

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

Data - 1

Project name -

Social Media Platform

Project description -

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

Feature name -

User Profiles

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for User Profile Creation

Here are 5 detailed test cases for the "User Profile Creation" feature, adhering to the provided template and covering all acceptance criteria:

****Test Case 1****

* **Test Case ID:** TC_UP_001

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

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

* **Test Suite:** User Profiles

* **Test Priority:** High

* **Preconditions:**

- * A valid user account exists.

- * The user is logged in.

* **Test Data:** Various valid names (e.g., "John Doe", "Jane Smith", "XYZ123").

* **Test Steps:**

1. Navigate to the user profile editing page.
2. Enter a valid name into the "Name" field.
3. Click the "Save" button.
4. Verify that the entered name is displayed correctly on the profile page.
5. Modify the name in the editing section.
6. Click "Save".
7. Verify that the updated name is now displayed on the profile page.

* **Postconditions:** No specific postconditions.

* **Expected Result:** The user's name should be added and edited successfully, reflecting the changes on their profile page.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case 2

* **Test Case ID:** TC_UP_002

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

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

* **Test Suite:** User Profiles

* **Test Priority:** High

* **Preconditions:**

- * A valid user account exists.

- * The user is logged in.

* **Test Data:** Varied sample bios (e.g., short text, long text, including emojis, special characters).

* **Test Steps:**

1. Navigate to the user profile editing page.
2. Enter a valid bio into the designated field.
3. Click the "Save" button.
4. Verify that the entered bio is displayed correctly on the profile page.
5. Modify the bio in the editing section.
6. Click "Save".
7. Verify that the updated bio is now displayed on the profile page.

* **Postconditions:** No specific postconditions.

* **Expected Result:** The user's bio should be added and edited successfully, reflecting the changes on their profile page.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case 3

* **Test Case ID:** TC_UP_003

* **Test Case Title:** Verify that users can upload a profile picture in supported formats.

* **Test Case Description:** This test case verifies that users can upload profile pictures in acceptable image formats and that unsupported formats are rejected.

* **Test Suite:** User Profiles

* **Test Priority:** High

* **Preconditions:**

- * A valid user account exists.

- * The user is logged in.

- * Access to a local device with sample images.

* **Test Data:**

- * Images in supported formats (JPEG, PNG, etc.)

- * Images in unsupported formats (GIF, PDF, etc.)

* **Test Steps:**

1. Navigate to the user profile editing page.
2. Click on the profile picture upload area.
3. Select a supported format image from the device.
4. Verify the image uploads successfully and displays on the profile preview.
5. Repeat steps 2-4 but select an unsupported format image.
6. Verify an error message is displayed indicating the format is not allowed.

* **Postconditions:** No specific postconditions.

* **Expected Result:** Supported image formats should upload successfully, while unsupported formats should be rejected with an appropriate error message.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional and Integration Testing

****Test Case 4****

* **Test Case ID:** TC_UP_004

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

* **Test Case Description:** This test case verifies that users have granular control over the privacy of their profile information.

* **Test Suite:** User Profiles

* **Test Priority:** Medium

* **Preconditions:**

- * A valid user account exists.

- * The user is logged in.

* **Test Data:** Various combinations of privacy settings for different fields (Name, Bio, Profile Picture, Location, Interests).

* **Test Steps:**

1. Navigate to the user profile editing page.
2. Access the privacy settings section for each field.
3. Set different privacy options (Public, Private, Friends Only) for each field.
4. Save the changes.

* **Postconditions:** No specific postconditions.

* **Expected Result:** Users should be able to adjust privacy settings for each profile field as desired, controlling the visibility of their information.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

Test Case 5

* **Test Case ID:** TC_UP_005

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

* **Test Case Description:** This test case verifies that the system provides confirmation to the user upon successfully saving their profile changes.

* **Test Suite:** User Profiles

* **Test Priority:** Low

* **Preconditions:**

- * A valid user account exists.

- * The user is logged in.

* **Test Data:**

- * Any valid changes to profile information.

* **Test Steps:**

1. Navigate to the user profile editing page.

2. Make changes to any profile field.

3. Click "Save" button.

* **Postconditions:** No specific postconditions.

* **Expected Result:** Upon clicking "Save", a confirmation message should be displayed to the user, indicating that the changes have been saved successfully.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

These test cases provide a comprehensive approach to ensuring the functionality and reliability of the User Profiles feature. Remember to execute these tests in various environments and scenarios to identify potential issues and ensure a high-quality product.

Remember, this is a starting point. Depending on the complexity of the platform, additional test cases might be needed to cover edge cases, boundary conditions, and potential vulnerabilities.

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:

Social Media Platform Test Cases: News Feed Feature

****Test Case 1:****

****Test Case ID:**** TC_NEWSFEED_001

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

****Test Case Description:**** This test case verifies that the user sees posts from accounts they are following in their news feed.

****Test Suite:**** User Interface (UI) Testing

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

****Preconditions:****

- * A user account is created and logged in.
- * Two or more accounts are followed by the logged-in user.
- * Each followed account has published at least one post.

****Test Data:**** List of followed accounts and their recent posts.

****Test Steps:****

1. Log in as a user with followed accounts.
2. Navigate to the news feed.
3. Examine the displayed posts.

4. Confirm that all posts belong to accounts the user is following.
5. Verify that posts are arranged in a chronological or algorithmic order as specified.

****Postconditions:****

- * The user remains logged in.

****Expected Result:**** The news feed displays posts from all the followed accounts in a clear and readable format.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 2:****

****Test Case ID:**** TC_NEWSFEED_002

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

****Test Case Description:**** This test case verifies that the news feed presents content chronologically or algorithmically as defined by the platform.

****Test Suite:**** User Interface (UI) Testing

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

****Preconditions:****

- * A user account is created and logged in.
- * Three or more accounts are followed by the logged-in user.
- * Each followed account has published at least two posts within different timestamps.

****Test Data:**** Timestamps of posts from different followed accounts.

****Test Steps:****

1. Log in as a user with followed accounts.
2. Navigate to the news feed.
3. Identify the timestamps of the posts displayed from different followed accounts.
4. Analyze the order of the posts based on their timestamps.
5. Verify that the posts are displayed either chronologically (most recent first) or algorithmically, adhering to the platform's defined logic.

****Postconditions:****

- * The user remains logged in.

****Expected Result:**** Posts are presented in a consistent order, either chronologically or algorithmically as specified by the platform.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 3:****

****Test Case ID:**** TC_NEWSFEED_003

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

****Test Case Description:**** This test case verifies the functionality of interacting with posts on the news feed.

****Test Suite:**** User Interface (UI) Testing

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

****Preconditions:****

* A user account is created and logged in.

* A post exists on the news feed.

****Test Data:**** Simple text content for a comment.

****Test Steps:****

1. Log in as a user.
2. Navigate to the news feed.
3. Locate a post on the news feed.
4. Click the "Like" button on the post.
5. Verify that the like count increments.
6. Click the "Comment" button and enter a simple comment.
7. Submit the comment.
8. Verify that the comment appears under the post.
9. Click the "Share" button.
10. Verify the options for sharing the post appear (e.g., Sharing to other platforms).

****Postconditions:****

* The user remains logged in.

****Expected Result:**** Users can successfully like, comment on, and share posts on the news feed.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 4:****

****Test Case ID:** TC_NEWSFEED_004****

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

****Test Case Description:** This test case verifies that the news feed refreshes and updates with new content in real-time.**

****Test Suite:** Performance Testing**

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

****Preconditions:****

* A user account is created and logged in.

* At least two accounts the user follows are actively posting content.

****Test Data:** Predetermined quantity and timing of posts from followed accounts.**

****Test Steps:****

1. Log in as a user and navigate to the news feed.
2. Monitor the news feed for updates from the actively posting accounts.
3. Track the timestamps of the new posts appearing in the feed.
4. Record the time lag between the new post publication and its appearance on the user's feed.
5. Repeat steps 1-4 multiple times over a significant period.

****Postconditions:****

* The user remains logged in.

****Expected Result:** New posts from followed accounts appear in the news feed within a reasonable delay, providing a real-time update experience.**

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Performance Testing**

****Test Case 5:****

****Test Case ID:** TC_NEWSFEED_005****

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

****Test Case Description:** This test case verifies the functionality of filtering content on the news feed by type or topic.**

****Test Suite:** User Interface (UI) Testing**

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

****Preconditions:****

* A user account is created and logged in.

* There are multiple posts with varying content types and potential topics.

****Test Data:** List of content types and topics present on the platform.**

****Test Steps:****

1. Log in as a user and navigate to the news feed.
2. Identify the filtering options available on the news feed (e.g., filter by type, topic, time period).
3. Select a specific content type (e.g., "photos") and apply the filter.
4. Analyze the displayed posts to confirm that they conform to the selected type.
5. Repeat steps 3 and 4 for multiple content types.
6. Select a specific topic (if applicable) and apply the filter.
7. Analyze the displayed posts to confirm they are related to the selected topic.

****Postconditions:****

* The user remains logged in.

****Expected Result:**** Users can effectively filter the news feed by content type and topic, narrowing down the displayed content to their desired preferences.

****Severity:**** Major

****Priority:**** Medium

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

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

Data - 3

Project name -

Social Media Platform

Project description -

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

Feature name -

Messaging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Messaging Feature:

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

****Test Case Title:**** Verify that users can send text messages.

****Test Case Description:**** This test case aims to verify the functionality of sending text messages through the messaging feature.

****Test Suite:**** Messaging Feature

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

****Preconditions:****

* Two users with registered accounts on the platform.

* Both users are logged in.

* Both users have each other as friends.

****Test Data:****

* Predefined sample text message content.

****Test Steps:****

1. Navigate to the messaging section of the platform.
2. Select the target friend's chat window.
3. Type the predefined sample text message in the message input field.
4. Click the "Send" button to send the message.
5. Verify that the text message is successfully sent and displayed in the chat window on both ends.

****Postconditions:****

* The chat window remains open.

****Expected Result:**** The message is successfully sent and received by the target user in real-time.

****Severity:**** Critical

****Priority:**** High

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

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

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

****Test Case Title:**** Verify that users can send images.

****Test Case Description:**** This test case aims to verify the functionality of sending images through the messaging feature.

****Test Suite:**** Messaging Feature

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

****Preconditions:****

- * Two users with registered accounts on the platform.
- * Both users are logged in.
- * Both users have each other as friends.
- * A sample image file is available on the user's device.

****Test Data:****

- * Sample image file.

****Test Steps:****

1. Navigate to the messaging section of the platform.
2. Select the target friend's chat window.
3. Click on the "Attach Image" button.
4. Select the predefined sample image file from the device.
5. Click "Send" to send the image.
6. Verify the image is successfully sent and displayed in the chat window on both ends.

****Postconditions:****

- * The chat window remains open.

****Expected Result:**** The image is successfully sent and received by the target user in real-time.

****Severity:** Critical**

****Priority:** High**

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

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

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

****Test Case Title:** Verify that users can start private conversations.**

****Test Case Description:** This test case aims to verify the functionality of initiating a new private conversation with another user.**

****Test Suite:** Messaging Feature**

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

****Preconditions:****

* A user is logged in.

* Another user with a registered account is available on the platform.

****Test Data:****

* Username of the target user.

****Test Steps:****

1. Navigate to the messaging section of the platform.
2. Click the "New Message" button.
3. Search for the target user by their username.

4. Select the target user from the search results.
5. Verify a new private conversation window opens with the selected user.

****Postconditions:****

- * A new chat window is open for the private conversation.

****Expected Result:**** A private conversation window is successfully opened with the selected user.

****Severity:**** Critical

****Priority:**** High

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

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

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

****Test Case Title:**** Verify that users receive read receipts for messages.

****Test Case Description:**** This test case aims to verify that users receive notifications indicating that their messages have been read by the recipient.

****Test Suite:**** Messaging Feature

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

****Preconditions:****

- * Two users with registered accounts on the platform.
- * Both users are logged in.
- * Both users have each other as friends.

****Test Data:****

- * A test message.

****Test Steps:****

1. Navigate to the messaging section of the platform.
2. Select the target friend's chat window.
3. Send the predefined test message.
4. Log out of the target friend's account.
5. Check the sender's message history.
6. Log back into the target friend's account and perform any actions relevant to read the message (e.g., view the chat).
7. Check the sender's message history again.

****Postconditions:****

- * No significant impact on platform functionality is expected.

****Expected Result:**** The sender receives a read receipt notification indicating that the target friend has read the message after they log back into the account.

****Severity:**** Minor

****Priority:**** Medium

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

****Test Case Approach:**** Positive & Negative

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

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

****Test Case Description:** This test case aims to verify that users receive notifications when they receive new messages in a private conversation.**

****Test Suite:** Messaging Feature**

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

****Preconditions:****

* Two users with registered accounts on the platform.

* Both users are logged in.

* Both users have each other as friends.

* Notification settings are enabled for both users.

****Test Data:****

* A test message.

****Test Steps:****

1. Log out of one user's account.
2. Navigate to the messaging section of the platform while logged in with the other user account.
3. Send the predefined test message to the user who is not currently logged in.
4. Verify that the offline user receives a notification about the new message. (e.g., a push notification or an alert in the app interface).

5. Log back into the offline user's account.

6. Open the messaging app and verify that the new message is present in the chat.

****Postconditions:****

* Both users remain logged in to the platform.

****Expected Result:****

The user who was offline receives a notification about the new message and can see it in their chat history when they log back in.

****Severity:**** Critical

****Priority:**** High

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

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

Data - 4

Project name -

Social Media Platform

Project description -

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

Feature name -

Content Sharing

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Social Media Platform: Content Sharing Test Cases

****Test Case ID:**** TS-CS-001

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

****Test Case Description:**** This test case ensures a user can successfully compose and publish a text post on the platform.

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

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

****Preconditions:****

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

* User is logged in.

****Test Data:**** Sample text post content

****Test Steps:****

1. Navigate to the "Create Post" section of the platform.
2. Enter the sample text post content into the designated text area.
3. Click the "Post" or "Share" button to publish the text post.
4. Verify that the text post is displayed on the user's profile and/or newsfeed.
5. Confirm that the post contains the provided text content accurately.

****Postconditions:****

* The text post is publicly visible (or visible to followers as per user setting).

****Expected Result:**** The user successfully creates and shares a text post as intended.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TS-CS-002

****Test Case Title:**** Verify that users can create and share photos.

****Test Case Description:**** This test case verifies the functionality of sharing photos on the platform.

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

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

****Preconditions:****

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

* User is logged in.

* User has access to a photo file on their device.

****Test Data:**** Sample photo file

****Test Steps:****

1. Navigate to the "Create Post" section of the platform.
2. Select "Upload Photo" or equivalent option to choose the sample photo file from the device.
3. Add any desired caption or description to the photo.
4. Click the "Post" or "Share" button to publish the photo.
5. Verify that the photo is displayed on the user's profile and/or newsfeed.

