enrolled in the selected challenge. The user's progress is displayed within the challenge. The challenge's objectives and rules are clearly visible.

| ****Expected Result**** | The user is able to join a challenge, and their progress is tracked within the app. |

| ****Severity**** | Major |

| ****Priority**** | High |

| ****Type of Testing**** | Functional Testing |

| ****Test Case Approach**** | Positive |

****Test Case 2: Verify that Challenges Have Clear Objectives and Rules****

| ****Test Case ID**** | TC02 |

| ****Test Case Title**** | Verify that challenges have clear objectives and rules |

| ****Test Case Description**** | As a user, I want challenges to have clear objectives and rules, so that I understand how to participate and succeed. This test case ensures that challenges have well-defined rules and objectives. |

| ****Test Suite**** | In-App Challenges Test Suite |

| ****Test Priority**** | High |

| ****Preconditions**** | The user has joined a challenge. The app has available in-app challenges with clear objectives and rules. The user has an active internet connection.

| ****Test Data**** | Participating user with access to a challenge with clear objectives and rules |

| ****Test Steps**** | 1. Load the specific challenge. 2. Review the challenge's objectives and rules. 3. Record the

challenge's specific details. |

| **Postconditions** | The challenge's objectives and rules are clearly stated. The rules are concise and unambiguous. The user can understand how to participate and succeed.

| **Expected Result** | The challenge's objectives and rules are clearly communicated, and the user can understand how to participate. |

| **Severity** | Major |

| **Priority** | High |

| **Type of Testing** | Functional Testing |

| **Test Case Approach** | Positive |

Test Case 3: Verify that Users Can Track Their Progress within a Challenge

| **Test Case ID** | TC03 |

| **Test Case Title** | Verify that users can track their progress within a challenge |

| **Test Case Description** | As a user, I want to track my progress within a challenge, so that I can stay motivated and inspired to reach my fitness goals. This test case ensures that users can see their progress within the app. |

| **Test Suite** | In-App Challenges Test Suite |

| **Test Priority** | Medium |

| **Preconditions** | The user is participating in a challenge. The app has available in-app challenges with tracking capabilities. The user has an active internet connection.

| **Test Data** | Participating user with access to a challenge with tracking capabilities |

| **Test Steps** | 1. Load the specific challenge. 2. Review the user's progress within the challenge. 3. Verify that the progress is updated in real-time. |

| **Postconditions** | The user's progress within the challenge is displayed. The progress is accurately updated in real-time. The user can see their progress history.

| **Expected Result** | The user's progress within the challenge is accurately tracked and displayed in

real-time. |

| **Severity** | Minor |

| **Priority** | Medium |

| **Type of Testing** | Functional Testing |

| **Test Case Approach** | Positive |

Test Case 4: Verify that Rewards Are Provided for Achieving Challenge Milestones

| **Test Case ID** | TC04 |

| **Test Case Title** | Verify that rewards are provided for achieving challenge milestones |

| **Test Case Description** | As a user, I want to earn rewards for achieving challenge milestones, so that I can feel motivated and inspired to reach my fitness goals. This test case ensures that rewards are provided for completing specific milestones. |

| **Test Suite** | In-App Challenges Test Suite |

| **Test Priority** | High |

| **Preconditions** |

- The user is participating in a challenge with rewards.
- The app has available in-app challenges with rewards.
- The user has an active internet connection.

| **Test Data** | Participating user with access to a challenge with rewards |

| **Test Steps** | 1. Load the specific challenge. 2. Complete specific milestones to receive rewards. 3. Verify that the rewards are delivered. |

| **Postconditions** |

- The user receives the specified rewards for achieving milestones.
- The rewards are accurately delivered in the app.
- The user can access the rewards.

| **Expected Result** | Rewards are provided for achieving specific challenge milestones. |

| **Severity** | Major |

| **Priority** | High |

| **Type of Testing** | Functional Testing |

| **Test Case Approach** | Positive |

****Test Case 5: Verify that Users Can Compete with Friends and View Leaderboards****

| ****Test Case ID**** | TC05 |

| ****Test Case Title**** | Verify that users can compete with friends and view leaderboards |

| ****Test Case Description**** | As a user, I want to compete with friends and view leaderboards, so that I can stay motivated and inspired to reach my fitness goals. This test case ensures that users can participate in friendly competition and view leaderboards. |

| ****Test Suite**** | In-App Challenges Test Suite |

| ****Test Priority**** | High |

| ****Preconditions**** | The user has friends in the app. The app has available in-app challenges that support friendly competition. The user has an active internet connection.

| ****Test Data**** | Participating user with friends in the app |

| ****Test Steps**** | 1. Load the specific challenge. 2. Invite friends to participate in the challenge. 3. View leaderboards to compare progress. |

| ****Postconditions**** | The user can see their progress in comparison to their friends. The leaderboard accurately displays user rankings. The user can adjust their strategy based on the leaderboard.

| ****Expected Result**** | Users can compete with friends and view leaderboards to stay motivated and inspired.

|

| ****Severity**** | Major |

| ****Priority**** | High |

| ****Type of Testing**** | Functional Testing |

| ****Test Case Approach**** | Positive |

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Customizable Dashboards

Feature description -

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

Userstory title -

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Acceptance criteria -

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

Generated Test Cases:

Here are five detailed test cases for the Customizable Dashboards feature, each covering one or more of the acceptance criteria:

****Test Case 1: Verify that users can select which metrics to display on their dashboard****

* **Test Case ID:** TC-001

* **Test Case Title:** Verify that users can select which metrics to display on their dashboard

* **Test Case Description:** This test case ensures that users can choose which fitness metrics to display on their dashboard, including steps taken, calories burned, distance traveled, and heart rate.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a dashboard set up

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the dashboard
2. Click on the "Customize" button
3. Select the metrics to display on the dashboard (e.g., steps taken, calories burned)
4. Verify that the selected metrics are displayed on the dashboard
5. Repeat steps 3-4 with different combinations of metrics

* **Postconditions:** Dashboard displays the selected metrics

* **Expected Result:** Users can select which metrics to display on their dashboard

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that the app allows customization of dashboard layout and theme

* **Test Case ID:** TC-002

* **Test Case Title:** Verify that the app allows customization of dashboard layout and theme

* **Test Case Description:** This test case ensures that users can customize the layout and theme of their dashboard to suit their preferences.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a dashboard set up

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the dashboard
2. Click on the "Customize" button
3. Select a different layout option (e.g., grid, list)
4. Select a different theme option (e.g., light, dark)
5. Verify that the dashboard layout and theme have changed

* **Postconditions:** Dashboard layout and theme have been customized

* **Expected Result:** Users can customize the dashboard layout and theme

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can save and update their dashboard settings

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

* **Test Case Title:** Verify that users can save and update their dashboard settings

* **Test Case Description:** This test case ensures that users can save their dashboard settings and update them as needed.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a dashboard set up

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the dashboard
2. Make changes to the dashboard settings (e.g., add a new metric)
3. Click on the "Save" button
4. Verify that the dashboard settings have been saved
5. Make further changes to the dashboard settings
6. Repeat steps 3-5 multiple times

* **Postconditions:** Dashboard settings have been saved and updated

* **Expected Result:** Users can save and update their dashboard settings

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that dashboard displays are dynamic and update in real-time

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

* **Test Case Title:** Verify that dashboard displays are dynamic and update in real-time