****Postconditions:****

* The photo is publicly visible (or visible to followers as per user setting).

****Expected Result:**** The user successfully uploads and shares the photo post as intended.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:** TS-CS-003**

****Test Case Title:** Verify that users can create and share videos.**

****Test Case Description:** This test case ensures the functionality of sharing videos on the platform.**

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

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

****Preconditions:****

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

* User is logged in.

* User has access to a video file on their device.

****Test Data:** Sample video file**

****Test Steps:****

1. Navigate to the "Create Post" section of the platform.
2. Select "Upload Video" or equivalent option to choose the sample video file from the device.
3. Add any desired description or caption to the video.
4. Click the "Post" or "Share" button to publish the video.
5. Verify that the video is displayed on the user's profile and/or newsfeed.

****Postconditions:****

* The video is publicly visible (or visible to followers as per user setting).

****Expected Result:**** The user successfully uploads and shares the video post as intended.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TS-CS-004

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

****Test Case Description:**** This test case ensures users can choose the audience for their posts.

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

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

****Preconditions:****

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

* User is logged in.

****Test Data:**** Sample text post content

****Test Steps:****

1. Navigate to the "Create Post" section of the platform.
2. Enter the sample text post content into the designated text area.
3. Access the audience selection option (e.g., "Share with Followers" or "Public").
4. Select "Share with Followers" and publish the post.
5. Verify that the post is visible only to the user's followers.

6. Repeat steps 3-5, selecting "Public" to publish the post.
7. Verify that the post is visible to all users on the platform.

****Postconditions:****

- * The post visibility aligns with the selected audience option.

****Expected Result:**** Users can successfully share content with either their followers or the entire public.

****Severity:**** Major

****Priority:**** Medium

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

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

****Test Case ID:**** TS-CS-005

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

****Test Case Description:**** This test case ensures users have the ability to modify or remove their own posts.

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

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

****Preconditions:****

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

- * User is logged in and has created at least one post.

****Test Data:**** An existing post created by the user.

****Test Steps:****

1. Locate the user's own existing post.
2. Access the post editing options (e.g., "Edit" or "Update").
3. Modify the post content or remove the post entirely.
4. Verify that the changes are reflected in the post display.

****Postconditions:****

* The post is edited or deleted successfully.

****Expected Result:**** Users can edit or delete their own posts as required.

****Severity:**** Major

****Priority:**** Medium

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

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

Data - 5

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share

posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Privacy Settings

Feature description -

Privacy settings allow users to control who can view their profile, posts, and other activity on the platform.

Users can choose between public, private, or restricted privacy settings, determining who can see their content, send them friend requests, or interact with them.

Userstory title -

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

Acceptance criteria -

['Users can set their profile to public, private, or restricted.', 'Privacy settings can be changed at any time.',

'Users can control visibility of individual posts.', 'The system should respect the privacy settings immediately.',

'Users should be notified of any changes in privacy settings.']

Generated Test Cases:

Test Cases for Privacy Settings Feature

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

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

****Test Case Description:**** This test case verifies the functionality of setting different privacy levels for a user profile. It will test the user's ability to select "public," "private," and "restricted" options and ensure the platform correctly applies the chosen privacy setting.

****Test Suite:**** Social Media Platform - User Settings

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

****Preconditions:****

* A registered user account exists.

* The user is logged into the platform.

****Test Data:****

* Valid username and password for user login.

****Test Steps:****

1. Navigate to the user's profile settings page.
2. Select "Public" privacy setting from the dropdown menu.
3. Confirm the selected privacy setting is "Public" on the settings page.
4. Select "Private" privacy setting from the dropdown menu.
5. Confirm the selected privacy setting is "Private" on the settings page.
6. Select "Restricted" privacy setting from the dropdown menu.
7. Confirm the selected privacy setting is "Restricted" on the settings page.

****Postconditions:****

* The user's profile should reflect the chosen privacy setting.

****Expected Result:**** The user should successfully change their profile privacy settings to public, private, or restricted without encountering any errors.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** System Testing, Functional Testing, Acceptance Testing

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

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

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

****Test Case Description:** This test case verifies the user's ability to modify their privacy settings after they have initially set them.**

****Test Suite:** Social Media Platform - User Settings**

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

****Preconditions:****

* A registered user account exists.

* The user has set their profile to a specific privacy level (public, private, or restricted).

* The user is logged into the platform.

****Test Data:** Same as TC-PS-001**

****Test Steps:****

1. Access the user's profile settings page.
2. Verify the current privacy setting.
3. Change the privacy setting to a different option.
4. Save the new privacy setting.
5. Confirm the profile privacy setting has been updated.

****Postconditions:****

* The user's profile should reflect the modified privacy setting.

****Expected Result:** The user must be able to change their privacy setting seamlessly between public, private, and restricted at any time without encountering errors or restrictions.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** System Testing, Functional Testing, Acceptance Testing**

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

Do you wish me to continue creating test cases for this feature?

Data - 6

Project name -

Social Media Platform

Project description -

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

Feature name -

Notifications

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Social Media Platform Notifications Feature

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

****Test Case ID:**** NOTIF-001

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

****Test Case Description:**** This test case verifies that users receive notifications when someone likes their posts. It will involve creating a post, simulating a like from another user, and checking if the user receives a notification.

****Test Suite:**** Notifications

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

****Preconditions:****

* A logged-in user with an active account.

* A post created by the user.

****Test Data:**** Test user account, sample post content.

****Test Steps:****

1. Create a post on the social media platform.
2. Log in with a different user account.
3. Like the post created in step 1.
4. Switch back to the original user's account.
5. Check the notification section for a notification indicating a like on their post.

****Postconditions:****

* Post remains intact on the platform.

* Notification persists for the user.

****Expected Result:**** The user receives a notification indicating that their post has been liked.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing, User Acceptance Testing

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

****Test Case ID:**** NOTIF-002

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

****Test Case Description:**** This test case verifies that users receive notifications when someone comments on their posts. It will involve creating a post, simulating a comment from another user, and checking if the user receives a notification.

****Test Suite:**** Notifications

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

****Preconditions:****

* A logged-in user with an active account.

* A post created by the user.

****Test Data:**** Test user account, sample post content, sample comment text.

****Test Steps:****

1. Create a post on the social media platform.
2. Log in with a different user account.
3. Add a comment to the post created in step 1.
4. Switch back to the original user's account.

5. Check the notification section for a notification indicating a comment on their post.

****Postconditions:****

- * Post remains intact on the platform.
- * Comment remains visible on the post.
- * Notification persists for the user.

****Expected Result:**** The user receives a notification indicating that their post has been commented on.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing, User Acceptance Testing

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

****Test Case ID:**** NOTIF-003

****Test Case Title:**** Verify that users receive notifications for new shares of their posts

****Test Case Description:**** This test case verifies that users receive notifications when someone shares their posts. It will involve creating a post, simulating a share from another user, and checking if the user receives a notification.

****Test Suite:**** Notifications

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

****Preconditions:****

- * A logged-in user with an active account.
- * A post created by the user.

****Test Data:**** Test user account, sample post content.

****Test Steps:****

1. Create a post on the social media platform.
2. Log in with a different user account.
3. Share the post created in step 1 to their own profile.
4. Switch back to the original user's account.
5. Check the notification section for a notification indicating that their post has been shared.

****Postconditions:****

- * Post remains intact on the platform.
- * Shared post is visible on the sharing user's profile.
- * Notification persists for the user.

****Expected Result:**** The user receives a notification indicating that their post has been shared.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing, System Testing, User Acceptance Testing**

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

****Test Case ID:** NOTIF-004**

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

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

****Test Suite:** Notifications**

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

****Preconditions:****

* Two logged-in user accounts: the user who sends the request and the user who receives the request.

****Test Data:** Test user accounts, sample friend request message.**

****Test Steps:****

1. A user (User A) logs in and navigates to their user profile.
2. Another user (User B) logs in and navigates to User A's profile.
3. User B sends a friend request to User A.
4. User A checks their notification section for a notification indicating a new friend request from User B.

****Postconditions:****

* Friend request notification persists for User A.

****Expected Result:**** User A receives a notification about the friend request from User B.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing, User Acceptance Testing

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

****Test Case ID:**** NOTIF-005

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

****Test Case Description:**** This test case verifies that users have options to customize their notification preferences and those changes are reflected in their notification behavior.

****Test Suite:**** Notifications

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

****Preconditions:****

* A logged-in user with an active account.

****Test Data:**** Test user account, various notification settings (e.g., likes, comments, mentions, friend requests).

****Test Steps:****

1. Login to the user's account.
2. Navigate to the user's profile settings or notification settings page.
3. Toggle various notification settings on and off (e.g., enable likes notifications, disable comment notifications).
4. Simulate activities generating notifications (likes, comments, mentions, friend requests).
5. Check the notification section to confirm only the enabled notifications are received.

****Postconditions:****

* User profile or notification settings persist the adjusted preferences.

****Expected Result:****

User receives notifications only for the activities whose notification preferences are enabled.

****Severity:**** Major

****Priority:**** High

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

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

Data - 7

Project name -

Social Media Platform

Project description -

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

Feature name -

Search Functionality

Feature description -

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

with others who share similar interests.

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Social Media Platform - Search Functionality

****Test Case ID:**** SMPT-001

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

****Test Case Description:**** This test case verifies that users can input keywords into the search bar and receive relevant search results for posts, profiles, and topics containing those keywords.

****Test Suite:**** Search Functionality

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

****Preconditions:****

- A user is logged into the platform.

****Test Data:****

- Keywords related to diverse content on the platform (e.g., "travel," "technology," "music")

****Test Steps:****

1. Navigate to the search bar on the platform.
2. Enter a predefined keyword into the search bar.
3. Click on the search button or press enter.
4. Verify that the search results page displays relevant posts, profiles, and topics containing the entered keyword.
5. Check if the results are displayed in a logical order of relevance.

****Postconditions:****

- Search results page is closed.
- No adverse effects on the platform's functionality.

****Expected Result:**** A list of relevant search results is displayed, including posts, profiles, and topics containing the entered keyword, ordered by relevance.

****Severity:**** Critical

****Priority:**** High

****Test Case ID:**** SMPT-002

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

****Test Case Description:**** This test case verifies that users can search for content using relevant hashtags and receive corresponding results.

****Test Suite:**** Search Functionality

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

****Preconditions:****

- A user is logged into the platform.

****Test Data:****

- Hashtags commonly used on the platform related to diverse content (e.g., #technews, #photography, #travel)

****Test Steps:****

1. Navigate to the search bar on the platform.
2. Enter a predefined hashtag into the search bar using the '#' symbol.
3. Click on the search button or press enter.
4. Verify that the search results page displays relevant posts, profiles, and topics containing the entered hashtag.
5. Check if the results are displayed in a logical order of relevance.

****Postconditions:****

- Search results page is closed.
- No adverse effects on the platform's functionality.

****Expected Result:**** A list of relevant search results is displayed, including posts, profiles, and topics containing the entered hashtag, ordered by relevance.

****Severity:** Critical**

****Priority:** High**

****Test Case ID:** SMPT-003**

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

****Test Case Description:**** This test case verifies that the search functionality returns results from all relevant content types: profiles, posts, and topics.

****Test Suite:**** Search Functionality

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

****Preconditions:****

- A user is logged into the platform.

****Test Data:****

- Search keyword related to content found in all three categories (e.g., "art").

****Test Steps:****

1. Navigate to the search bar on the platform.
2. Enter a predefined keyword into the search bar.
3. Click on the search button or press enter.
4. Verify that the search results page displays a mix of profiles, posts, and topics relevant to the entered keyword.

****Postconditions:****

- Search results page is closed.

- No adverse effects on the platform's functionality.

****Expected Result:**** A diverse range of search results is displayed, including profiles, posts, and topics related to the entered keyword.

****Severity:**** Critical

****Priority:**** High

****Test Case ID:**** SMPT-004

****Test Case Title:**** Verify that search results can be filtered by content type.

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

****Test Suite:**** Search Functionality

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

****Preconditions:****

- A user is logged into the platform.

****Test Data:****

- Search keyword related to various types of content (e.g., "music").

****Test Steps:****

1. Navigate to the search bar on the platform.
2. Enter a predefined keyword into the search bar.
3. Click on the search button or press enter.
4. Locate and select the filter option to display only specific content types (e.g., posts).
5. Verify that the search results page only displays relevant posts containing the entered keyword.
6. Repeat steps 4-5 with different content type filters.

****Postconditions:****

- Search results page for each filter is closed.
- No adverse effects on the platform's functionality.

****Expected Result:**** Successfully filtering search results by content type displays only relevant results matching the selected filter criteria for each keyword used.

****Severity:**** Critical

****Priority:**** High

****Test Case ID:**** SMPT-005

****Test Case Title:**** Verify that search functionality provides auto-suggestions.

****Test Case Description:**** This test case verifies that the search bar provides relevant auto-suggestions as the user types, aiding in quick and efficient searching.

****Test Suite:** Search Functionality**

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

****Preconditions:****

- A user is logged into the platform.

****Test Data:****

- Partial keywords or phrases related to various content on the platform.

****Test Steps:****

1. Navigate to the search bar on the platform.
2. Begin typing a partial keyword or phrase.
3. Observe the auto-suggestions appearing below the search bar.
4. Verify that the suggestions are relevant to the typed characters and offer various search options.

****Postconditions:****

- Search bar is cleared after testing suggestions.
- No adverse effects on the platform's functionality.

****Expected Result:**** As the user types, the search bar displays relevant auto-suggestions for keywords, hashtags, or profiles, helping them refine their search quickly.

****Severity:** Minor**

****Priority:** Medium**

Data - 8

Project name -

Social Media Platform

Project description -

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

Feature name -

Hashtags and Tagging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Hashtags and Tagging Feature

Here are 5 detailed test cases for the "Hashtags and Tagging" feature:

****Test Case ID:**** TC_HT_001

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

****Test Case Description:**** This test case will verify the functionality of adding hashtags to user posts. It will ensure that users can correctly input hashtags, separate them with spaces or commas, and that they appear correctly on the post.

****Test Suite:**** Social Media Platform - User Content Features

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

****Preconditions:****

- * User Account is created and logged in.

- * A new post is initiated.

- * At least one valid hashtag is known.

****Test Data:**** Valid hashtags (e.g., #SocialMedia, #TechTrends, #QATesting)

****Test Steps:****

1. Compose a new post with text content.
2. Type a hashtag without the "#" symbol followed by a space.
3. Type another hashtag, separated by a space.
4. Optionally, type additional hashtags, separated by spaces or commas.
5. Observe the hashtags in the post preview.
6. Submit the post.

****Postconditions:****

- * The post is successfully published.

****Expected Result:**** Hashtags are displayed correctly on the post, with the "#" symbol appended automatically. Clicking on a hashtag leads to a list of posts containing that hashtag.