* **Test Case Description:** This test case ensures that the dashboard displays update in real-time as changes are made to the dashboard settings.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a dashboard set up

+ Device has live updates enabled

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the dashboard

2. Make changes to the dashboard settings (e.g., add a new metric)

3. Verify that the dashboard display updates immediately

4. Repeat steps 2-3 multiple times with different changes

* **Postconditions:** Dashboard display has been updated in real-time

* **Expected Result:** Dashboard displays are dynamic and update in real-time

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that users can reset the dashboard to default settings

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

* **Test Case Title:** Verify that users can reset the dashboard to default settings

* **Test Case Description:** This test case ensures that users can reset their dashboard to its default settings at any time.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

+ User is logged in to the app

+ User has a dashboard set up

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the dashboard

2. Click on the "Reset" button

3. Verify that the dashboard settings have been reset to their default values

4. Repeat steps 2-3 multiple times

* **Postconditions:** Dashboard settings have been reset to default

* **Expected Result:** Users can reset the dashboard to default settings

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional 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:

Here are the detailed test cases for the Social Sharing feature in the Fitness Tracker App, following the provided Test Case Creation Template:

****Test Case 1: Verify that users can share workout summaries and achievements.****

*** **Test Case ID:** SSHA001**

*** **Test Case Title:** Verify that users can share workout summaries and achievements.**

*** **Test Case Description:**** This test case is designed to ensure that users can successfully share their workout summaries and achievements on social media platforms.

*** **Test Suite:** Social Sharing**

*** **Test Priority:** High**

*** **Preconditions:****

- + User account is created and logged in.
- + User has completed at least one workout session.

* **Test Data:** None

* **Test Steps:**

1. Log in to the app with a valid user account.
2. Open the workout summary page for a completed workout session.
3. Click the "Share" button to post the workout summary on social media.
4. Select a social media platform from the options provided.
5. Confirm the post on the social media platform.

* **Postconditions:**

- + Workout summary is posted on the selected social media platform.
- + User is redirected back to the app.

* **Expected Result:** Workout summary is successfully shared on social media.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2: Verify that the app supports sharing on multiple social media platforms.****

* **Test Case ID:** SSHA002

* **Test Case Title:** Verify that the app supports sharing on multiple social media platforms.

* **Test Case Description:** This test case is designed to ensure that the app allows users to share their fitness achievements on multiple social media platforms.

* **Test Suite:** Social Sharing

* **Test Priority:** High

* **Preconditions:**

- + User account is created and logged in.
- + User has completed at least one workout session.

* **Test Data:** None

* **Test Steps:**

1. Log in to the app with a valid user account.
2. Open the workout summary page for a completed workout session.
3. Click the "Share" button to post the workout summary on social media.
4. Select multiple social media platforms from the options provided.
5. Confirm the post on each selected social media platform.

* **Postconditions:**

- + Workout summary is posted on all selected social media platforms.
- + User is redirected back to the app.

* **Expected Result:** Workout summary is successfully shared on multiple social media platforms.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that users can customize the content and format of shared posts.****

* **Test Case ID:** SSHA003

* **Test Case Title:** Verify that users can customize the content and format of shared posts.

* **Test Case Description:** This test case is designed to ensure that users can personalize the content and format of their shared workout summaries and achievements.

* **Test Suite:** Social Sharing

* **Test Priority:** High

* **Preconditions:**

- + User account is created and logged in.
- + User has completed at least one workout session.

* **Test Data:** None

* **Test Steps:**

1. Log in to the app with a valid user account.
2. Open the workout summary page for a completed workout session.
3. Click the "Share" button to post the workout summary on social media.
4. Select the "Edit" option to customize the content and format of the shared post.
5. Make changes to the content and format (e.g., add text, change font, add emojis).
6. Confirm the changes and post the updated content on social media.

* **Postconditions:**

- + Shared post is updated with customized content and format.
- + User is redirected back to the app.

* **Expected Result:** Workout summary is successfully shared with customized content and format.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4: Verify that sharing features are integrated seamlessly with social media.****

* **Test Case ID:** SSHA004

* **Test Case Title:** Verify that sharing features are integrated seamlessly with social media.

* **Test Case Description:** This test case is designed to ensure that the sharing feature in the app is integrated smoothly with social media platforms.

* **Test Suite:** Social Sharing

* **Test Priority:** High

* **Preconditions:**

- + User account is created and logged in.

+ User has completed at least one workout session.

* **Test Data:** None

* **Test Steps:**

1. Log in to the app with a valid user account.
2. Open the workout summary page for a completed workout session.
3. Click the "Share" button to post the workout summary on social media.
4. Observe the sharing process to ensure that it is smooth and seamless.
5. Verify that the shared post is visible on the social media platform.

* **Postconditions:**

+ Workout summary is successfully shared on social media.

+ User is redirected back to the app.

* **Expected Result:** Sharing feature is integrated seamlessly with social media.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that users can track the engagement of their shared posts.

* **Test Case ID:** SSHA005

* **Test Case Title:** Verify that users can track the engagement of their shared posts.

* **Test Case Description:** This test case is designed to ensure that users can view the engagement metrics (e.g., likes, comments, shares) for their shared workout summaries and achievements.

* **Test Suite:** Social Sharing

* **Test Priority:** High

* **Preconditions:**

+ User account is created and logged in.

+ User has completed at least one workout session.

* **Test Data:** None

* **Test Steps:**

1. Log in to the app with a valid user account.
2. Open the workout summary page for a completed workout session.
3. Click the "Share" button to post the workout summary on social media.
4. Wait for the shared post to receive engagement (e.g., likes, comments).
5. Open the user profile or settings page to view the engagement metrics.

* **Postconditions:**

+ User can view engagement metrics for their shared post.

+ User is redirected back to their profile or settings page.

* **Expected Result:** User can track the engagement of their shared post.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

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:

Here are 5 detailed test cases, following the Test Case Creation Template and project requirements, for the Feature "Rest and Recovery Tracking":

****Test Case 1:****

Test Case ID: TCT_RRT-01

Test Case Title: Verify that users can log rest days and recovery activities successfully

Test Case Description: This test case will ensure that users can log their rest days and recovery activities without any issues, testing the core functionality of the feature.

Test Suite: Rest and Recovery Tracking Module

Test Priority: High

Preconditions:

- * User has a registered account with the Fitness Tracker App.
- * The user's profile is set up with the necessary details.

Test Data:

- * User account credentials (username, password)
- * Rest day log with date, activity, and metric data

Test Steps:

1. Log in to the app using valid user credentials.
2. Navigate to the "Rest and Recovery" section.
3. Click on "Log Rest Day" and fill in the required fields (date, activity, and metric data).
4. Verify that the rest day is successfully logged.
5. Repeat the process for logging recovery activities.

Postconditions:

- * User is able to view their logged rest days and recovery activities.
- * The logged data is correctly updated in the user's profile.

Expected Result: The test case will validate that users can successfully log their rest days and recovery activities, verifying the core functionality of the feature.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2:****

Test Case ID: TCT_RRT-02

Test Case Title: Verify that the app tracks metrics such as resting heart rate and sleep quality correctly

Test Case Description: This test case will ensure that the app accurately tracks resting heart rate and sleep quality metrics, as mentioned in the user story.

Test Suite: Rest and Recovery Tracking Module

Test Priority: High

Preconditions:

* User has a registered account with the Fitness Tracker App.

- * The user's profile is set up with the necessary details.
- * User has logged several rest days and recovery activities.

Test Data:

- * User account credentials (username, password)
- * Rest day and recovery activity logs with relevant metric data (HR and sleep quality)

Test Steps:

1. Log in to the app using valid user credentials.
2. Navigate to the "Rest and Recovery" section.
3. View the rest day and recovery activity logs with metric data (HR and sleep quality).
4. Verify that the metrics are correctly tracked and displayed.
5. Perform some mathematical calculations to ensure that the metrics are correctly calculated.

Postconditions:

- * User is able to view their resting heart rate and sleep quality metrics.
- * The metrics are correctly updated in the user's profile.

Expected Result: The test case will validate that the app accurately tracks resting heart rate and sleep quality metrics, verifying the feature's functionality.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3:****

Test Case ID: TCT_RRT-03

Test Case Title: Verify that users receive recommendations for recovery based on their activities

Test Case Description: This test case will ensure that users receive personalized recovery recommendations based on their logged activities and metrics.

Test Suite: Rest and Recovery Tracking Module

Test Priority: Medium

Preconditions:

- * User has a registered account with the Fitness Tracker App.
- * The user's profile is set up with the necessary details.
- * User has logged several rest days and recovery activities.

Test Data:

- * User account credentials (username, password)

* Rest day and recovery activity logs with relevant data

Test Steps:

1. Log in to the app using valid user credentials.
2. Navigate to the "Rest and Recovery" section.
3. View the rest day and recovery activity logs.
4. Verify that the user receives recovery recommendations based on their logged activities and metrics.
5. Engage with the recommendations by accepting or dismissing them.

Postconditions:

- * User is able to view and engage with recovery recommendations.
- * The recommendations are based on the user's logged activities and metrics.

Expected Result: The test case will validate that users receive personalized recovery recommendations, verifying the feature's functionality.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 4:****

Test Case ID: TCT_RRT-04

Test Case Title: Verify that the app provides alerts for insufficient recovery time

Test Case Description: This test case will ensure that the app provides timely alerts for insufficient recovery time, ensuring that users take necessary steps to recover.

Test Suite: Rest and Recovery Tracking Module

Test Priority: High

Preconditions:

- * User has a registered account with the Fitness Tracker App.
- * The user's profile is set up with the necessary details.
- * User has logged several rest days and recovery activities.

Test Data:

- * User account credentials (username, password)
- * Insufficient recovery time scenario (e.g., 2 days of intense exercise with little rest)

Test Steps:

1. Log in to the app using valid user credentials.
2. Navigate to the "Rest and Recovery" section.

3. View the user's activity and recovery log.
4. Verify that the app provides a recovery alert based on insufficient recovery time.
5. Engage with the alert by dismissing or acknowledging it.

Postconditions:

- * User is able to view a recovery alert.
- * The user takes action to recover or dismiss the alert.

Expected Result: The test case will validate that the app provides timely alerts for insufficient recovery time, verifying the feature's functionality.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 5:****

Test Case ID: TCT_RRT-05

Test Case Title: Verify that users can view trends and history of their recovery data

Test Case Description: This test case will ensure that users can access and review their recovery data trends

and history, supporting informed decision-making.

Test Suite: Rest and Recovery Tracking Module

Test Priority: Medium

Preconditions:

- * User has a registered account with the Fitness Tracker App.
- * The user's profile is set up with the necessary details.
- * User has logged several rest days and recovery activities.

Test Data:

- * User account credentials (username, password)
- * Rest day and recovery activity logs with relevant data

Test Steps:

1. Log in to the app using valid user credentials.
2. Navigate to the "Rest and Recovery" section.
3. View the recovery data trend and history chart.
4. Verify that the data is correctly displayed and accessible.
5. Engage with the data by exploring the trend and history views.

Postconditions:

* User is able to view recovery data trends and history.

* The data is accurately displayed and accessible.

Expected Result: The test case will validate that users can access and review their recovery data trends and history, verifying the feature's functionality.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 93

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Audio Coaching

Feature description -

Users can access audio-guided workouts, coaching cues, and motivational messages during their workouts, enhancing engagement and adherence. Audio coaching provides real-time feedback, encouragement, and guidance, simulating the experience of having a personal trainer or workout buddy.

Userstory title -

As a user, I want to access audio coaching during workouts so that I can stay motivated and receive real-time guidance.

Acceptance criteria -

['Users can access a variety of audio-guided workouts.', 'Audio coaching includes motivational messages and coaching cues.', 'Users can choose coaching based on workout type and intensity.', 'Audio coaching integrates seamlessly with other app features.', 'Users can control audio settings and preferences.']

Generated Test Cases:

Here are 5 detailed test cases for the Feature: Audio Coaching:

****Test Case 1: Verify that Users can Access a Variety of Audio-Guided Workouts****

Test Case ID 001
--- ---
Test Case Title Verify that Users can Access a Variety of Audio-Guided Workouts
Test Case Description This test case verifies that users can access a variety of audio-guided workouts as specified in the audio coaching feature.
Test Suite Audio Coaching Feature
Test Priority High
Preconditions User is logged in to the app User has GPS and internet connectivity
Test Data No test data needed
Test Steps Launch the app and navigate to the audio coaching feature Select a workout

type (e.g., cardio, strength training, yoga) Choose an intensity level (e.g., beginner, advanced)
Click on a workout to start audio coaching |

| Postconditions | Workout audio coaching starts and plays as expected |

| Expected Result | The user can access a variety of audio-guided workouts as specified in the audio coaching feature. |

| Severity | Major |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 2: Verify that Audio Coaching Includes Motivational Messages and Coaching Cues****

| Test Case ID | 002 |

| --- | --- |

| Test Case Title | Verify that Audio Coaching Includes Motivational Messages and Coaching Cues |

| Test Case Description | This test case verifies that audio coaching includes motivational messages and coaching cues as specified in the audio coaching feature. |

| Test Suite | Audio Coaching Feature |

| Test Priority | High |

| Preconditions | User is logged in to the app User has started an audio coaching session
 |

| Test Data | No test data needed |

| Test Steps | Start an audio coaching session Verify that motivational messages and coaching cues are provided at regular intervals Take note of the timing and content of the messages |

| Postconditions | Audio coaching session completes as expected |

| Expected Result | Audio coaching includes motivational messages and coaching cues as specified in the

feature. |

| Severity | Major |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 3: Verify that Users can Choose Coaching Based on Workout Type and Intensity****

| Test Case ID | 003 |

| --- | --- |

| Test Case Title | Verify that Users can Choose Coaching Based on Workout Type and Intensity |

| Test Case Description | This test case verifies that users can choose coaching based on workout type and intensity as specified in the audio coaching feature. |

| Test Suite | Audio Coaching Feature |

| Test Priority | Medium |

| Preconditions | User is logged in to the app User has navigated to the audio coaching feature |

| Test Data | No test data needed |

| Test Steps | Launch the app and navigate to the audio coaching feature Select a workout type (e.g., cardio, strength training, yoga) Choose an intensity level (e.g., beginner, advanced) |

| Postconditions | Coaching is selected as expected |

| Expected Result | Users can choose coaching based on workout type and intensity as specified in the feature. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