****Severity:** Major**

****Priority:** High**

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

****Test Case ID:** TC_HT_002**

****Test Case Title:** Verify that users can tag other profiles, brands, or locations.**

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

****Test Suite:** Social Media Platform - User Content Features**

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

****Preconditions:****

* User Account is created and logged in.

* At least two user accounts exist.

* A new post is initiated.

****Test Data:** Names of valid user profiles, brands, or locations.**

****Test Steps:****

1. Compose a new post with text content.
2. Type "@" followed by the name of a user.
3. Observe the completion suggestions appearing as the user types.
4. Select the desired user from the suggestions.
5. Repeat steps 2-4 to tag additional users, brands, or locations.
6. Submit the post.

****Postconditions:****

* The post is successfully published.

****Expected Result:** The tagged users, brands, or locations appear as clickable links in the post. Clicking on them leads to their respective profiles or pages.**

****Severity:** Major**

****Priority:** High**

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

****Test Case ID:** TC_HT_003**

****Test Case Title:** Verify that hashtags and tags are clickable.**

****Test Case Description:** This test case verifies that hashtags and tags are clickable and lead to relevant content.**

****Test Suite:** Social Media Platform - User Content Features**

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

****Preconditions:****

* Several posts have been created with different hashtags and tags.

****Test Data:** Hashtags and tags from a variety of posts.**

****Test Steps:****

1. View a post that contains hashtags and tags.
2. Click on a hashtag.
3. Observe the feed containing posts with that hashtag.
4. Click on a tag (user profile).
5. Observe the user's profile page.

****Postconditions:****

* No errors occur when clicking on the hashtags or tags.

****Expected Result:** Clicking on a hashtag navigates to a page displaying all posts associated with that hashtag. Clicking on a tag navigates to the profile page of the tagged user.**

****Severity:** Major**

****Priority:** Medium**

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

****Test Case ID:** TC_HT_004**

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

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

****Test Suite:** Social Media Platform - User Content Features**

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

****Preconditions:****

*** Several posts have been created with different hashtags.**

****Test Data:** Valid hashtags used in previous posts.**

****Test Steps:****

1. Navigate to the search page.
2. Type a hashtag into the search bar.
3. Observe the search results displayed.

****Postconditions:****

*** The results displayed are relevant to the entered hashtag.**

****Expected Result:** The search results page displays a list of posts containing the entered hashtag.**

****Severity:** Major**

****Priority:** High**

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

****Test Case ID:** TC_HT_005**

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

****Test Case Description:** This test case verifies the functionality of hashtag suggestions.**

****Test Suite:** Social Media Platform - User Content Features**

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

****Preconditions:****

* Users have been actively posting on the platform.

****Test Data:**** Commonly used hashtags or expected popular suggestions based on user activity.

****Test Steps:****

1. Compose a new post.
2. Start typing a relevant keyword or phrase.
3. Observe the suggestions for hashtags displayed below the text.
4. Select a suggested hashtag and observe its behavior.

****Postconditions:****

* Suggestions are relevant to the entered keyword.

****Expected Result:****

As the user types, the system suggests popular hashtags related to the input, making it easier for users to categorize their content effectively.

These test cases provide a comprehensive framework for evaluating the functionality of the Hashtags and Tagging feature. By executing these tests, QA engineers can ensure that the feature meets the specified acceptance criteria and delivers a high-quality user experience.

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:

Okay, here are 5 test cases for the "Emojis and Reactions" feature of the Social Media Platform, following the template you provided.

Test Case 1

****Test Case ID:**** TC_EmojiReaction_001

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

****Test Case Description:**** This test case verifies the functionality of reacting to posts with different available emojis. It will involve selecting various emojis to express different reactions to a sample post.

****Test Suite:**** User Interaction - Reactions

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

****Preconditions:****

- * A user is logged into the platform.
- * A sample post with no reactions is available for testing.
- * The "Reactions" feature is enabled.

****Test Data:****

- * Different emojis representing various reactions (e.g., 🍕 , 🍕🍕 , 🍕 , 🍕)

****Test Steps:****

1. Navigate to the sample post.
2. Click on the "Reaction" button or icon below the post.
3. Select one of the available emojis from the reaction selection panel.
4. Verify that the selected emoji appears as a reaction to the post.
5. Repeat steps 3-4 with different emojis.

****Postconditions:****

- * No changes to user data or profile settings.
- * Correct emoji reactions remain visible under the post.

****Expected Result:**** The selected emoji reaction should be successfully applied to the post and visually displayed.

****Severity:** Major**

****Priority:** High**

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

Test Case 2

****Test Case ID:**** TC_EmojiReaction_002

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

****Test Case Description:**** This test case checks if emojis can be inserted and displayed correctly within comments and direct messages.

****Test Suite:**** User Interaction - Reactions

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

****Preconditions:****

* A user is logged into the platform.

* Two users are available for testing.

****Test Data:****

* Different emojis representing various reactions (e.g., 🤔 , 🤔 , 🤔)

****Test Steps:****

1. User 1 creates a new post with sample content.
2. User 2 navigates to the post and types a comment containing an emoji.
3. User 2 posts the comment.
4. Verify that the emoji is correctly displayed in the comment section.
5. User 1 sends a direct message to User 2 containing an emoji.
6. User 2 receives and reads the direct message.
7. Verify that the emoji is correctly displayed within the direct message.

****Postconditions:****

- * No changes to user data or privacy settings.
- * Existing messages and comments remain unaltered.

****Expected Result:**** Emojis should be inserted and displayed correctly within comments and direct messages.

****Severity:**** Major

****Priority:**** High

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

Test Case 3

****Test Case ID:**** TC_EmojiReaction_003

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

****Test Case Description:**** This test case ensures that the platform correctly displays the number of reactions a post has received.

****Test Suite:**** User Interface - Reactions

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

****Preconditions:****

- * A user is logged into the platform.
- * A sample post with multiple reactions is available for testing.

****Test Data:****

- * Various counts of reactions (e.g., 1, 5, 10, 100)

****Test Steps:****

1. Navigate to the sample post.
2. Observe the display of the reaction count below the post.
3. Add reactions to the post (using different emoji types).
4. Observe the change in the reaction count after each subsequent reaction is added.
5. Remove reactions from the post.
6. Verify that the reaction count correctly decreases with each removal.

****Postconditions:****

* The reaction count accurately reflects the current number of reactions.

****Expected Result:**** The reaction count should update in real-time as reactions are added and removed, displaying the accurate total.

****Severity:**** Major

****Priority:**** High

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

Test Case 4

****Test Case ID:**** TC_EmojiReaction_004

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

****Test Case Description:**** This test case verifies that the platform supports a substantial selection of

commonly used emojis across various categories.

****Test Suite:**** User Interface - Reactions

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

****Preconditions:****

* A user is logged into the platform.

* Access to emoji picker/menu.

****Test Data:****

* List of diverse emoji categories (e.g., emotions, activities, objects)

****Test Steps:****

1. Open a post or message composition field.
2. Access the emoji picker/menu.
3. Carefully browse through the available emoji categories.
4. Select a representative emoji from each category.
5. Verify that all selected emojis are correctly displayed and inserted into the composition field.

****Postconditions:****

* No changes to user data or profile settings.

****Expected Result:**** The platform should display and support a wide array of emojis across various categories.

****Severity:**** Minor

****Priority:**** Medium

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

Test Case 5

****Test Case ID:**** TC_EmojiReaction_005

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

****Test Case Description:**** This test case examines the ability of users to modify or delete their previously applied reactions.

****Test Suite:**** User Interaction - Reactions

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

****Preconditions:****

* A user is logged into the platform.

* A sample post with the user's existing reaction.

****Test Data:****

* Different emoji reactions to be replaced or removed.

****Test Steps:****

1. Navigate to the sample post where the user has already reacted.
2. Click on the user's existing reaction.
3. Select a different emoji reaction from the menu.
4. Verify the reaction on the post is successfully changed.

5. Interact with the post again and click "Remove Reaction."
6. Verify that the reaction is removed, and the post shows the updated reaction count.

****Postconditions:****

* Reactions on the post are accurately updated.

****Expected Result:**** Users should be able to modify or remove their reactions to posts without encountering errors or unexpected behavior.

****Severity:**** Major

****Priority:**** High

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

Data - 10

Project name -

Social Media Platform

Project description -

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

Feature name -

Live Streaming

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Live Streaming Feature

****Test Case ID:**** TC_LIVESTREAM_001

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

****Test Case Description:**** This test case verifies the functionality of starting and stopping a live stream from a user perspective.

****Test Suite:**** Live Streaming

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

****Preconditions:****

* User is logged into the social media platform.

- * User has a properly set up profile.
- * User has access to the live streaming feature.

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

****Test Steps:****

1. Navigate to the live streaming functionality.
2. Click the "Start Live Stream" button.
3. Monitor the live stream interface and verify that broadcasting begins successfully.
4. Click the "Stop Live Stream" button.
5. Verify that the live stream stops and the interface reflects the change.

****Postconditions:****

- * Stream recording (if applicable) will be available for viewing afterwards.
- * Live stream is no longer broadcasting.

****Expected Result:**** The user should be able to start and stop live streams without encountering any errors or unexpected behavior.

****Severity:**** Critical

****Priority:**** High

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

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

****__****

****Test Case ID:** TC_LIVESTREAM_002**

****Test Case Title:** Verify that live streams can be public or private**

****Test Case Description:** This test case verifies the functionality of setting live stream privacy settings.**

****Test Suite:** Live Streaming**

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

****Preconditions:****

- * User is logged into the social media platform.
- * User has a properly set up profile.
- * User has access to the live streaming feature.

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

****Test Steps:****

1. Navigate to the live streaming functionality.
2. Click the "Start Live Stream" button.
3. Select the "Public" privacy setting.
4. Verify that the selected setting is reflected in the stream information.
5. Repeat steps 2-4, this time selecting the "Private" privacy setting.
6. Verify that the selected setting is reflected in the stream information.

****Postconditions:** The stream is broadcast with the selected privacy setting.**

****Expected Result:** The user should be able to select either public or private privacy settings for their live**

streams.

****Severity:** Critical**

****Priority:** High**

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

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

****__****

****Test Case ID:** TC_LIVESTREAM_003**

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

****Test Case Description:** This test case verifies the functionality of viewer interaction with a live stream.**

****Test Suite:** Live Streaming**

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

****Preconditions:****

* A live stream is running.

* Viewer user is logged into the social media platform.

****Test Data:** Sample comments and reactions to utilize.**

****Test Steps:****

1. Navigate to the live stream.

2. Select an option from the available reactions. (e.g., Like, Love, Laugh, etc.)
3. Enter a sample comment in the comment box and click "Post."
4. Verify that the reaction and comment appear in the live stream interface.
5. Repeat steps 2-4 with a different reaction and comment.

****Postconditions:****

* Comment and reaction are visible on the live stream.

****Expected Result:**** Viewers should be able to interact with the live streamer by sending comments and using available reactions without encountering errors.

****Severity:**** Critical

****Priority:**** High

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

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

****__****

****Test Case ID:**** TC_LIVESTREAM_004

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

****Test Case Description:**** This test case ensures that the live stream interface accurately displays the number of viewers and incoming comments.

****Test Suite:**** Live Streaming

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

****Preconditions:****

- * A live stream is running.
- * At least one viewer is connected to the stream.

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

****Test Steps:****

1. Navigate to the live stream.
2. Observe the viewer count displayed on the stream interface.
3. Send a comment through the interface.
4. Verify that the viewer count is updated correctly and the new comment appears in the comment section.

****Postconditions:**** No specific postconditions.

****Expected Result:**** The live stream interface should dynamically update the viewer count and display incoming comments in real time.

****Severity:**** Critical

****Priority:**** High

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

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

****__****

****Test Case ID:** TC_LIVESTREAM_005**

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

****Test Case Description:** This test case verifies functionality of recording and saving a live stream for subsequent viewing.**

****Test Suite:** Live Streaming**

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

****Preconditions:****

- * User is logged into the social media platform.
- * User has a properly set up profile.
- * User has access to the live streaming feature.
- * Live stream recording is enabled (if applicable).

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

****Test Steps:****

1. Navigate to the live streaming functionality.
2. Start a live stream.
3. End the live stream.
4. (Optional) Verify the recording availability in a designated section within the user's profile or dashboard.
5. Access the saved recording.
6. Verify that the recording playback functionality is working as expected.

****Postconditions:****

* The live stream recording is saved and accessible to the user.

****Expected Result:**** Users should be able to locate and view the saved recording of their live streams with proper playback functionality.

****Severity:**** Major

****Priority:**** Medium

****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:

Social Media Platform: Stories Feature Test Cases

These test cases aim to verify the functionality of the "Stories" feature on the Social Media Platform, as outlined in the user story "As a user, I want to share stories so that I can share temporary moments with my followers."

****Test Case ID:**** STORY-001

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

****Test Case Description:**** This test case verifies the user's ability to create a story using a single image from their device gallery.

****Test Suite:**** Stories Feature

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

****Preconditions:****

- * User is logged into the platform.
- * User has at least one image in their device gallery.

****Test Data:****

Test image (.jpg or .png format)

****Test Steps:****

1. Navigate to the "Create Story" option within the platform.
2. Select "Photo" as the content type.
3. Choose the desired image from the gallery.
4. Add any desired text overlays or filters.
5. Publish the story.

****Postconditions:****

- * The story appears on the user's profile and is viewable by their followers.
- * The story has the expected duration (24 hours).
- * The story can be viewed by other users who follow the account.

****Expected Result:**** The story is successfully created and published with the selected photo.

****Test Case ID:** STORY-002**

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

****Test Case Description:**** This test case verifies the user's ability to create a story using a short video recording or selecting from their device's video library.

****Test Suite:**** Stories Feature

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

****Preconditions:****

* User is logged into the platform.

* User has access to the device's camera or has a video file in their device library.

****Test Data:****

Short video clip (.mp4 format)

****Test Steps:****

1. Navigate to the "Create Story" option within the platform.
2. Select "Video" as the content type.
3. Record a new video clip using the device camera or choose a pre-recorded video from the library.
4. Add any desired text overlays, stickers, or filters.
5. Publish the story.

****Postconditions:****

- * The story appears on the user's profile and is viewable by their followers.
- * The story has the expected duration (24 hours).
- * The video plays correctly within the story.

****Expected Result:**** The story is successfully created and published with the selected video.

****Test Case ID:**** STORY-003

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

****Test Case Description:**** This test case verifies that users can add text to their stories, customize font style, color, and size, and position the text elements.

****Test Suite:**** Stories Feature

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

****Preconditions:****

- * User is logged into the platform.
- * User has created a story with either an image or video

****Test Data:****

Simple text phrases

****Test Steps:****

1. Open the story creation screen.
2. Add an image or video to the story.
3. Tap on the "Text" overlay option.
4. Type different text phrases.
5. Adjust the font style, size, and color of the text.
6. Drag and drop the text overlay to different positions on the story.
7. Publish the story.

****Postconditions:****

- * Text overlays are visible on the published story.
- * Text formatting (font, size, color) is applied correctly.
- * Text overlays can be positioned anywhere within the story canvas.

****Expected Result:**** The story is successfully created with various text overlays, demonstrating customizable text features.

****Test Case ID:**** STORY-004

****Test Case Title:**** Verify that users can share stories publicly or with selected followers.

****Test Case Description:**** This test case verifies that users can control the visibility of their stories by choosing to share them publicly or with a specific list of followers.

****Test Suite:**** Stories Feature

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

****Preconditions:****

* User is logged into the platform

* User has at least one follower

****Test Data:****

* A list of followers to be shared with

****Test Steps:****

1. Navigate to the "Create Story" option within the platform.
2. Create a new story with either a photo or video.
3. Select "Share with" option.
4. Choose either "Public" or "Followers Only" option.
5. If "Followers Only" is chosen, select specific followers to share with.
6. Publish the story.

****Postconditions:****

* The story should only be visible to the chosen audience (public or selected followers).

* The user should not be able to see their own story once it is published in the "Followers Only" mode unless they are logged as one of the selected followers.

****Expected Result:**** The story is shared accordingly to the selected privacy setting (public or with chosen followers).

****Test Case ID:**** STORY-005

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

****Test Case Description:**** This test case verifies that stories expire after 24 hours from the time of publishing.

****Test Suite:**** Stories Feature

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

****Preconditions:****

* User is logged into the platform and has created a story.

****Test Data:****

Timestamp of the story creation

****Test Steps:****

1. Create a new story at a specific time (record the timestamp).
2. Leave the app and wait for 24 hours (allowing for a slight error margin).
3. Re-open the app and navigate to the user's profile.
4. Attempt to view the story created 24 hours ago.

****Postconditions:****

* The story should not be visible on the user's profile.

Expected Result: The story disappears after the expected 24-hour period.

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:

Test Cases for Analytics and Insights Feature

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

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

****Test Case Description:**** This test case verifies that users can access and view demographic information about their followers, including age range, gender, location, and interests.

****Test Suite:**** User Stories - Analytics & Insights

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

****Preconditions:****

* A user is logged into the platform.

* The user has at least one follower.

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

****Test Steps:****

1. Navigate to the user's profile page.
2. Click on the "Analytics" or "Insights" tab.
3. Locate the "Follower Demographics" section.
4. Verify that data is displayed for age range, gender, location, and interests.

****Postconditions:****

* User remains on their profile page.

* Analytics dashboard is still accessible.

****Expected Result:**** Demographics data is displayed accurately and comprehensively for the user's followers.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Acceptance Testing

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

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

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

****Test Case Description:**** This test case verifies that users can access and view metrics related to the reach and engagement of their posts, including views, likes, comments, shares, and reach percentage.

****Test Suite:**** User Stories - Analytics & Insights

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

****Preconditions:****

* A user is logged into the platform.

* The user has published at least one post.

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

****Test Steps:****

1. Navigate to the user's profile page.
2. Click on the "Analytics" or "Insights" tab.
3. Locate the "Post Performance" or "Engagement Metrics" section.
4. Select a specific post from the list.
5. Verify that metrics for views, likes, comments, shares, and reach percentage are displayed accurately.

****Postconditions:****

- * User remains on the analytics dashboard.
- * Specific post remains highlighted.

****Expected Result:**** Detailed engagement and reach metrics are available for each post, providing insights into audience interaction.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Acceptance Testing

****Test Case Approach:**** Positive, Destructive

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

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

****Test Case Description:**** This test case verifies that users can visualize trends in their follower growth over time through charts or graphs.

****Test Suite:**** User Stories - Analytics & Insights

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

****Preconditions:****

* A user is logged into the platform.

* The user has a history of follower growth.

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

****Test Steps:****

1. Navigate to the user's profile page.
2. Click on the "Analytics" or "Insights" tab.
3. Locate the "Audience Growth" section.
4. Verify that a chart or graph is displayed showing follower growth trends over a chosen time period.
5. Observe the chart to identify general trends in follower growth (increase, decrease, stability).

****Postconditions:****

* User remains on the analytics dashboard.

* Chart or graph remains displayed.

****Expected Result:**** Visually informative charts or graphs accurately depict the trends in user follower growth over time.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Acceptance Testing

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

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

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

****Test Case Description:** This test case ensures that the presentation of analytics data is intuitive, understandable, and visually appealing to users.**

****Test Suite:** User Stories - Analytics & Insights**

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

****Preconditions:****

* A user is logged into the platform.

* The user has accessed at least one analytics section (e.g., follower demographics, post performance).

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

****Test Steps:****

1. Navigate to the user's profile page.
2. Click on the "Analytics" or "Insights" tab.
3. Observe the layout, colors, font sizes, and use of visual elements (charts, graphs, tables) across different analytics sections.
4. Assess the overall clarity and ease of understanding the presented data.

****Postconditions:****

* User remains on the analytics dashboard.

* No errors or issues with data presentation are encountered.

****Expected Result:**** The analytics are presented in a clear, concise, and visually appealing manner, making it easy for users to interpret and understand their data.

****Severity:**** Major

****Priority:**** Medium

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

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

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

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

****Test Case Description:**** This test case examines the functionality of filtering analytics data by specific time periods and content types (e.g., posts, stories, videos).

****Test Suite:**** User Stories - Analytics & Insights

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

****Preconditions:****

* A user is logged into the platform.

* The user has a history of posts and/or content creation within different time ranges.

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

****Test Steps:****

1. Navigate to the user's profile page.

2. Click on the "Analytics" or "Insights" tab.
3. Locate the filter options for time range and content type.
4. Select different time ranges (e.g., past week, past month, all time).
5. Choose specific content types to filter (e.g., posts, stories, videos).
6. Verify that the displayed analytics data reflects the chosen filters accurately.

****Postconditions:****

- * User remains on the analytics dashboard.
- * Filtered data is displayed according to the selected criteria.

****Expected Result:**** Filtering options effectively allow users to focus on specific time periods and content types, providing targeted insights.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Acceptance Testing

****Test Case Approach:**** Positive, Negative

Data - 13

Project name -

Social Media Platform

Project description -

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

Feature name -

Events and Groups

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Event and Groups Feature

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

****Test Case Title:**** Verify that a user can create an event.

****Test Case Description:**** This test case verifies the functionality of creating an event on the platform, ensuring all required information is provided and displayed correctly.

****Test Suite:**** Event & Groups Feature

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

****Preconditions:****

* User is logged in.

* User has necessary permissions to create events.

****Test Data:****

* Event name: "Test Event"

* Event date: 2024-03-15

* Event time: 10:00 AM

* Event location: "Test City, Test State"

* Event description: "This is a test event for the social media platform."

****Test Steps:****

1. Navigate to the "Events" section.

2. Click on the "Create Event" button.

3. Fill in the required event details using the provided test data.

4. Click on the "Save" button.

5. Verify that the created event is displayed with the entered details.

****Postconditions:****

* Created event remains accessible to the user.

* Other users can view the created event.

****Expected Result:**** The event is successfully created and displayed with the entered information.

****Severity:**** Blocker

****Priority:**** High

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

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

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

****Test Case Title:**** Verify that a user can join an interest-based group.

****Test Case Description:**** This test case verifies the functionality of joining an existing interest-based group on the platform.

****Test Suite:**** Event & Groups Feature

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

****Preconditions:****

* User is logged in.

* An interest-based group exists with the name "Test Group".

****Test Data:****

* Group name: "Test Group"

****Test Steps:****

1. Navigate to the "Groups" section.
2. Search for the group using the defined test data.
3. Click on the "Join Group" button for the "Test Group".
4. Verify that the user is successfully added to the group.

****Postconditions:****

* User can view the group's posts and discussions.

* User can participate in group discussions.

****Expected Result:**** The user successfully joins the "Test Group".

****Severity:**** Blocker

****Priority:**** High

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

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

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

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

****Test Case Description:**** This test case verifies the functionality of organizing group discussions by topic.

****Test Suite:**** Event & Groups Feature

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

****Preconditions:****

* User is logged in.

* User is a member of an existing interest-based group.

****Test Data:****

* Group name: "Test Group"

* Discussion topic: "Technology"

* Discussion topic: "Travel"

****Test Steps:****

1. Navigate to the "Groups" section.
2. Select the "Test Group".
3. Click on the "New Discussion" button.
4. Create a discussion post with the topic "Technology".
5. Create a second discussion post with the topic "Travel".
6. Verify that discussions are displayed separately by topic.

****Postconditions:****

- * Discussions remain organized by topic.
- * Users can filter discussions by topic.

****Expected Result:**** Discussions are displayed and organized correctly by the specified topics.

****Severity:**** Major

****Priority:**** Medium

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

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

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

****Test Case Title:**** Verify that event details are displayed correctly.

****Test Case Description:**** This test case verifies the display of all required event details, ensuring accuracy and completeness.

****Test Suite:**** Event & Groups Feature

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

****Preconditions:****

* User is logged in.

* An event exists with the name "Test Event".

****Test Data:****

* Event name: "Test Event"

* Event date: 2024-03-15

* Event time: 10:00 AM

* Event location: "Test City, Test State"

* Event description: "This is a test event for the social media platform."

****Test Steps:****

1. Navigate to the event page for "Test Event".
2. Verify that the event name, date, time, location, and description are displayed correctly.
3. Ensure that any additional event details are also displayed accurately.

****Postconditions:****

* Event details remain visible and accurate.

****Expected Result:**** All event details are displayed correctly on the event page.

****Severity:**** Blocker

****Priority:** High**

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

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

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

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

****Test Case Description:** This test case verifies the functionality of RSVPing to an event and displaying the list of attendees.**

****Test Suite:** Event & Groups Feature**

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

****Preconditions:****

* User 1 is logged in.

* User 2 is logged in.

* An event with the name "Test Event" exists.

****Test Data:****

* Event name: "Test Event"

****Test Steps:****

1. User 1 navigates to the event page for "Test Event" and clicks on "RSVP".
2. User 2 navigates to the same event page and clicks on "RSVP".

3. User 1 verifies that their RSVP status is displayed on the event page.
4. User 1 views the list of attendees and confirms that User 2 is listed.
5. User 2 verifies that their RSVP status is displayed.
6. User 2 views the list of attendees and verifies that User 1 is listed.

****Postconditions:****

- * RSVPs are reflected on the event page.
- * Attendee list is updated after RSVP changes.

****Expected Result:**** Users can successfully RSVP to an event and view the list of attendees.

****Severity:**** Blocker

****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:

Test Cases for Monetization Options Feature

****Test Case ID:**** TC_MNT_001

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

****Test Case Description:**** This test case verifies that registered users are able to initiate the application process for monetization options on the platform.

****Test Suite:**** User Authentication & Monetization

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

****Preconditions:****

* User is successfully logged in.

* User has a complete profile with necessary information.

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

****Test Steps:****

1. Navigate to the user profile settings page.
2. Locate and click on the "Monetization Options" section.
3. Review the displayed eligibility criteria and requirements.
4. Click on the "Apply Now" or similar button to initiate the application process.
5. Verify that the application form loads appropriately and prompts the user to provide required information.

****Postconditions:** Application is submitted.**

****Expected Result:****

The user successfully navigates to the monetization application form and is able to start the process.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing, User Acceptance Testing**

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

****_____****

****Test Case ID:** TC_MNT_002**

****Test Case Title:** Verify that monetization options include advertising revenue**

****Test Case Description:** This test case verifies that advertising revenue generation is offered as a monetization option to eligible users.**

****Test Suite:** User Monetization Methods**

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

****Preconditions:****

* User has successfully applied for and been approved for monetization.

* User has created at least one piece of eligible content (e.g., post, video).

****Test Data:****

Relevant advertising content categories for testing

****Test Steps:****

1. Browse the platform and locate a post or video created by the test user.
2. Verify that the post or video incorporates display advertisements.
3. Examine the advertisements displayed and ensure they are relevant to the user's content or target audience.

****Postconditions:****

No specific post-conditions

****Expected Result:**** Advertisements are displayed on the user's content, indicating the availability of advertising revenue as a monetization option.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, Regression Testing

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

****_____****

****Test Case ID:**** TC_MNT_003

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

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

****Test Suite:**** Payment Gateway Integration

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

****Preconditions:****

* User has successfully applied for and been approved for monetization.

* User has generated a nominal amount of revenue through approved monetization methods.

****Test Data:**** Test payment information (e.g., virtual card)

****Test Steps:****

1. Navigate to the user earnings dashboard or payment settings.
2. Verify that the earnings are accurately displayed based on the generated revenue.
3. Initiate a withdrawal request using the test payment information.
4. Monitor the status of the withdrawal request and confirm successful payment processing (may require simulated delay).

****Postconditions:****

* Funds are successfully transferred to the linked payment account (virtually, for testing).

****Expected Result:**** The user is able to initiate and process a successful withdrawal of earned funds through the platform's payment system.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Security Testing

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

****_____****

****Test Case ID:**** TC_MNT_004

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

****Test Case Description:**** This test case verifies the availability and functionality of detailed earnings

reports for monetized users.

****Test Suite:**** User Analytics & Reports

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

****Preconditions:****

- * User has successfully applied for and been approved for monetization.

- * User has generated revenue through approved monetization methods.

****Test Data:****

No specific test data needed.

****Test Steps:****

1. Navigate to the user earnings dashboard or analytics section.
2. Verify that detailed earnings reports are available, showcasing income from different monetization sources.
3. Examine the report for information like date ranges, revenue breakdowns, payment methods, and possibly other relevant metrics.
4. Filter or sort the report to validate the functionality and accuracy of the data presentation.

****Postconditions:****

No specific post-conditions

****Expected Result:**** Users have access to comprehensive earnings reports that display a detailed breakdown of their revenue streams and financial activity within the platform.

****Severity:** Major**

****Priority:** Medium**

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

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

****_____****

****Test Case ID:** TC_MNT_005**

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

****Test Case Description:** This test case verifies that the platform communicates monetization eligibility requirements clearly and understandably to users.**

****Test Suite:** Platform UI & User Experience**

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

****Preconditions:****

*** User is logged in and navigating the platform.**

****Test Data:****

No specific test data needed.

****Test Steps:****

1. Navigate to the section related to monetization options (e.g., user profile settings, help center).
2. Locate and review the information about eligibility requirements for monetization.
3. Assess the clarity and comprehensibility of the language used to describe the requirements.
4. Evaluate the organization and presentation of the information, ensuring it is easy to understand.

****Postconditions:****

No specific post-conditions.