****Test Case 4: Verify that Audio Coaching Integrates Seamlessly with Other App Features****

Test Case ID 004
--- ---
Test Case Title Verify that Audio Coaching Integrates Seamlessly with Other App Features
Test Case Description This test case verifies that audio coaching integrates seamlessly with other app features as specified in the audio coaching feature.
Test Suite Audio Coaching Feature
Test Priority Medium
Preconditions User is logged in to the app User has GPS and internet connectivity
Test Data No test data needed
Test Steps Start an audio coaching session Verify that heart rate and distance tracking are integrating with the coaching session Take note of any synchronization issues
Postconditions Audio coaching session completes as expected
Expected Result Audio coaching integrates seamlessly with other app features as specified in the feature.
Severity Minor
Priority Medium
Type of Testing Functional Testing
Test Case Approach Positive

****Test Case 5: Verify that Users can Control Audio Settings and Preferences****

Test Case ID 005
--- ---
Test Case Title Verify that Users can Control Audio Settings and Preferences

| Test Case Description | This test case verifies that users can control audio settings and preferences as specified in the audio coaching feature. |

| Test Suite | Audio Coaching Feature |

| Test Priority | Medium |

| Preconditions | User is logged in to the app User has navigated to the audio coaching feature |

| Test Data | No test data needed |

| Test Steps | Launch the app and navigate to the audio coaching feature Go to audio settings and preferences Verify that users can adjust volume, playback control, and other settings |

| Postconditions | User can control audio settings and preferences as expected |

| Expected Result | Users can control audio settings and preferences as specified in the feature. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

Data - 94

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Health Apps

Feature description -

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

Userstory title -

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

Acceptance criteria -

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

Generated Test Cases:

Here are five detailed test cases for the Integration with Health Apps feature of the Fitness Tracker App, following the provided Test Case Creation Template:

****Test Case 1: Integration with Popular Health and Wellness Apps****

*** **Test Case ID:** FTTTR-001**

*** **Test Case Title:** Verify that the fitness tracker app integrates with popular health and wellness apps**

*** **Test Case Description:**** This test case verifies that the app successfully integrates with popular health and wellness apps, such as calorie counters, meditation apps, and mental health trackers. The test case checks if the app can consolidate data from these apps and provide a comprehensive view of the user's overall well-being.

*** **Test Suite:** HealthAppIntegration**

* **Test Priority:** High

* **Preconditions:** The app is installed on a test device with internet connectivity. The test user has a valid account on multiple health and wellness apps.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a test user account.
2. Navigate to the integration settings page.
3. Enable integration with multiple health and wellness apps (e.g., MyFitnessPal, Headspace, Calm).
4. Allow the app to access the health data from these apps.
5. Verify that the app consolidates the health data from the integrated apps.

* **Postconditions:** The app successfully integrates with the selected health and wellness apps.

* **Expected Result:** The app displays a consolidated view of the user's health data, including metrics such as steps taken, calories burned, distance traveled, and heart rate.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Acceptance Testing

* **Test Case Approach:** Positive

Test Case 2: Syncing Data from Connected Health Apps

* **Test Case ID:** FTTTR-002

* **Test Case Title:** Verify that users can sync data from connected health apps

* **Test Case Description:** This test case verifies that users can successfully sync data from connected health and wellness apps. The test case checks if the app updates the user's health metrics in real-time after syncing data from these apps.

* **Test Suite:** HealthAppIntegration

* **Test Priority:** High

* **Preconditions:** The app is installed on a test device with internet connectivity. The test user has a valid account on multiple health and wellness apps.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a test user account.
2. Connect multiple health and wellness apps to the app.
3. Record the initial health metrics (e.g., steps taken, calories burned).
4. Trigger a sync operation from one of the connected health apps.
5. Verify that the app updates the user's health metrics in real-time.

* **Postconditions:** The app successfully updates the user's health metrics after syncing data from the connected health app.

* **Expected Result:** The app displays the updated health metrics in real-time.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Acceptance Testing

* **Test Case Approach:** Positive

****Test Case 3: Consolidated View of Health Metrics****

* **Test Case ID:** FTTTR-003

* **Test Case Title:** Verify that the app provides a consolidated view of health metrics

* **Test Case Description:** This test case verifies that the app displays a comprehensive view of the user's health metrics, including data from multiple health and wellness apps.

* **Test Suite:** HealthAppIntegration

* **Test Priority:** High

* **Preconditions:** The app is installed on a test device with internet connectivity. The test user has a valid account on multiple health and wellness apps.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a test user account.
2. Connect multiple health and wellness apps to the app.
3. Verify that the app displays a consolidated view of the user's health metrics.
4. Check that the app includes metrics from multiple health and wellness apps.

* **Postconditions:** The app displays a comprehensive view of the user's health metrics.

* **Expected Result:** The app shows a consolidated view of the user's health metrics, including data from multiple health and wellness apps.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Acceptance Testing

* **Test Case Approach:** Positive

****Test Case 4: Insights Based on Combined Data****

* **Test Case ID:** FTTTR-004

* **Test Case Title:** Verify that users receive insights based on combined data from multiple apps

* **Test Case Description:** This test case verifies that the app provides actionable insights to users based on the combined data from multiple health and wellness apps.

* **Test Suite:** HealthAppIntegration

* **Test Priority:** Medium

* **Preconditions:** The app is installed on a test device with internet connectivity. The test user has a valid account on multiple health and wellness apps.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a test user account.

2. Connect multiple health and wellness apps to the app.
3. Enable the insight feature in the app.
4. Verify that the app provides actionable insights based on the combined data.

* **Postconditions:** The app provides actionable insights to the user.

* **Expected Result:** The app displays actionable insights based on the combined data from multiple health and wellness apps.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Acceptance Testing

* **Test Case Approach:** Positive

Test Case 5: Easy Integration and Configuration

* **Test Case ID:** FTTTR-005

* **Test Case Title:** Verify that integration settings are easy to manage and configure

* **Test Case Description:** This test case verifies that the app provides an intuitive and user-friendly interface for integrating with health and wellness apps.

* **Test Suite:** HealthAppIntegration

* **Test Priority:** High

* **Preconditions:** The app is installed on a test device with internet connectivity.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a test user account.
2. Navigate to the integration settings page.
3. Verify that the app provides an intuitive interface for integrating with health and wellness apps.
4. Check that the app allows users to easily configure integration settings.

* **Postconditions:** The app provides an intuitive and user-friendly interface for integrating with health and

wellness apps.

* **Expected Result:** The app displays an intuitive interface for integrating with health and wellness apps.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Acceptance Testing

* **Test Case Approach:** Positive

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:

Here are five detailed test cases for the Body Composition Tracking feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that users can log body composition metrics****

* Test Case ID: BCT-001

* Test Case Title: Verify that users can log body composition metrics such as weight, body fat percentage, and muscle mass.

* Test Case Description: This test case ensures that users can enter and save body composition metrics such as weight, body fat percentage, and muscle mass, which will then be used for calculating BMI and displaying trends.

* Test Suite: Body Composition

* Test Priority: High

* Preconditions:

+ User has a registered account

+ User has navigated to the body composition tracking feature

* Test Data:

+ User input data (weight, body fat percentage, muscle mass)

* Test Steps:

1. Open the app and navigate to the body composition tracking feature

2. Enter valid user input data (weight, body fat percentage, muscle mass) into the respective fields

3. Click the "Save" button to save the entered data

4. Verify that the data is saved and displayed on the body composition tracking screen

* Postconditions:

+ Data is saved correctly

* Expected Result: User input data is saved and displayed correctly.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

Test Case 2: Verify that the app calculates and displays BMI based on logged data

* Test Case ID: BCT-002

* Test Case Title: Verify that the app calculates and displays BMI based on logged body composition data.

* Test Case Description: This test case ensures that the app correctly calculates and displays BMI based on the logged body composition metrics.

* Test Suite: Body Composition

* Test Priority: High

* Preconditions:

+ User has logged body composition metrics (weight, body fat percentage, muscle mass)

+ User has navigated to the body composition tracking feature

* Test Data:

+ User input data (weight, body fat percentage, muscle mass)

* Test Steps:

1. Open the app and navigate to the body composition tracking feature

2. Enter valid user input data (weight, body fat percentage, muscle mass) into the respective fields

3. Click the "Save" button to save the entered data

4. Verify that the app correctly calculates and displays BMI

* Postconditions:

+ BMI is calculated correctly and displayed on the body composition tracking screen

* Expected Result: BMI is calculated correctly and displayed accurately.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Verify that users can view trends and history of body composition changes****

* Test Case ID: BCT-003

* Test Case Title: Verify that users can view trends and history of body composition changes.

* Test Case Description: This test case ensures that users can view trends and history of body composition changes, including weight, body fat percentage, and muscle mass.

* Test Suite: Body Composition

* Test Priority: Medium

* Preconditions:

+ User has logged body composition metrics (weight, body fat percentage, muscle mass)

+ User has navigated to the body composition tracking feature

* Test Data:

+ User input data (weight, body fat percentage, muscle mass)

* Test Steps:

1. Open the app and navigate to the body composition tracking feature
2. Enter valid user input data (weight, body fat percentage, muscle mass) into the respective fields
3. Click the "Save" button to save the entered data
4. Verify that the app displays the correct trends and history of body composition changes

* Postconditions:

+ Trends and history of body composition changes are displayed correctly

* Expected Result: Trends and history of body composition changes are displayed accurately.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 4: Verify that the app provides insights and recommendations based on body composition data****

* Test Case ID: BCT-004

* Test Case Title: Verify that the app provides insights and recommendations based on body composition data.

* Test Case Description: This test case ensures that the app provides actionable insights and recommendations based on the logged body composition metrics.

* Test Suite: Body Composition

* Test Priority: High

* Preconditions:

+ User has logged body composition metrics (weight, body fat percentage, muscle mass)

+ User has navigated to the body composition tracking feature

* Test Data:

+ User input data (weight, body fat percentage, muscle mass)

* Test Steps:

1. Open the app and navigate to the body composition tracking feature

2. Enter valid user input data (weight, body fat percentage, muscle mass) into the respective fields

3. Click the "Save" button to save the entered data

4. Verify that the app provides insights and recommendations based on the logged body composition metrics

* Postconditions:

- + Insights and recommendations are provided correctly

- * Expected Result: Actionable insights and recommendations are provided based on the logged body composition metrics.

- * Severity: Critical

- * Priority: High

- * Type of Testing: Functional Testing

- * Test Case Approach: Positive

****Test Case 5: Verify that users can set goals for body composition metrics****

- * Test Case ID: BCT-005

- * Test Case Title: Verify that users can set goals for body composition metrics.

- * Test Case Description: This test case ensures that users can set realistic and achievable goals for body composition metrics such as weight, body fat percentage, and muscle mass.

- * Test Suite: Body Composition

- * Test Priority: Medium

- * Preconditions:

- + User has navigated to the body composition tracking feature

- * Test Data:

- + User input data (target weight, target body fat percentage, target muscle mass)

- * Test Steps:

1. Open the app and navigate to the body composition tracking feature

2. Enter valid user input data (target weight, target body fat percentage, target muscle mass) into the respective fields

3. Click the "Save" button to save the goals

4. Verify that the app displays the set goals and provides a mechanism for tracking progress

- * Postconditions:

- + Goals are set correctly and displayed on the body composition tracking screen
- * Expected Result: Users can set realistic and achievable goals for body composition metrics, and the app provides a mechanism for tracking progress towards the set goals.
- * Severity: Major
- * Priority: Medium
- * Type of Testing: Functional Testing
- * Test Case Approach: Positive

In summary, these five test cases cover the key acceptance criteria for the Body Composition Tracking feature, ensuring that users can log body composition metrics, view trends and history, receive actionable insights and recommendations, and set realistic goals for body composition metrics. These test cases focus on ensuring that the app accurately calculates and displays BMI, provides accurate trends and history, and offers actionable insights and recommendations based on body composition data.

Data - 96

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Analysis

Feature description -

The app offers detailed workout analysis tools, including performance metrics, trends, and insights into exercise intensity, duration, and calorie burn. Workout analysis helps users assess the effectiveness of their workouts, identify areas for improvement, and optimize training strategies for better results.

Userstory title -

As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

Acceptance criteria -

['Users can view detailed performance metrics for each workout.', 'The app provides trends and insights into workout intensity and duration.', 'Users can track calorie burn and other health metrics.', 'The app offers recommendations based on workout analysis.', 'Users can compare current workouts with past performance.']

Generated Test Cases:

Here are the 5 test cases for the Workout Analysis feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that users can view detailed performance metrics for each workout****

*** **Test Case ID:** WTA-001**

*** **Test Case Title:** Verify that users can view detailed performance metrics for each workout**

*** **Test Case Description:** This test case verifies that users can view detailed performance metrics, including exercise intensity, duration, and calorie burn, for each workout logged in the app.**

*** **Test Suite:** Workout Analysis Features**

*** **Test Priority:** High**

*** **Preconditions:** User has created an account, logged in to the app, and has at least one workout logged.**

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the app and navigate to the workout log.

2. Select a recent workout from the log.
3. Verify that the performance metrics, including exercise intensity, duration, and calorie burn, are displayed.

* **Postconditions:** None

* **Expected Result:** Users can view detailed performance metrics for each workout.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that the app provides trends and insights into workout intensity and duration

* **Test Case ID:** WTA-002

* **Test Case Title:** Verify that the app provides trends and insights into workout intensity and duration

* **Test Case Description:** This test case verifies that the app provides trends and insights into workout intensity and duration over time, allowing users to identify patterns and areas for improvement.

* **Test Suite:** Workout Analysis Features

* **Test Priority:** High

* **Preconditions:** User has created an account, logged in to the app, and has at least three workouts logged with varying intensity and duration.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the workout log.
2. Select a time period (e.g., week, month) to view trends and insights.
3. Verify that the app displays trends and insights into workout intensity and duration.

* **Postconditions:** None

* **Expected Result:** The app provides trends and insights into workout intensity and duration over time.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that users can track calorie burn and other health metrics

* **Test Case ID:** WTA-003

* **Test Case Title:** Verify that users can track calorie burn and other health metrics

* **Test Case Description:** This test case verifies that users can track calorie burn and other health metrics, such as steps taken and distance traveled, for each workout.

* **Test Suite:** Workout Analysis Features

* **Test Priority:** Medium

* **Preconditions:** User has created an account, logged in to the app, and has at least one workout logged.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the workout log.
2. Select a workout from the log.
3. Verify that calorie burn and other health metrics are displayed.

* **Postconditions:** None

* **Expected Result:** Users can track calorie burn and other health metrics for each workout.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that the app offers recommendations based on workout analysis

* **Test Case ID:** WTA-004

* **Test Case Title:** Verify that the app offers recommendations based on workout analysis

* **Test Case Description:** This test case verifies that the app offers personalized recommendations for improving workouts based on analysis of user performance and trends.