****Expected Result:****

Users can readily find detailed and clearly stated eligibility requirements for monetization upon accessing the relevant section of the platform.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Usability 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 5 test cases for the "Explore or Discover Tab" feature, designed to cover the provided acceptance criteria:

****Test Case ID:**** TC_EXPLORE_001

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

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

****Test Suite:**** User Interface - Explore Tab

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

****Preconditions:****

- * User is logged into the platform.
- * The platform has populated trending content data.

****Test Data:**** N/A

****Test Steps:****

1. Navigate to the Explore tab.
2. Verify that the tab displays a list of trending content.
3. Examine the content and confirm it represents popular posts, accounts, and topics.

****Postconditions:****

- * No modifications are made to user data.
- * The user is returned to the Explore tab.

****Expected Result:**** Trending content is prominently displayed on the Explore tab.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** System Testing, Functional Testing

****Test Case ID:**** TC_EXPLORE_002

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

****Test Case Description:**** This test case checks if the Explore tab recommends content tailored to the user's interests, considering their past engagement and profile information.

****Test Suite:**** User Interface - Explore Tab

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

****Preconditions:****

* User is logged into the platform.

* The user has interacted with content in specific categories (e.g., liked posts about technology, followed accounts in the travel niche).

****Test Data:****

* Set up a user profile with defined interests.

* Log in as this user.

****Test Steps:****

1. Navigate to the Explore tab.

2. Observe the recommended content and note the categories or topics.

3. Compare the recommended content to the user's defined interests and past engagement.

****Postconditions:****

* User is returned to the Explore tab.

****Expected Result:**** The Explore tab recommends content primarily aligned with the user's declared interests and past engagement patterns.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** System Testing, Functional Testing**

****Test Case ID:** TC_EXPLORE_003**

****Test Case Title:** Verify that users can filter content by category or topic on the Explore tab.**

****Test Case Description:** This test case confirms the functionality of content filters on the Explore tab,**

allowing users to refine their content discovery based on specific categories.

****Test Suite:**** User Interface - Explore Tab

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

****Preconditions:****

- * User is logged into the platform.

- * The platform provides a diverse set of content across multiple categories.

****Test Data:**** N/A

****Test Steps:****

1. Navigate to the Explore tab.
2. Identify the available content filters (e.g., "News," "Technology," "Lifestyle").
3. Select a specific category filter.
4. Observe the displayed content and ensure it aligns with the selected category.

****Postconditions:****

- * User is returned to the Explore tab. The selected filter remains active.

****Expected Result:**** Filtering by category effectively narrows down the displayed content to match the chosen category.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** System Testing, Functional Testing

****Test Case ID:**** TC_EXPLORE_004

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

****Test Case Description:**** This test case ensures that the Explore tab dynamically refreshes and incorporates

new trending content in a timely manner.

****Test Suite:**** User Interface - Explore Tab

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

****Preconditions:****

* User is logged into the platform.

****Test Data:****

* A new trending post should be scheduled at a specific time.

****Test Steps:****

1. Open the Explore tab and note the initial set of trending content.
2. Set a timer for a predetermined interval (e.g., 15 minutes).
3. At the end of the interval, return to the Explore tab.
4. Compare the content to the initial set, looking for new entries.

****Postconditions:****

* No modifications are made to user data.

****Expected Result:**** The Explore tab reflects at least one new trending item or update within the specified interval.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** System Testing, Functional Testing

****Test Case ID:**** TC_EXPLORE_005

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

****Test Case Description:**** This test case checks if users have the option to save or follow discovered content from the Explore tab for later viewing or continued engagement.

****Test Suite:**** User Interface - Explore Tab

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

****Preconditions:****

* User is logged into the platform.

****Test Data:**** Content on the Explore tab (posts, accounts, topics)

****Test Steps:****

1. Navigate to the Explore tab.
2. Locate a post, account, or topic that the user wants to save or follow.
3. Identify the save or follow button (e.g., a bookmark icon or "Follow" button) associated with the content.
4. Click the save or follow button.
5. Verify that the content is successfully saved or followed (e.g., it appears in a saved list or a new follower notification is displayed).

****Postconditions:****

* The user is returned to the Explore tab. Saved content is accessible through the user's profile or settings.

****Expected Result:**** Users can easily save or follow content found on the Explore tab, allowing for personalized recommendation and future reference.

Severity: Major

Priority: Medium

Type of Testing: System Testing, Functional Testing

Please note that these are examples, and you may need to adjust them based on the specific design and functionality of the "Explore or Discover Tab" in your social media platform.

Data - 16

Project name -

Social Media Platform

Project description -

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

Feature name -

Accessibility Features

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Social Media Platform Accessibility Features

****Test Case ID:**** AC-SM-1

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

****Test Case Description:**** This test case verifies that all images on the platform have descriptive alt text attributes, ensuring screen reader accessibility.

****Test Suite:**** Accessibility Testing

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

****Preconditions:****

- Access to platform with browser enabled for screen reader use.
- User is logged into the platform.
- Platform displays various types of images (profile pictures, posts, attachments).

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

****Test Steps:****

1. Enable the screen reader on the browser.
2. Navigate to different sections of the platform, such as the News Feed, User Profiles, and Posts.
3. Identify all images displayed on the pages.
4. Use the screen reader to read the alt text associated with each image.
5. Verify that all images have meaningful and descriptive alt text.

****Postconditions:**** Screen reader is disabled.

****Expected Result:**** All images display appropriate alt text that accurately describes the visual content.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional & Accessibility Testing**

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

****Test Case ID:** AC-SM-2**

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

****Test Case Description:** This test case verifies that the platform can be effectively navigated and understood by users employing screen reader software.**

****Test Suite:** Accessibility Testing**

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

****Preconditions:****

- Access to platform with browser enabled for screen reader use.
- User is logged into the platform.
- Known screen reader software (e.g., JAWS, NVDA) is installed and configured.

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

****Test Steps:****

1. Open the platform using the screen reader software.
2. Navigate through the platform's menus, navigation bars, and various sections using keyboard and screen reader commands.
3. Verify that all interactive elements (buttons, links, forms) are accessible and their functionality is conveyed clearly by the screen reader.
4. Test user actions like clicking, submitting forms, and interacting with content elements.
5. Ensure the screen reader accurately interprets headings, labels, and other structural elements.

****Postconditions:** Screen reader is disabled.**

****Expected Result:** The platform is fully navigable and understandable using screen reader software, providing an accessible experience for users.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional & Accessibility Testing**

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

****Test Case ID:** AC-SM-3**

****Test Case Title:** Verify that keyboard shortcuts are functional for navigation.**

****Test Case Description:** This test case validates the functionality of keyboard shortcuts for users who prefer or require navigating the platform using the keyboard.**

****Test Suite:** Accessibility Testing**

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

****Preconditions:****

- Access to platform with keyboard enabled.
- User is logged into the platform.

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

****Test Steps:****

1. Navigate to various sections of the platform using only the keyboard, utilizing the available shortcuts (e.g., Tab, Enter, Arrow Keys, Esc).
2. Test shortcuts for common actions like navigating menus, opening links, submitting forms, etc.
3. Ensure that accessibility shortcuts are clearly documented within the platform's interface.

****Postconditions:** No specific postconditions required.**

****Expected Result:** Keyboard navigation is intuitive and efficient, allowing users to navigate and interact with the platform without a mouse.**

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Functional & Accessibility Testing**

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

****Test Case ID:** AC-SM-4**

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

****Test Case Description:** This test case ensures that users can adjust the platform's color contrast and font size for improved readability and accessibility.**

****Test Suite:** Accessibility Testing**

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

****Preconditions:****

- Access to platform with browser enabled.
- User is logged into the platform.

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

****Test Steps:****

1. Navigate to the platform's accessibility settings or preferences.
2. Test the color contrast options by adjusting the settings and visually verifying the changes.
3. Test the font size options by increasing and decreasing the font size and verifying the changes in text display.
4. Verify that the adjustments are saved and applied consistently across all platform sections and content.

****Postconditions:** Accessibility settings are returned to default values.**

****Expected Result:** Users can easily adjust color contrast and font size to suit their visual needs and preferences, ensuring comfortable readability.**

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Functional & Accessibility Testing**

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

****Test Case ID:** AC-SM-5**

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

****Test Case Description:** This test case checks the clarity and comprehensiveness of the documentation regarding the platform's accessibility features.**

****Test Suite:** Accessibility Testing**

****Test Priority:** Low**

****Preconditions:****

- Access to the platform's help center or documentation section.

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

****Test Steps:****

1. Locate the documentation regarding accessibility features within the platform's help center.
2. Review the documentation for clarity, completeness, and accuracy.
3. Verify that the documentation covers all implemented accessibility features.
4. Assess whether the instructions for using accessibility features are easily understood.
5. Check if the documentation provides information on keyboard shortcuts, screen reader compatibility, and color contrast options.

****Postconditions:** No specific postconditions required.**

****Expected Result:** The platform's documentation clearly and comprehensively describes all its accessibility features, enabling users to understand and utilize them effectively.**

****Severity:** Minor**

****Priority:** Low**

****Type of Testing:** Non-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:

Test Cases for Safety and Moderation Tools

Project Context: Social Media Platform

Feature: Safety and Moderation Tools

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

Test Case 1:

Test Case ID: SM-SAFETY-TC01

Test Case Title: Verify that content moderation algorithms filter hate speech

Test Case Description: This test case verifies the effectiveness of content moderation algorithms in identifying and filtering hate speech from user posts.

Test Suite: Safety & Moderation

Test Priority: High

Preconditions:

- * A functioning user account with posting privileges is available.

- * The system is initialized with pre-defined hate speech keywords and phrases.

****Test Data:****

- * Sample posts containing various forms of hate speech.

****Test Steps:****

1. Log in to the platform using the pre-created account.
2. Compose a new post containing a pre-defined hate speech phrase.
3. Attempt to publish the post.
4. Observe the system's response to the hate speech content.

****Postconditions:****

- * The hate speech post should be flagged or prevented from being published.
- * An appropriate message should be displayed to the user explaining the reason for the post being blocked.

****Expected Result:**** The content moderation algorithm successfully identifies and filters the hate speech from the user post.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

****Test Case 2:****

****Test Case ID:** SM-SAFETY-TC02**

****Test Case Title:** Verify that users can effectively report harmful content**

****Test Case Description:** This test case ensures the functionality of the reporting mechanism for users to flag inappropriate content.**

****Test Suite:** Safety & Moderation**

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

****Preconditions:****

- * A functioning user account with posting privileges is available.
- * A pre-defined piece of potentially harmful content is uploaded to the platform.

****Test Data:****

- * User account details for reporting the content.

****Test Steps:****

1. Log in to the platform using the pre-created account.
2. Locate the pre-defined harmful content.
3. Click on the "Report" button associated with the content.
4. Select the appropriate reason for reporting from the provided options.
5. Submit the report.

****Postconditions:****

- * Confirmation message is displayed to the user indicating successful submission of the report.
- * The reported content is flagged for review by the moderation team.

****Expected Result:**** The reporting system successfully captures the user's feedback and flags the content for review.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

Please note that this is just a starting point. Additional test cases can be created to cover various other aspects of the safety and moderation tools, including:

- * Testing different types of inappropriate content (e.g., nudity, violence, harassment)
- * Testing the effectiveness of comment filters
- * Testing the blocking functionality for users and keywords
- * Testing the enforcement of community guidelines
- * Testing the reporting process from different user perspectives (e.g., reporting their own content, reporting another user)
- * User interface testing for reporting and blocking features

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 5 test cases for the "Multi-Language Support" feature, following the provided template:

****Test Case ID:** MTL-001**

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

****Test Case Description:** This test case verifies the functionality of the language selection feature, ensuring users can choose their desired language from a list of supported options.**

****Test Suite:** User Interface Testing**

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

****Preconditions:****

- * A new user account is created.
- * The platform supports at least 3 different languages.

****Test Data:** A list of available languages (e.g., English, Spanish, French)**

****Test Steps:****

1. Navigate to the user profile settings.
2. Locate the "Language" preference setting.
3. Select a language from the provided list other than the default language.
4. Observe if the platform interface language changes to the selected language.
5. Attempt to access different sections of the platform (e.g., news feed, messaging, profile) and verify that the language is consistent throughout.

****Postconditions:** Language preference should be saved.**

****Expected Result:** The platform interface should successfully change to the selected language, and all elements should be displayed in the chosen language.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing, User Interface Testing**

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

****Test Case ID:** MTL-002**

****Test Case Title:** Verify that content translation features are available.**

****Test Case Description:** This test case confirms the functionality of the content translation feature, ensuring that posts and messages can be translated into different languages.**

****Test Suite:** Content Management Testing**

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

****Preconditions:****

* A user account is logged in.

* A post or message is created in a specific language (e.g., English).

****Test Data:** A sample post or message in a non-English language.**

****Test Steps:****

1. Create a post or message in a language different from the user's selected interface language.
2. View the post or message to verify the original language is displayed correctly.
3. Observe if a translation option is available for the post or message.
4. Select the desired translation language (different from the original language).
5. Verify that the content is translated accurately into the selected language.

****Postconditions:** Original post/message should remain unchanged.**

****Expected Result:** The content translation feature should be functional, providing an accurate translation of the post or message into the selected language.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing, User Interface Testing**

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

****Test Case ID:** MTL-003**

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

****Test Case Description:** This test case ensures that language preferences set by users are saved and**

applied consistently across all platform sessions and devices.

****Test Suite:**** User Authentication & Settings Testing

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

****Preconditions:****

- * A user account is logged in.

- * The platform supports multiple languages.

****Test Data:**** Two different languages.

****Test Steps:****

1. Change the user's language preference to Language A.
2. Log out of the platform.
3. Log back in with the same user account.
4. Verify that the platform interface reflects the chosen Language A.
5. Repeat steps 1-4 using Language B as the preferred language.

****Postconditions:**** User should be able to switch language preferences easily.

****Expected Result:**** The language preference selected by the user should be saved and consistently applied across all sessions and devices.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, User Authentication Testing

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

****Test Case ID:**** MTL-004

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

****Test Case Description:**** This test case verifies the breadth of supported languages, ensuring the platform caters to a diverse global user base.

****Test Suite:**** Internationalization & Localization Testing

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

****Preconditions:****

- * A user account is logged in.
- * The platform supports a predetermined number of languages (e.g., 10 or more).

****Test Data:**** List of supported languages from the platform.

****Test Steps:****

1. Iterate through the list of supported languages.
2. For each language, navigate to the user profile settings and select the language.
3. Verify that the platform interface seamlessly switches to the selected language.
4. Observe if all platform functionalities, such as news feeds, messaging, and menus, are correctly translated and displayed.

****Postconditions:**** System should return to the initial settings.

****Expected Result:**** The platform should successfully switch to and display all supported languages accurately, ensuring a smooth and localized experience for users across different regions and cultures.

****Severity:**** Minor

****Priority:**** Low

****Type of Testing:**** Functional Testing, Internationalization Testing

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

Remember to adapt these test cases to your specific implementation details and consider additional scenarios to ensure comprehensive coverage.