* **Test Suite:** Workout Analysis Features

* **Test Priority:** High

* **Preconditions:** User has created an account, logged in to the app, and has at least three workouts logged with varying intensity and duration.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the workout log.
2. Select a time period (e.g., week, month) to view trends and insights.
3. Verify that the app displays recommendations for improving workouts.

* **Postconditions:** None

* **Expected Result:** The app offers personalized recommendations for improving workouts.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that users can compare current workouts with past performance

* **Test Case ID:** WTA-005

* **Test Case Title:** Verify that users can compare current workouts with past performance

* **Test Case Description:** This test case verifies that users can compare their current workouts with past performance, allowing them to assess progress and identify areas for improvement.

* **Test Suite:** Workout Analysis Features

* **Test Priority:** High

* **Preconditions:** User has created an account, logged in to the app, and has at least three workouts logged with varying intensity and duration.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the workout log.
2. Select a recent workout from the log.
3. Verify that the app allows users to compare the current workout with past performance.

* **Postconditions:** None

* **Expected Result:** Users can compare current workouts with past performance.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

Data - 97

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Weather Integration

Feature description -

Integrated weather forecasts and conditions allow users to plan outdoor workouts and activities based on current weather conditions and forecasts. Weather integration helps users make informed decisions, stay safe, and optimize their outdoor exercise routines according to weather conditions.

Userstory title -

As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

Acceptance criteria -

['The app provides current weather conditions and forecasts.', 'Users can view weather details for different locations.', 'Weather data is updated in real-time.', 'The app offers recommendations based on weather conditions.', 'Users can receive weather alerts for extreme conditions.']

Generated Test Cases:

I will create detailed test cases covering all the acceptance criteria for the "Weather Integration" feature of the "Fitness Tracker App". I will use the provided Test Case Creation Template to ensure that each test case is clear, concise, and relevant to the project context and user story.

Test Case 1: Verify that the app provides current weather conditions.

****Test Case ID:**** WC-001

****Test Case Title:**** Verify that the app provides current weather conditions.

****Test Case Description:**** This test case verifies that the app displays the current weather conditions for the user's location, ensuring that the user gets accurate and timely information to plan their workouts accordingly.

****Test Suite:**** Weather Integration

****Test Priority:**** High

****Preconditions:****

* User has location services enabled on their device.

* User has installed the app and has logged in.

****Test Data:**** Default location (user's device location).

****Test Steps:****

1. Launch the app and navigate to the weather section.
2. Verify that the current weather conditions display prominently.
3. Check that the displayed weather conditions match the expected conditions for the user's location (temperature, conditions, etc.).
4. Test different weather conditions (e.g., sunny, rainy, cloudy) to ensure the app updates accordingly.

****Postconditions:**** none

****Expected Result:**** Current weather conditions are displayed prominently on the app's main page.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive, Unit Testing

Test Case 2: Verify that users can view weather details for different locations.

****Test Case ID:**** WC-002

****Test Case Title:**** Verify that users can view weather details for different locations.

****Test Case Description:**** This test case tests the app's ability to display weather forecasts for multiple locations, allowing users to plan their workouts in various cities or areas.

****Test Suite:**** Weather Integration

****Test Priority:**** Medium

****Preconditions:****

* User has installed the app and has logged in.

****Test Data:**** List of multiple locations (cities) with known weather conditions.

****Test Steps:****

1. Launch the app and navigate to the weather section.

2. Change the location to a different city from the list.
3. Verify that the weather details (current and forecast) update accordingly for the new location.
4. Test multiple locations to ensure the app handles cross-location information display correctly.

****Postconditions:**** The app updates to reflect the new location's weather.

****Expected Result:**** Users can view weather forecasts for different locations.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Integration Testing

****Test Case Approach:**** Positive

Test Case 3: Verify that weather data is updated in real-time.

****Test Case ID:**** WC-003

****Test Case Title:**** Verify that weather data is updated in real-time.

****Test Case Description:**** This test case ensures that the app updates weather conditions in real-time to keep users informed about the latest weather conditions and forecasts.

****Test Suite:**** Weather Integration

****Test Priority:**** High

****Preconditions:****

* User has the app running in the background.

****Test Data:**** Weather updates received from external APIs.

****Test Steps:****

1. Launch the app while also accessing the app's settings to observe real-time updates.
2. Change the location to a city with frequently changing weather.
3. Verify that the weather app updates accurately and quickly in response to changes in the actual weather.
4. Set a timer and measure time intervals between updates to ensure real-time refreshment is observed.

****Postconditions:**** The app reflects the latest weather updates.

****Expected Result:**** Weather data is updated in real-time to ensure users have the most accurate information.

****Severity:**** Blocker

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive, Data-Driven Testing

Test Case 4: Verify that the app offers personalized recommendations based on weather conditions.

****Test Case ID:**** WC-004

****Test Case Title:**** Verify that the app offers personalized recommendations based on weather conditions.

****Test Case Description:**** This test case validates that the app gives users workout or activity recommendations tailored to the weather conditions they are in, increasing the effectiveness of workouts.

****Test Suite:**** Weather Integration

****Test Priority:**** Medium

****Preconditions:****

* User has performed workout history logged into the app.

****Test Data:**** Different weather conditions and user preferences.

****Test Steps:****

1. Set user preferences and goals within the app.
2. Simulate various weather conditions and verify the app generates personalized recommendations.
3. Test the recommendations for both general weather forecasts and emergency alerts.
4. Ensure the recommendations are logical and offer real value to the user's fitness goals and weather condition.

****Postconditions:**** The app displays tailored recommendations based on weather.

****Expected Result:**** The app provides actionable recommendations based on current or forecasted weather conditions.

****Severity:**** Major

****Priority:** Medium**

****Type of Testing:** Integration Testing**

****Test Case Approach:** Positive, Exploratory Testing**

Test Case 5: Verify that users can receive weather alerts for extreme conditions.

****Test Case ID:** WC-005**

****Test Case Title:** Verify that users can receive weather alerts for extreme conditions.**

****Test Case Description:** This test case ensures users can receive urgent weather alerts when extreme conditions are forecasted, helping them stay safe and adjust workout plans accordingly.**

****Test Suite:** Weather Integration**

****Test Priority:** High**

****Preconditions:****

* User has notifications enabled on their device.

****Test Data:** Weather data indicating extreme conditions (e.g., heavy rain, high winds, thunderstorms).**

****Test Steps:****

1. Set up push notifications for extreme weather conditions within the app.
2. Test pushing the app to simulate extreme weather alerts.
3. Observe how the app displays and handles extreme weather alerts (sound, on-screen prompt, email push).
4. Test turning off and on notifications to verify that the app remembers the settings and behaves correctly.

****Postconditions:** The user receives weather alerts for extreme conditions.**

****Expected Result:** Users receive timely and actionable alerts for significant weather conditions to inform their safety and workout plans.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive, Performance Testing**

Each test case meticulously addresses one or more of the acceptance criteria for the Weather Integration feature, emphasizing the importance of timely and accurate weather information in planning workouts safely. This comprehensive approach helps in ensuring the quality of the Fitness Tracker App by identifying and addressing potential defects early in the development cycle.