Project name -

Social Media Platform

Project description -

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

Feature name -

Cross-Platform Integration

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Cases for Cross-Platform Integration Feature

Test Case ID: TC_XPI_001

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

****Test Case Description:**** This test case validates the functionality allowing users to connect their social media platform accounts with external services. The test will confirm successful linking with at least two different supported platforms.

****Test Suite:**** Cross-Platform Integration

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

****Preconditions:****

- * User has a registered account on the social media platform.
- * User is logged into the social media platform.
- * The target external platforms are supported by the cross-platform integration feature.

****Test Data:**** Valid user credentials for the social media platform and target external platforms.

****Test Steps:****

1. Navigate to the user settings section of the platform.
2. Locate and select the "Connect Accounts" option.
3. Choose one supported external platform from the list.
4. Enter the required login credentials for the chosen external platform.
5. Authorize the connection request from the social media platform.
6. Repeat steps 3-5 for a second supported external platform.

****Postconditions:****

- * User account is successfully linked with both external platforms.

- * Link confirmation messages are displayed on the screen.
- * Linked accounts are reflected in the user profile settings.

****Expected Result:**** Success: Both external platform accounts are successfully linked to the user's profile.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

****Test Case ID:**** TC_XPI_002

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

****Test Case Description:**** This test case checks the functionality of cross-posting content from the platform to connected external social media accounts.

****Test Suite:**** Cross-Platform Integration

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

****Preconditions:****

- * User has a registered account on the platform and at least one external platform linked (as per TC_XPI_001).
- * User has created a post on the platform (text, image, or video).

****Test Data:**** Sample content (text, image, or video) to be cross-posted.

****Test Steps:****

1. Compose and publish a post on the platform.
2. Navigate to the post's sharing options.
3. Select the option to share the post on the linked external platform.
4. Click "Confirm" or "Share" to complete the cross-posting process.
5. Verify the post is visible on the external platform user's feed.

****Postconditions:****

- * The cross-posted content appears on the user's external platform profile.
- * Platform-specific formatting is preserved in the cross-posted content.
- * User account is not negatively impacted by the cross-posting action.

****Expected Result:**** Success: The posted content appears on the linked external platform without errors.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

****Test Case ID:**** TC_XPI_003

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

****Test Case Description:**** This test case verifies the functionality of single sign-on (SSO) with external platforms for user authentication.

****Test Suite:**** Cross-Platform Integration

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

****Preconditions:****

* User is registered on the platform and has linked their account with an external platform supporting SSO (e.g., Google, Facebook).

* SSO is enabled for the chosen external platform.

****Test Data:**** Valid credentials for the chosen external platform account (e.g., email and password).

****Test Steps:****

1. Access the platform's login page.
2. Select the SSO option for the chosen external platform.
3. Authenticate with the external platform's credentials (login information).
4. Verify successful platform login and a redirect to the user's profile.

****Postconditions:****

- * User is logged into the platform without needing platform-specific credentials.
- * SSO token is successfully validated and user information is retrieved.
- * User is redirected to their profile page after SSO authentication.

****Expected Result:**** Success: User is logged into the platform seamlessly using SSO credentials from an external platform.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

****Test Case ID:** TC_XPI_004**

****Test Case Title:** Verify that integration with external media sources is available.**

****Test Case Description:** This test case verifies the functionality to embed and display content from external media sources (e.g., YouTube, Instagram) within posts on the platform.**

****Test Suite:** Cross-Platform Integration**

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

****Preconditions:****

* User has a registered account on the platform.

****Test Data:****

* URLs of publicly available content from supported external media sources.

****Test Steps:****

1. Compose a new post on the platform.
2. Locate and select the option to embed external media (e.g., "Insert Media" or "Embed Video").
3. Paste the URL of a video or image from a supported external media source.
4. Preview the embedded media within the post.
5. Publish the post.

****Postconditions:****

- * Embedded media displays correctly within the post.
- * User can interact with basic media controls (e.g., play/pause, volume adjustment).
- * Proper attribution is displayed for the external media source.

****Expected Result:**** Success: External media is successfully embedded and displayed within the platform post.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, System Testing

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

****Test Case ID:**** TC_XPI_005

****Test Case Title:**** Verify seamless content sharing experience across platforms.

****Test Case Description:**** This test case assesses the ease and functionality of sharing content from the platform to different supported networks. It covers scenarios of sharing various content types and handling potential errors or limitations.

****Test Suite:**** Cross-Platform Integration

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

****Preconditions:****

* User has a registered account on the platform and at least one external platform linked.

****Test Data:****

- * Sample text post
- * Sample image post
- * Sample video post

****Test Steps:****

1. Create each sample content type (text, image, video) on the platform.
2. Navigate to the sharing options for each post.
3. Select the target external platform for sharing.
4. Confirm the sharing action.

****Postconditions:****

- * Shared content appears on the target external platform.
- * Platform-specific formatting is preserved.
- * User receives clear confirmation messages upon successful share.

****Expected Result:**** Success: Content sharing is seamless across different platforms and content types with no significant formatting issues.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

****Test Case Approach:**** Positive 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 5 detailed test cases for the "Gamification and Rewards" feature, following the provided template:

Test Case 1

* **Test Case ID:** TC_GAM_001

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

* **Test Case Description:** This test case aims to verify that users successfully earn points for creating various types of content, such as text posts, photos, and videos.

* **Test Suite:** Gamification Feature

* **Test Priority:** High

* **Preconditions:**

- * A user account is created and logged in.

- * The gamification system is enabled on the platform.

* **Test Data:** Pre-written content (text, image, video), user account details

* **Test Steps:**

1. Navigate to the content creation page.
2. Create a new text post with relevant content.
3. Publish the post.
4. Check the user's profile page or points dashboard to confirm the awarded points.
5. Repeat steps 1-4 for creating a photo post and a video post.

* **Postconditions:**

- * The user account remains active and logged in.
- * Points earned for each post are accurately reflected.
- * **Expected Result:** The user should earn a predetermined number of points for each type of content published, as defined in the gamification rules.
- * **Severity:** Critical
- * **Priority:** High
- * **Type of Testing:** Functional Testing
- * **Test Case Approach:** Positive

Test Case 2

- * **Test Case ID:** TC_GAM_002
- * **Test Case Title:** Verify that users can earn badges for achieving milestones
- * **Test Case Description:** This test case verifies that users receive badges when they accomplish specific milestones within the platform, such as reaching a certain number of followers or posts.
- * **Test Suite:** Gamification Feature
- * **Test Priority:** Medium
- * **Preconditions:**
 - * A user account is created and logged in.
 - * The gamification system (badges and milestones) is active.
- * **Test Data:** User account data, target follower count, target post count.
- * **Test Steps:**
 1. Create a new user account.
 2. Start posting regularly.
 3. Continue interacting with the platform to gather followers.
 4. Reach the predetermined follower and post milestones.
 5. Verify the awarded badges in the user's profile or achievement section.

* **Postconditions:**

- * The user account remains active.

- * The earned badges are correctly displayed.

* **Expected Result:** The user should receive the designated badges upon achieving the specified follower or post milestones.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 3****

* **Test Case ID:** TC_GAM_003

* **Test Case Title:** Verify that leaderboards correctly display rankings

* **Test Case Description:** This test case verifies that leaderboards accurately display the top users based on their point totals or achievement counts.

* **Test Suite:** Gamification Feature

* **Test Priority:** High

* **Preconditions:**

- * Users are actively earning points and achieving milestones.

- * The leaderboard system is functioning.

* **Test Data:**

* **Test Steps:**

1. Access the leaderboard section of the platform.
2. Verify that users are ranked based on their point scores or achievements.
3. Confirm that the displayed rankings are accurate and reflect the current point standings.
4. Repeat steps 1-3 at different times to ensure real-time leaderboard updates.

* **Postconditions:**

- * The leaderboard remains accessible and functional.

* **Expected Result:** The leaderboard displays users correctly sorted by their achievements and points, with the highest-ranking users at the top.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Functional Testing

Test Case 4

* **Test Case ID:** TC_GAM_004

* **Test Case Title:** Verify that challenges are available and offer rewards

* **Test Case Description:** This test case verifies that challenges are presented to users, and successful completion leads to the awarding of rewards.

* **Test Suite:** Gamification Feature

* **Test Priority:** Medium

* **Preconditions:**

- * A user account is created and logged in.

- * Challenges are implemented within the gamification system.

* **Test Data:** Challenge description, reward type (virtual goods, discounts, etc.)

* **Test Steps:**

1. Navigate to the challenges section of the platform.
2. Identify an available challenge and understand its requirements.
3. Complete the challenge by fulfilling the specified criteria.
4. Verify the reward is credited to the user's account (e.g., virtual goods added).

* **Postconditions:**

* The user's account reflects the awarded reward.

* **Expected Result:** The challenge successfully completes, and the user receives the promised reward as defined in the challenge description.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC_GAM_005

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

* **Test Case Description:** This test case confirms that users can access a dedicated section to monitor their progress, earned points, badges, and overall achievements within the gamification system.

* **Test Suite:** Gamification Feature

* **Test Priority:** Medium

* **Preconditions:**

* A user account is created and logged in.

* **Test Data:** Test User data

* **Test Steps:**

1. Navigate to the user's profile page or dedicated achievement section.

2. Verify the display of points earned, a list of badges acquired, and a record of past accomplishments (including challenges completed and milestones reached).

* **Postconditions:**

* The user's profile remains accessible.

* **Expected Result:** The user's achievements are clearly presented and reflect their activities and progress within the gamification system.

* **Severity:** Major

* **Priority:** Medium

* **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:

Test Cases for Task Management Tool - Task Creation Feature

Test Case ID: TC_TASK_001

Test Case Title: Verify that users can enter a task title, description, due date, and priority level.

Test Case Description: This test case verifies the functionality of entering all required task details upon creation.

Test Suite: Task Management

Test Priority: High

Preconditions:

- * User is logged into the Task Management Tool.
- * An empty task list is available.

Test Data:

- * Task Title: "Project X Brainstorming"
- * Task Description: "Brainstorm ideas for new project X features."
- * Due Date: 2024-03-15
- * Priority Level: High

Test Steps:

1. Navigate to the "Add New Task" page.
2. Enter "Project X Brainstorming" in the "Task Title" field.
3. Type "Brainstorm ideas for new project X features." in the "Description" field.
4. Select "2024-03-15" as the "Due Date".
5. Choose "High" as the "Priority Level".

6. Click the "Create Task" button.

****Postconditions:****

- * A new task with the specified details is created and displayed on the task list.
- * The system has saved the task details in its database.

****Expected Result:****

The task is successfully created and displayed on the task list with the entered title, description, due date, and priority level.

****Severity:** Critical**

****Priority:** High**

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

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

****Test Case ID:** TC_TASK_002**

****Test Case Title:** Verify that tasks can be assigned to specific team members.**

****Test Case Description:** This test case verifies the functionality of assigning a created task to a specific team member.**

****Test Suite:** Task Management**

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

****Preconditions:****

- * User is logged into the Task Management Tool.
- * At least two users exist with different login credentials.
- * An empty task list is available.

****Test Data:****

- * Task Title: "Marketing Campaign Launch"

* Task Description: "Plan and execute the marketing campaign launch."

* Due Date: 2024-03-29

* Priority Level: Medium

* Assignee: "John Doe" (existing user in the system)

****Test Steps:****

1. Navigate to the "Add New Task" page.
2. Follow steps 2-5 from Test Case TC_TASK_001.
3. Select "John Doe" from the "Assignee" dropdown list.
4. Click the "Create Task" button.

****Postconditions:****

- * A new task with the specified details, including the assignee, is created and displayed on the task list.
- * The task is associated with the selected user in the system's database.

****Expected Result:****

The task is successfully created and displays "John Doe" as the assigned person.

****Severity:** Critical**

****Priority:** High**

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

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

****... continue with more test cases to cover:****

- * Editing task details after creation
- * System validating required fields

Data - 22

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Assignment

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

members.', 'Task assignment history should be tracked.']

Generated Test Cases:

Test Cases for Task Assignment Feature

****Test Case ID:** TC_TASK_ASSIGNMENT_001**

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

****Test Case Description:** This test case verifies the functionality of assigning tasks to specific team members from the user interface.**

****Test Suite:** Task Management - Assignment**

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

****Preconditions:****

- * User has access to the task management system.
- * A project exists within the system.
- * At least two team members are present in the system.

****Test Data:****

- * Project name: "Test Project"
- * Task title: "Test Task 1"
- * Team members: "UserA", "UserB"

****Test Steps:****

1. Log in as a user with administrative privileges.
2. Navigate to the "Test Project" within the system.
3. Create a new task with the title "Test Task 1."
4. In the task creation form, select "UserA" from the list of available team members as the assignee.
5. Save the task.
6. Verify that "UserA" is displayed as the assignee for "Test Task 1" in the task details.

****Postconditions:****

* Task "Test Task 1" is created and assigned to "UserA"

****Expected Result:**** The task should be successfully assigned to "UserA" and reflected in the task details.

****Severity:**** Blocker

****Priority:**** High

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

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

****Test Case ID:**** TC_TASK_ASSIGNMENT_002

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

****Test Case Description:**** This test case ensures that an assignee receives a notification when a new task is assigned to them.

****Test Suite:**** Task Management - Assignment

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

****Preconditions:****

- * User has access to the task management system.
- * A project exists within the system.
- * At least one team member is active in the system with an email address.

****Test Data:****

- * Project name: "Test Project"
- * Task title: "Test Task 2"
- * Assignees: "UserB"

****Test Steps:****

1. Log in as a user with administrative privileges.
2. Navigate to the "Test Project" within the system.
3. Create a new task titled "Test Task 2" and assign it to "UserB."
4. Log out of the system.
5. As "UserB," log in to the system.
6. Verify that UserB receives a notification about the assigned task "Test Task 2."

****Postconditions:****

- * "UserB" has been notified about the assigned task.

****Expected Result:**** UserB should receive a notification containing details about the assigned task, "Test

Task 2."

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TC_TASK_ASSIGNMENT_003

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

****Test Case Description:**** This test case confirms that assigned tasks appear correctly in the assignee's task list within the user interface.

****Test Suite:**** Task Management - Assignment

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

****Preconditions:****

* User has access to the task management system.

* A project exists within the system.

* A task has been previously assigned to a team member.

****Test Data:****

* Project name: "Test Project"

* Task title: "Test Task 3"

****Test Steps:****

1. Log in as the user assigned to "Test Task 3."
2. Navigate to the ?My Tasks? section.
3. Verify that "Test Task 3" is listed in the assigned tasks section.

****Postconditions:****

* "Test Task 3" is displayed in the assigned tasks list.

****Expected Result:**** The task "Test Task 3" will be visible in the assignee's task list.

****Severity:**** Major

****Priority:**** Medium

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

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

****Note:**** The remaining test cases (TC_TASK_ASSIGNMENT_004 and TC_TASK_ASSIGNMENT_005) for reassignment history and various edge cases should follow a similar structure, ensuring comprehensive

coverage of all acceptance criteria.

Data - 23

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Deadline Reminders

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are 5 test cases designed for the "Task Deadline Reminders" feature of your Task Management Tool,

following the provided template.

****Test Case 1****

* ****Test Case ID:**** TC_DeadlineReminders_001

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

* ****Test Case Description:**** This test case verifies the functionality of setting reminders for tasks based on specified deadlines. It will involve creating a task, setting a deadline, and confirming that a reminder is generated and delivered according to the set deadline.

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

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

* ****Preconditions:****

- * A user account is created and logged in.

- * A new task is created.

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

- * Task title: "Project Proposal"

- * Deadline date: "2024-03-15 10:00 AM"

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

1. Navigate to the task details page for "Project Proposal".

2. Locate the "Reminder" section within the task details.

3. Set a reminder for the task's deadline.

4. Verify that the reminder setting is successfully saved.

5. Within the specified time frame (e.g., one day before the deadline), check for a notification or reminder regarding the "Project Proposal" task.

* ****Postconditions:****

- * The reminder should be cancelled.

* ****Expected Result:****

The system should send a reminder notification to the user one day before the specified deadline for the task "Project Proposal."

****Test Case 2****

* ****Test Case ID:**** TC_DeadlineReminders_002

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

* ****Test Case Description:**** This test case ensures that users can set different levels of urgency for task reminders and that the system differentiates between them accordingly.

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

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

* ****Preconditions:****

- * A user account is created and logged in.

- * A new task is created.

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

- * Task title: "Meeting Preparation"

- * Deadline date: "2024-03-16 09:00 AM"

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

1. Navigate to the task details page for "Meeting Preparation".
2. Locate the "Reminder" section within the task details.
3. Select a high-urgency level for the reminder.
4. Select a low-urgency level for a second reminder.
5. Verify that the reminder settings are saved with their respective urgency levels.

* ****Postconditions:****

- * No further actions required.

* ****Expected Result:**** The system should differentiate between the high and low urgency levels, potentially delivering the high-urgency reminder earlier or through a more prominent notification method.

****Test Case 3****

* **Test Case ID:** TC_DeadlineReminders_003

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

* **Test Case Description:** This test case confirms the core functionality of the reminder system by ensuring that notifications are sent for tasks approaching their deadlines.

* **Test Suite:** Task Management Feature

* **Test Priority:** High

* **Preconditions:**

- * A user account is created and logged in.

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

* **Test Data:**

- * Task title: "Report Submission"

- * Deadline date: "2024-03-18 12:00 PM"

* **Test Steps:**

1. Set a reminder for the "Report Submission" task.
2. Allow sufficient time for the reminder to be triggered (e.g., 1 hour before the deadline).
3. Verify that a notification is received through the chosen delivery method (e.g., email, in-app).

* **Postconditions:**

- * The notification should be read.

- * The user should be able to see the corresponding reminder in the app.

* **Expected Result:**

The system should successfully send a notification one hour before the deadline date of "Report Submission" through the selected notification method.

****Test Case 4****

* **Test Case ID:** TC_DeadlineReminders_004

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

* **Test Case Description:** This test case validates that users have control over how they receive reminders and can select their preferred delivery method.

* **Test Suite:** Task Management Feature

* **Test Priority:** Medium

* **Preconditions:**

- * User account is created and logged in.

- * User has access to the reminder settings page.

* **Test Data:**

- * Choose two different delivery methods (e.g., email and in-app).

* **Test Steps:**

1. Navigate to the user settings or notification preferences.

2. Look for options to customize reminder delivery methods.

3. Select "email" as the primary reminder method.

4. Set up a reminder for a task and verify that a test email is sent.

5. Change the delivery method to "in-app" for a different task and confirm that a reminder appears within the application.

* **Postconditions:**

- * Settings should revert to previously selected preference.

* **Expected Result:**

The system should successfully send reminders through the selected delivery method for each individual task.

Test Case 5

* **Test Case ID:** TC_DeadlineReminders_005

* **Test Case Title:** Verify that reminder settings can be adjusted or disabled.

* **Test Case Description:** This test case ensures users have the flexibility to modify or turn off reminder settings for tasks as needed.

* **Test Suite:** Task Management Feature

* **Test Priority:** Medium

* **Preconditions:**

- * A user account is created and logged in.

- * A task with an active reminder is created.

* **Test Data:**

- * Task title: "Project Update"

- * Existing reminder configured.

* **Test Steps:**

1. Navigate to the task details page for "Project Update".
2. Locate the "Reminder" section and change the reminder time or frequency.
3. Disable the reminder completely for the task.
4. Verify that the updated reminder settings are saved.

* **Postconditions:**

- * Reminder settings should be kept aligned with user expectations.

* **Expected Result:**

The system should effectively apply any changes made to the reminder settings. Disabled reminders should no longer trigger notifications.

Data - 24

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are 5 test cases for the "Task Prioritization" feature, structured according to your template:

Test Case 1:

****Test Case ID:** TC_TASK_PRIOR_01**

****Test Case Title:** Verify that users can set priority levels for tasks.**

****Test Case Description:** This test case verifies that users can successfully assign high, medium, or low priority levels to individual tasks.**

****Test Suite:** Features/TaskManagement/TaskPrioritization**

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

****Preconditions:****

* A user is logged into the Task Management Tool.

* A new task is created.

****Test Data:****

* Task title: "Project Proposal"

****Test Steps:****

1. Navigate to the task list.
2. Select the newly created task "Project Proposal."
3. Locate the "Priority" field.
4. Select "High" from the priority dropdown menu.
5. Save the changes to the task.

****Postconditions:****

* The task status should be updated.

****Expected Result:** The task "Project Proposal" will be successfully assigned a priority level of "High."**

****Severity:** Major**

****Priority:** High**

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

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

Test Case 2:

****Test Case ID:** TC_TASK_PRIOR_02**

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

****Test Case Description:** This test case ensures that the task list can be effectively sorted by priority level (High, Medium, Low).**

****Test Suite:** Features/TaskManagement/TaskPrioritization**

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

****Preconditions:****

* A user is logged into the Task Management Tool.

* Both high and low priority tasks exist in the task list.

****Test Data:****

* Several tasks with varying priority levels.

****Test Steps:****

1. Navigate to the task list.
2. Click on the "Priority" column header to sort the tasks.
3. Verify that tasks are listed in descending order of priority (High, Medium, Low).

****Postconditions:****

- * Task list should remain sorted.

****Expected Result:**** Tasks are correctly sorted from High to Low priority.

****Severity:**** Major

****Priority:**** Medium

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

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

Test Case 3:

****Test Case ID:**** TC_TASK_PRIOR_03

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

****Test Case Description:**** This test case confirms that users can modify the priority of an existing task.

****Test Suite:**** Features/TaskManagement/TaskPrioritization

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

****Preconditions:****

- * A user is logged into the Task Management Tool.

- * A task with a predefined priority level exists. (e.g., Medium)

****Test Data:****

- * Task title: "Meeting Preparation"

- * Initial priority: Medium

****Test Steps:****

1. Navigate to the task list and select "Meeting Preparation."
2. Locate the "Priority" field.
3. Change the priority level to "High."
4. Save the changes to the task.

****Postconditions:****

* The task status should be updated.

****Expected Result:**** The priority of the "Meeting Preparation" task successfully changes to "High."

****Severity:**** High

****Priority:**** High

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

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

Test Case 4:

****Test Case ID:**** TC_TASK_PRIOR_04

****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 identifiable on the interface.

****Test Suite:**** Features/TaskManagement/TaskPrioritization

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

****Preconditions:****

- * A user is logged into the Task Management Tool.
- * Several tasks are present with different priority levels (High, Medium, Low).

****Test Data:****

- * A task with each priority level.

****Test Steps:****

1. Navigate to the task list.
2. Observe the visual appearance of tasks with high, medium, and low priority.
3. Note any distinct visual cues used to differentiate priority levels (e.g., color coding, icons, font weight).

****Postconditions:** N/A**

****Expected Result:**** Tasks are visually differentiated based on their assigned priority levels.

****Severity:**** Major

****Priority:**** High

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

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

Test Case 5:

****Test Case ID:** TC_TASK_PRIOR_05**

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

****Test Case Description:** This test case confirms that when no priority level is assigned to a task, it defaults to "Medium."**

****Test Suite:** Features/TaskManagement/TaskPrioritization**

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

****Preconditions:****

* A user is logged into the Task Management Tool.

****Test Data:****

* A new task is created without a specified priority level.

****Test Steps:****

1. Create a new task.
2. Do not assign a priority level to the task.
3. Save the task.
4. View the task details.

****Postconditions:** N/A**

****Expected Result:** The task will have a default priority level of "Medium."**

****Severity:**** Minor

****Priority:**** Medium

****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:

Task Management Tool - Subtasks Test Cases

Here are 5 test cases designed to verify the functionality of the "Subtasks" feature in the Task Management Tool:

****Test Case 1: Verify that subtasks can be created under main tasks****

* Test Case ID: TC_SUBTASK_001

* Test Case Title: Verify that subtasks can be created under a main task

* Test Case Description: This test case verifies the user's ability to create subtasks directly beneath a primary task, ensuring proper hierarchical structure and organization.

* Test Suite: Subtasks Feature

* Test Priority: High

* Preconditions:

 * A user is logged in to the application.

 * A main task has been created.

* Test Data:

 * Subtask name: "Write Introduction"

 * Subtask description: "Draft a compelling introduction for the report"

 * Due date: 2024-03-15

* Test Steps:

1. Navigate to the task details page of the pre-existing main task.
2. Click on the "Add Subtask" button.

3. Enter the subtask name, description, and due date as specified in the Test Data.
4. Click "Save" to create the subtask.
5. Verify that the subtask "Write Introduction" is listed under the main task in the task list.

* Postconditions:

- * The new subtask remains visible in the task list.
- * The subtask's details are accurately reflected.
- * Expected Result: The subtask should be successfully created and displayed under the main task.
- * Severity: Critical
- * Priority: High
- * Type of Testing: Functional Testing
- * Test Case Approach: Positive

****Test Case 2: Verify that subtasks can have their own due dates and assignees****

* Test Case ID: TC_SUBTASK_002

* Test Case Title: Verify that subtasks can have independent due dates and assignees

* Test Case Description: This test case validates the functionality of assigning different due dates and individuals to subtasks, ensuring independent task management.

* Test Suite: Subtasks Feature

* Test Priority: High

* Preconditions:

- * A user is logged in to the application.
- * A main task with an existing subtask is present.

* Test Data:

- * New Subtask Name: "Finalize Designs"
- * New Subtask Description: "Complete final versions of the wireframes"
- * New Subtask Due Date: 2024-03-20

* Assignees: John Doe

* Test Steps:

1. Open the detailed view of the main task.
2. Locate the existing subtask.
3. Click the "Edit" button for the existing subtask.
4. Change the subtask's due date, description, and assignee as per the Test Data.
5. Click "Save" to apply the changes.

* Postconditions:

* The subtask's details are updated with the new information.

* Expected Result: The subtask should be updated with the new due date and assignee successfully reflected.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

****Test Case 3: Verify that the system tracks the progress of each subtask****

* Test Case ID: TC_SUBTASK_003

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

* Test Case Description: This test case ensures that the system accurately reflects the completion status of each subtask, allowing for effective project monitoring.

* Test Suite: Subtasks Feature

* Test Priority: High

* Preconditions:

* A user is logged in to the application.

* A main task with multiple subtasks is present.

* Test Data:

- * Subtask: 1 completed out of 3

- * Test Steps:

1. Navigate to the task details page of the main task.
2. As previously completed, signify the completion status of one of the subtasks within the task context.
3. Observe the progress indicator for the main task based on the completed subtask.

- * Postconditions:

- * The progress of the main task reflects the completion status of at least one subtask.

- * Expected Result: The progress indicator for the main task should reflect the completion of at least one subtask (showing approximately 33% complete).

- * Severity: Critical

- * Priority: High

- * Type of Testing: Functional Testing

- * Test Case Approach: Positive

****Test Case 4: Verify that subtasks can be marked as complete independently of the main task****

- * Test Case ID: TC_SUBTASK_004

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

- * Test Case Description: This test case confirms that a subtask can be marked as complete even if the parent task is not yet finished, allowing for granular progress tracking.

- * Test Suite: Subtasks Feature

- * Test Priority: High

- * Preconditions:

- * A user is logged in to the application.
 - * A main task with at least one incomplete subtask exists.

- * Test Data:

- * Selected Subtask: "Write Code"

- * This subtask should be in an incomplete state

- * Test Steps:

1. Go to the task details page of the main task.
2. Locate and select the "Write Code" subtask.
3. Click the "Mark as Complete" button for the selected subtask.

- * Postconditions:

- * The selected subtask displays as complete.

- * Expected Result: The "Write Code" subtask should be marked as completed, even if the main task remains incomplete.

- * Severity: Major

- * Priority: High

- * Type of Testing: Functional Testing

- * Test Case Approach: Positive

****Test Case 5: Verify that dependencies between subtasks can be set****

- * Test Case ID: TC_SUBTASK_005

- * Test Case Title: Verify that dependencies between subtasks can be set

- * Test Case Description: This test case ensures that subtasks can be linked together based on logical dependencies, forcing completion of one task before another can begin.

- * Test Suite: Subtasks Feature

- * Test Priority: Medium

- * Preconditions:

- * A user is logged in to the application.
 - * A main task with multiple subtasks exists.

- * Test Data:

- * Subtask 1: "Create Presentation Design"

* Subtask 2: "Add Content to Presentation" (Dependent on Subtask 1)

* Test Steps:

1. Open the task details page for the main task.
2. Locate Subtask 1: "Create Presentation Design."
3. Click the "Add Dependency" button.
4. Select Subtask 2: "Add Content to Presentation" as the dependent task.
5. Save the dependency relationship.

* Postconditions:

* A dependency link is established between Subtask 1 and Subtask 2.

* Expected Result: Subtask 2 should be marked as dependent on the completion of Subtask 1. Attempting to mark Subtask 2 as complete before Subtask 1 should result in an error or warning message.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

These test cases are starting points and can be expanded upon to explore various scenarios, edge cases, and potential vulnerabilities within the "Subtasks" feature.

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:

Test Cases for Task Dependencies Feature

Here are 5 detailed test cases for the "Task Dependencies" feature in the Task Management Tool, following the provided template:

****Test Case ID:**** TC_TASK_001

****Test Case Title:**** Verify that users can set dependencies between tasks

****Test Case Description:**** This test case verifies the ability to establish dependencies between tasks,

ensuring that a new task can be marked as dependent on an existing task.