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:

Here are the 5 detailed test cases for the "Mood Tracking" feature of the Fitness Tracker App:

****Test Case 1: Verify that users can log their mood, energy levels, and motivation****

*** **Test Case ID:** MVT-001**

*** **Test Case Title:** Verify that users can log their mood, energy levels, and motivation**

*** **Test Case Description:**** This test case verifies that users can successfully log their mood, energy levels, and motivation before and after workouts. It also checks that the app provides a drop-down menu for selecting mood, energy levels, and motivation, and that invalid input is handled correctly.

*** **Test Suite:** Mood Tracking Module**

*** **Test Priority:** High**

*** **Preconditions:****

+ User is logged in to the app

+ User has a workout scheduled

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the app and schedule a workout
2. Open the "Mood Tracking" section and select a date and time
3. Enter a valid mood, energy level, and motivation
4. Submit the form and verify that the data is saved correctly
5. Attempt to log invalid data (e.g. negative mood, invalid energy level)

*** **Postconditions:****

- + The app displays a confirmation message that the data has been saved
- + The user's mood, energy levels, and motivation are reflected in the app's dashboard
- * **Expected Result:** The user can successfully log their mood, energy levels, and motivation
- * **Severity:** Major
- * **Priority:** High
- * **Type of Testing:** Functional Testing
- * **Test Case Approach:** Positive

****Test Case 2: Verify that the app provides insights into mood and energy trends over time****

- * **Test Case ID:** MVT-002
- * **Test Case Title:** Verify that the app provides insights into mood and energy trends over time
- * **Test Case Description:** This test case verifies that the app displays a chart or graph showing mood and energy levels over time, and that the data is accurate and up-to-date.
- * **Test Suite:** Mood Tracking Module
- * **Test Priority:** High
- * **Preconditions:**
 - + User has logged multiple workouts with mood and energy data
- * **Test Data:** No test data needed
- * **Test Steps:**

1. Log in to the app and navigate to the "Mood Tracking" section
2. Select a date range to view mood and energy data
3. Verify that the app displays a chart or graph showing the trend
4. Check that the data is accurate and up-to-date

- * **Postconditions:**
 - + The app displays a chart or graph showing mood and energy levels over time
- * **Expected Result:** The app provides accurate insights into mood and energy trends over time

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

****Test Case 3: Verify that users receive recommendations based on their mood and energy data****

* **Test Case ID:** MVT-003

* **Test Case Title:** Verify that users receive recommendations based on their mood and energy data

* **Test Case Description:** This test case verifies that users receive personalized recommendations based on their mood and energy data, such as suggesting rest days or modifying workout intensity.

* **Test Suite:** Mood Tracking Module

* **Test Priority:** High

* **Preconditions:**

+ User has logged multiple workouts with mood and energy data

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the "Mood Tracking" section
2. Select a date range to view mood and energy data
3. Verify that the app provides a list of personalized recommendations
4. Check that the recommendations are accurate and tailored to the user's mood and energy levels

* **Postconditions:**

+ The app displays a list of personalized recommendations

* **Expected Result:** The app provides accurate and personalized recommendations based on mood and energy data

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Systems Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that the app supports journaling for personal reflections on mood and workouts

* **Test Case ID:** MVT-004

* **Test Case Title:** Verify that the app supports journaling for personal reflections on mood and workouts

* **Test Case Description:** This test case verifies that the app allows users to add journal entries for workouts and mood tracking, and that the entries are saved correctly.

* **Test Suite:** Mood Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in to the app

+ User has a workout scheduled

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the "Mood Tracking" section

2. Open the journal section and add a new entry

3. Verify that the entry is saved correctly

4. Check that the journal entries are listed correctly

* **Postconditions:**

+ The app displays a list of journal entries

* **Expected Result:** The app supports journaling for personal reflections on mood and workouts

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 5: Verify that users can set goals for improving mood and energy levels****

*** **Test Case ID:** MVT-005**

*** **Test Case Title:** Verify that users can set goals for improving mood and energy levels**

*** **Test Case Description:** This test case verifies that users can set goals for improving mood and energy levels, and that the app provides a plan for achieving those goals.**

*** **Test Suite:** Mood Tracking Module**

*** **Test Priority:** Medium**

*** **Preconditions:****

+ User is logged in to the app

*** **Test Data:** No test data needed**

*** **Test Steps:****

1. Log in to the app and navigate to the "Mood Tracking" section
2. Open the goal-setting section and select "Mood" or "Energy" as the goal
3. Set a target value for the goal
4. Verify that the app provides a plan for achieving the goal

*** **Postconditions:****

+ The app displays a plan for achieving the goal

*** **Expected Result:** The app allows users to set goals for improving mood and energy levels**

*** **Severity:** Minor**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Creation

Feature description -

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

Userstory title -

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

Acceptance criteria -

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', 'Recipes are saved in the user's profile.', 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

Generated Test Cases:

Here are the test cases for the Recipe Creation feature based on the provided acceptance criteria:

****Test Case 1: Verify that users can input and save ingredients and cooking instructions****

Test Case ID: REC-001

Test Case Title: Verify that users can input and save ingredients and cooking instructions

Test Case Description: This test case covers the acceptance criteria "Users can input and save ingredients and cooking instructions" to ensure that users can provide accurate recipe details. This includes checking that users can enter ingredients and cooking instructions correctly, save them successfully, and retrieve the saved recipe information.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- * User has an account and is logged in
- * Recipe form is open for input

Test Data: Valid ingredient names, cooking instructions

Test Steps:

1. Log in as a registered user
2. Click on the "Create New Recipe" button
3. Fill in the ingredient name field with valid input (e.g., "Tomatoes")
4. Fill in the cooking instruction field with valid input (e.g., "Heat the tomatoes in a pan over medium heat.")
5. Click on the "Save Recipe" button
6. Verify that the ingredient and cooking instruction lines are saved in the recipe

Postconditions:

- * Recipe is saved in the user's profile
- * User can view the saved recipe

Expected Result: The ingredient and cooking instruction lines are correctly saved in the recipe.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 2: Verify that users can upload photos for their recipes (optional)****

Test Case ID: REC-002

Test Case Title: Verify that users can upload photos for their recipes (optional)

Test Case Description: This test case tests the optional feature of uploading photos for recipes, ensuring that users can attach relevant images to enhance their creations. It covers the acceptance criterion "Users can upload photos for their recipes."

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- * User has an account and is logged in
- * Recipe form is open for input

Test Data: Valid image file (e.g., JPEG or PNG)

Test Steps:

1. Log in as a registered user
2. Click on the "Create New Recipe" button
3. Select a valid image file from the user's device
4. Click on the "Upload" button
5. Verify that the photo is successfully uploaded and attached to the recipe
6. Check the uploaded image's visibility and quality

Postconditions:

- * Recipe contains the attached photo
- * User can view the uploaded image

Expected Result: The photo is correctly uploaded and attached to the recipe.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

****Test Case 3: Verify that recipes are saved in the user's profile****

Test Case ID: REC-003

Test Case Title: Verify that recipes are saved in the user's profile

Test Case Description: This test case focuses on the acceptance criterion "Recipes are saved in the user's profile." It ensures that once users create and save a recipe, the recipe becomes visible in their profile.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- * User has created and saved a recipe

Test Data: None

Test Steps:

1. Log in as a registered user
2. Navigate to the "My Recipes" section
3. Verify that the created recipe is listed in the user's recipe list
4. Check the saved recipe's details (e.g., ingredients, cooking instructions)

Postconditions:

- * Recipe saved in the user's profile is correctly updated

Expected Result: The saved recipe appears in the user's profile.