****Test Suite:**** Task Management Feature Testing

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

****Preconditions:****

- * A user is logged in to the Task Management Tool.
- * At least two tasks exist in the system.

****Test Data:**** Task names, descriptions, and dates

****Test Steps:****

1. Navigate to the task list.
2. Select an existing task (Task A).
3. Click on the "Dependencies" section within the task details.
4. Select the "Add Dependency" button.
5. Choose another existing task (Task B) as the dependent task.
6. Save the changes.

****Postconditions:****

- * Task B should now be listed as a dependency of Task A.

****Expected Result:**** The system successfully adds Task B as a dependency for Task A, reflecting the relationship in the task details.

****Severity:** Critical**

****Priority:** High**

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

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

****Test Case ID:** TC_TASK_002**

****Test Case Title:** Verify that the system displays dependencies in the task details**

****Test Case Description:** This test case verifies that the system accurately displays the dependencies of a task in its details view.**

****Test Suite:** Task Management Feature Testing**

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

****Preconditions:****

- * A user is logged in to the Task Management Tool.
- * At least two tasks exist with an established dependency relationship.

****Test Data:** Task names, descriptions, dates, and dependencies**

****Test Steps:****

1. Navigate to the task list.
2. Select a task with dependencies (Task A).
3. Open the task details view.
4. Observe the "Dependencies" section.

****Postconditions:****

- * No changes are required.

****Expected Result:**** The "Dependencies" section clearly lists all tasks that are dependent on Task A.

****Severity:**** Major

****Priority:**** Medium

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

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

****Test Case ID:**** TC_TASK_003

****Test Case Title:**** Verify that tasks cannot be marked as complete if dependent tasks are not completed

****Test Case Description:**** This test case verifies that the system prevents the completion of a task if any of its dependent tasks remain incomplete.

****Test Suite:**** Task Management Feature Testing

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

****Preconditions:****

- * A user is logged in to the Task Management Tool.

- * At least two tasks exist with a dependency relationship where Task A depends on Task B.

****Test Data:**** Task names, descriptions, dates, and dependencies

****Test Steps:****

1. Navigate to the task list.
2. Select Task A, which has a dependency on Task B.
3. Ensure Task B is marked as "Incomplete."
4. Attempt to mark Task A as "Complete."

****Postconditions:****

- * No changes are required.

****Expected Result:**** The system does not allow Task A to be marked as "Complete" because Task B, its dependent task, is still "Incomplete."

****Severity:**** Critical

****Priority:**** High

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

****Test Case Approach:**** Negative

****Test Case ID:**** TC_TASK_004

****Test Case Title:**** Verify that users can update or remove dependencies as needed

****Test Case Description:**** This test case verifies the ability to modify or delete existing dependencies between tasks.

****Test Suite:**** Task Management Feature Testing

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

****Preconditions:****

- * A user is logged in to the Task Management Tool.
- * Two tasks exist, with a pre-established dependency relationship.

****Test Data:**** Task names, descriptions, dates, and dependencies

****Test Steps:****

1. Navigate to the task list.
2. Select the task with existing dependencies (Task A).
3. Open the task details view and go to the "Dependencies" section.
4. Choose "Edit" for an existing dependency.
5. Update the dependent task to a different task.
6. Save the changes.

****Postconditions:****

- * The dependency should be updated to the new task.

****Expected Result:**** The system successfully modifies the dependency relationship for Task A.

****Severity:**** Major

****Priority:**** Medium

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

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

****Test Case ID:** TC_TASK_005**

****Test Case Title:** Verify that the system provides alerts or warnings if a dependent task is not completed**

****Test Case Description:** This test case verifies the system's ability to alert or warn the user when a dependent task is not completed before the deadline of the task relying on it.**

****Test Suite:** Task Management Feature Testing**

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

****Preconditions:****

- * A user is logged in to the Task Management Tool.
- * A task (Task A) with a dependent task (Task B) exists.
- * Task B has a deadline before Task A's deadline.

****Test Data:** Task names, descriptions, dates, dependencies, and deadlines**

****Test Steps:****

1. Navigate to the task list.
2. Select Task B, which is dependent on Task A.
3. Mark Task B as "Incomplete."
4. Attempt to mark Task A as "Complete."

****Postconditions:****

* The system should show an alert or warning message.

****Expected Result:**** The system prevents Task A from being marked as "Complete" and displays an alert or warning message indicating that Task B, a dependent task, is still incomplete.

****Severity:**** Critical

****Priority:**** High

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

****Test Case Approach:**** Negative

Data - 27

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Comments and Collaboration

Feature description -

Users can collaborate on tasks by leaving comments, attaching files, and engaging in discussions directly

within the task interface. This feature facilitates communication and teamwork, allowing team members to provide updates, share feedback, and ask questions related to the task.

Userstory title -

As a user, I want to comment on tasks so that I can collaborate with my team.

Acceptance criteria -

['Users can add comments to tasks.', 'Comments can include text, attachments, and links.', 'Users can reply to and edit their comments.', 'The system notifies users of new comments.', 'Comments are displayed in chronological order.']

Generated Test Cases:

Test Case Creation for Task Management Tool: Task Comments & Collaboration

Here are 5 detailed test cases for the user story "As a user, I want to comment on tasks so that I can collaborate with my team.":

Test Case 1

****Test Case ID:** TC_001**

****Test Case Title:** Verify that users can add text comments to tasks.**

****Test Case Description:** This test case verifies the functionality of adding text-based comments to a task. It will ensure users can input free-form text, submit the comment, and view it correctly within the task interface.**

****Test Suite:** Collaboration Functionality**

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

****Preconditions:****

- * User is logged in to the Task Management Tool.
- * A task exists that is accessible to the logged-in user.

****Test Data:****

- * Test comment text: "This is a test comment."

****Test Steps:****

1. Navigate to the task in the Test Management Tool.
2. Click on the "Add Comment" button or field.
3. Enter the test comment text: "This is a test comment."
4. Click on the "Submit" button or press Enter.
5. Verify that the comment "This is a test comment." is displayed under the task information.

****Postconditions:****

- * The comment remains visible on the task page.

****Expected Result:**** The user's comment is successfully added to the task and displayed chronologically.

****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 users can add attachments to task comments.**

****Test Case Description:** This test case verifies the functionality of adding files as attachments to task comments. Users should be able to select a file from their device and attach it to a comment.**

****Test Suite:** Collaboration Functionality**

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

****Preconditions:****

- * User is logged in to the Task Management Tool.
- * A task exists that is accessible to the logged-in user.

****Test Data:****

- * A test file (e.g., a text document, image, PDF)

****Test Steps:****

1. Navigate to the task in the Test Management Tool.
2. Click on the "Add Comment" button or field.
3. Click on the "Attach File" button or area.
4. Select the test file from your device.
5. Submit the comment with the attached file.

****Postconditions:****

* The comment, along with the attached file, is visible on the task page.

****Expected Result:**** The user's comment with the attached file is successfully added to the task and displayed chronologically.

****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 edit their own comments.

****Test Case Description:**** This test case verifies the functionality for users to edit their previously added comments. Only the user who originally posted the comment should have the ability to edit it.

****Test Suite:**** Collaboration Functionality

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

****Preconditions:****

- * User is logged in to the Task Management Tool.
- * The user has previously added a comment to a task.

****Test Data:****

- * Original Comment text: "Initial comment!"
- * Edited Comment text: "Revised comment."

****Test Steps:****

1. Navigate to the task with the previous comment.
2. Locate the comment added by the current user.
3. Click on the edit button associated with the comment.
4. Modify the comment text to "Revised comment."
5. Click the "Save Changes" button.

****Postconditions:****

- * The comment is successfully updated to the revised text.

****Expected Result:**** The user's comment is successfully edited and the updated text is displayed.

****Severity:** Major**

****Priority:** Medium**

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

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

Test Case 4

****Test Case ID:** TC_004**

****Test Case Title:** Verify that users receive notifications of new task comments.**

****Test Case Description:** This test case verifies that users are notified about new comments added to tasks they are involved with. This may involve email, in-app notifications, or other designated communication channels.**

****Test Suite:** Collaboration Functionality**

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

****Preconditions:****

* User is logged in to the Task Management Tool.

* A task exists that the user is assigned to or otherwise interested in.

****Test Data:****

* Test comment text: "New Comment!"

****Test Steps:****

1. Log out of the Task Management Tool.
2. Another user adds the test comment to the specific task.
3. Log back into the Task Management Tool.
4. Check for notifications regarding the new comment.

****Postconditions:****

* The user has received a notification about the new comment.

****Expected Result:**** The user receives a timely notification about the new comment added to the task.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Acceptance Testing

****Test Case Approach:**** Positive, Integration Testing (Assuming notification feature integrates with user login and task assignment)

Test Case 5

****Test Case ID:**** TC_005

****Test Case Title:**** Verify that comments are displayed in chronological order.

****Test Case Description:**** This test case verifies that comments are displayed on the task page in the order they were added, from oldest to newest.

****Test Suite:**** Collaboration Functionality

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

****Preconditions:****

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

- * A task exists that allows for comments.

****Test Data:****

- * Multiple comment texts (e.g., "Comment 1", "Comment 2", "Comment 3")

****Test Steps:****

1. Add "Comment 1" to the task.
2. Add "Comment 2" to the task.
3. Add "Comment 3" to the task.

****Postconditions:****

- * Comments are displayed in the correct chronological order.

****Expected Result:**** The comments are displayed on the task page as follows:

Comment 1

Comment 2

Comment 3

****Severity:**** Low

****Priority:**** Medium

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

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

Data - 28

Project name -

Task Management Tool

Project description -

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

Feature name -

File Attachments

Feature description -

Users can attach relevant files, documents, and resources to tasks, ensuring that all necessary information is accessible in one place. File attachments can include documents, spreadsheets, images, links, and other resources needed for task execution.

Userstory title -

As a user, I want to attach files to tasks so that all relevant information is easily accessible.

Acceptance criteria -

['Users can attach multiple files to a task.', 'Attachments can include various file types (documents, images, etc.).', 'Users can view and download attachments.', 'The system limits the size of attachments to prevent data overload.', 'Users can remove attachments if needed.']

Generated Test Cases:

Test Cases for "File Attachments" Feature

****User Story:**** As a user, I want to attach files to tasks so that all relevant information is easily accessible.

****Acceptance Criteria:****

- * Users can attach multiple files to a task.
- * Attachments can include various file types (documents, images, etc.).
- * Users can view and download attachments.
- * The system limits the size of attachments to prevent data overload.
- * Users can remove attachments if needed.

****Test Case 1****

****Test Case ID:** TC_001**

****Test Case Title:** Verify that users can attach multiple files to a task.**

****Test Case Description:** This test case aims to verify that users can successfully attach more than one file to a single task within the system.**

****Test Suite:** Attachment Functionality**

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

****Preconditions:****

* A user is logged into the system.

* A new task is created.

****Test Data:****

* 3 different file types (e.g., .docx, .jpg, .pdf) with varying sizes.

****Test Steps:****

1. Navigate to the created task.
2. Click on the "Attach File" button.
3. Select the three files prepared in Test Data.
4. Click "Open" to attach all selected files.
5. Verify that all three files are listed under the "Attachments" section of the task.

****Postconditions:****

* The task remains accessible.

* The attached files remain visible.

****Expected Result:** All three files should be successfully attached to the task.**

****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 the system allows attachments of various file types.

****Test Case Description:**** This test case verifies that the system accepts different file types as attachments to tasks, covering various file formats commonly used.

****Test Suite:**** Attachment Functionality

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

****Preconditions:****

* A user is logged into the system.

* A new task is created.

****Test Data:****

* .docx, .jpg, .pdf, .xlsx, .png, .txt

****Test Steps:****

1. Navigate to the created task.
2. Click on the "Attach File" button.
3. Select each file type in the Test Data sequentially.
4. Click "Open" to attach each selected file.
5. Verify that all six files are listed under the "Attachments" section of the task.

****Postconditions:****

* The task remains accessible.

* All attached files remain visible.

****Expected Result:**** All six file types should be successfully attached to the task.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Cases 3-5****

Continue following the Template above, focusing on:

- * ****Test Case 3:**** Verify that the system limits the size of attachments. (File size beyond the limit should result in an error message.)
- * ****Test Case 4:**** Verify that users can download attachments successfully.
- * ****Test Case 5:**** Verify that users can remove attached files from a task.

Remember to include negative scenarios and edge cases in your test cases to ensure comprehensive coverage of the "File Attachments" feature.

Data - 29

Project name -

Task Management Tool

Project description -

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

Feature name -

Recurring Tasks

Feature description -

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

Userstory title -

As a user, I want to set recurring tasks so that regular activities are automated.

Acceptance criteria -

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

Generated Test Cases:

Task Management Tool - Recurring Tasks Test Cases

Test Case 1: Verify that tasks can be set to recur daily.

****Test Case Description:**** This test case verifies that users can set a task to recur daily and that the system generates instances of the task accordingly.

****Test Suite:**** Recurring Tasks

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

****Preconditions:****

* A user account is created and logged in.

****Test Data:****

* Task title: "Daily Report"

* Task description: "Generate a daily report on project progress."

* Due date: Tomorrow

****Test Steps:****

1. Navigate to the "Create New Task" page.
2. Enter the task title, description, and due date.
3. Select "Daily" from the recurrence frequency dropdown menu.
4. Click the "Create Task" button.
5. Verify that a new recurring task is created with daily recurrence.
6. Check the task calendar view for upcoming daily instances of the task.

****Postconditions:****

* The recurring task is active.

****Expected Result:**** The system successfully creates and schedules daily recurring instances of the task, with notifications sent for each instance.

****Severity:**** Blocker

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

Test Case 2: Verify that tasks can be set to recur weekly.

****Test Case Description:**** This test case verifies the functionality of setting tasks to recur weekly and confirms the system generates recurring instances accordingly.

****Test Suite:**** Recurring Tasks

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

****Preconditions:****

* A user account is created and logged in.

****Test Data:****

* Task title: "Weekly Meeting"

* Task description: "Schedule a weekly team meeting."

* Due date: Next Tuesday

****Test Steps:****

1. Navigate to the "Create New Task" page.
2. Enter the task title, description, and due date.
3. Select "Weekly" from the recurrence frequency dropdown menu.
4. Choose a specific day of the week for recurrence.
5. Click the "Create Task" button.
6. Verify that a new recurring task is created with weekly recurrence on the selected day.
7. Check the task calendar view for upcoming weekly instances of the task.

****Postconditions:****

* The recurring task is active.

****Expected Result:****

The system successfully creates and schedules weekly recurring instances of the task on the specified day, with notifications sent for each instance.

****Severity:**** Blocker

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

Test Case 3: Verify that tasks can be set to recur monthly.

****Test Case Description:**** This test case verifies the functionality of setting tasks to recur monthly and confirms the system generates recurring instances accordingly.

****Test Suite:**