Severity: Major

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 4: Verify that recipes are accessible to other users on the platform****

Test Case ID: REC-004

Test Case Title: Verify that recipes are accessible to other users on the platform

Test Case Description: This test case ensures that users have access to recipes created by other users, which aligns with the acceptance criterion "Recipes are accessible to other users on the platform."

Test Suite: Recipe Sharing

Test Priority: High

Preconditions:

- * Other user has created and saved a recipe
- * User is logged in

Test Data: None

Test Steps:

1. Log in as a different registered user
2. Search for a recipe created by another user
3. Click on the searched recipe's thumbnail or link
4. Verify that the recipe appears in the correct format (e.g., ingredients, cooking instructions, images)
5. Check public visibility and access levels for the searched recipe

Postconditions:

- * Other users can access the searched recipe correctly

Expected Result: Other users can view the recipe as intended.

Severity: Major

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

****Test Case 5: Verify that users receive a confirmation upon successful recipe submission****

Test Case ID: REC-005

Test Case Title: Verify that users receive a confirmation upon successful recipe submission

Test Case Description: This test case confirms that users receive notification or feedback after submitting their recipe successfully. This is in accordance with the acceptance criterion "Users receive a confirmation upon successful recipe submission."

Test Suite: Recipe Submission

Test Priority: High

Preconditions:

- * User has submitted a new recipe

Test Data: None

Test Steps:

1. Log in as a registered user
2. Click on the "Submit Recipe" button
3. Verify that the system provides confirmation upon successful submission (e.g., a pop-up or success message)
4. Check that the user is notified via email (if email notification is configured)

Postconditions:

- * User is confirmed with the successful submission of the recipe
- * Confirmation methods work correctly (if implemented)

Expected Result: User receives the desired confirmation method(s) after successful recipe submission.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Note: This template may require minor adjustments according to specific requirements of your software development life cycle (SDLC). Please tailor this structure according to your company's QA practices.

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:

Here are five detailed test cases for the Recipe Discovery feature, following the provided Test Case Creation Template:

****Test Case 1: Verify that Users Can Search for Recipes Using Keywords and Ingredients****

****Test Case ID:** RC-001**

****Test Case Title:** Verify that Users Can Search for Recipes Using Keywords and Ingredients**

****Test Case Description:**** As a user, I want to search for recipes using keywords and ingredients, so that I can find new recipes to try. This test case ensures that users can successfully search for recipes using keywords and ingredients.

****Test Suite:** Recipe Discovery**

****Test Priority:** High**

****Preconditions:****

* User is logged in to the platform

* User has a basic profile set up

* There are at least 10 recipes available on the platform

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the platform using valid credentials
2. Navigate to the search bar
3. Enter a keyword (e.g. "chicken") and an ingredient (e.g. "onion") in the search fields
4. Click the search button
5. Verify that the search results display relevant recipes that match the search criteria
6. Verify that the search results include recipes with the keyword and ingredient specified

****Postconditions:**** None

****Expected Result:**** Search results display relevant recipes that match the search criteria.

****Severity:**** Critical

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case 2: Verify that Search Results are Relevant and Accurately Match the Search Criteria****

****Test Case ID:** RC-002**

****Test Case Title:** Verify that Search Results are Relevant and Accurately Match the Search Criteria**

****Test Case Description:** As a user, I want to ensure that search results are relevant and accurately match the search criteria, so that I can find new recipes to try. This test case ensures that search results are accurate and relevant.**

****Test Suite:** Recipe Discovery**

****Test Priority:** High**

****Preconditions:****

* User is logged in to the platform

* User has a basic profile set up

* There are at least 10 recipes available on the platform

****Test Data:** No test data needed**

****Test Steps:****

1. Log in to the platform using valid credentials
2. Navigate to the search bar
3. Enter a keyword (e.g. "chicken") and an ingredient (e.g. "onion") in the search fields
4. Click the search button
5. Verify that the search results display at least 5 recipes that match the search criteria
6. Verify that the search results do not include recipes that do not match the search criteria

****Postconditions:**** None

****Expected Result:**** Search results are accurate and relevant.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 3: Verify that Users Can Filter Search Results by Cuisine Type, Dietary Preferences, and Other Criteria****

****Test Case ID:**** RC-003

****Test Case Title:**** Verify that Users Can Filter Search Results by Cuisine Type, Dietary Preferences, and Other Criteria

****Test Case Description:**** As a user, I want to filter search results by cuisine type, dietary preferences, and other criteria, so that I can find recipes that fit my preferences. This test case ensures that users can successfully filter search results.

****Test Suite:**** Recipe Discovery

****Test Priority:**** Medium

****Preconditions:****

* User is logged in to the platform

* User has a basic profile set up

* There are at least 10 recipes available on the platform

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the platform using valid credentials

2. Navigate to the search bar
3. Enter a keyword (e.g. "chicken") and an ingredient (e.g. "onion") in the search fields
4. Click the search button
5. Select a cuisine type (e.g. "Italian") from the filter options
6. Verify that the search results display only recipes that match the selected cuisine type
7. Select a dietary preference (e.g. "Vegetarian") from the filter options
8. Verify that the search results display only recipes that match the selected dietary preference

****Postconditions:**** None

****Expected Result:**** Search results are filtered correctly based on selected criteria.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case 4: Verify that Recipes in the Search Results are Accessible and Detailed****

****Test Case ID:**** RC-004

****Test Case Title:**** Verify that Recipes in the Search Results are Accessible and Detailed

****Test Case Description:**** As a user, I want to ensure that recipes in the search results are accessible and detailed, so that I can view and save recipes easily. This test case ensures that recipes in the search results are accessible and detailed.

****Test Suite:**** Recipe Discovery

****Test Priority:**** Medium

****Preconditions:****

* User is logged in to the platform

* User has a basic profile set up

* There are at least 10 recipes available on the platform

Test Data: No test data needed

Test Steps:

1. Log in to the platform using valid credentials
2. Navigate to the search bar
3. Enter a keyword (e.g. "chicken") and an ingredient (e.g. "onion") in the search fields
4. Click the search button
5. Click on a recipe in the search results
6. Verify that the recipe details are displayed correctly, including ingredients, instructions, and tags
7. Verify that the recipe is accessible and can be viewed without errors

Postconditions: None

Expected Result: Recipes in the search results are accessible and detailed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5: Verify that Users Can Save Search Results for Future Reference

Test Case ID: RC-005

Test Case Title: Verify that Users Can Save Search Results for Future Reference

Test Case Description: As a user, I want to save search results for future reference, so that I can easily access my favorite recipes. This test case ensures that users can successfully save search results.

Test Suite: Recipe Discovery

Test Priority: Low

Preconditions:

* User is logged in to the platform

* User has a basic profile set up

* There are at least 10 recipes available on the platform

****Test Data:**** No test data needed

****Test Steps:****

1. Log in to the platform using valid credentials

2. Navigate to the search bar

3. Enter a keyword (e.g. "chicken") and an ingredient (e.g. "onion") in the search fields

4. Click the search button

5. Click the "Save" button next to a recipe in the search results

6. Verify that the recipe is saved to the user's favorites

7. Verify that the saved recipe can be viewed in the user's favorites list

****Postconditions:**** None

****Expected Result:**** Search results can be saved for future reference.

****Severity:**** Minor

****Priority:**** Low

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive
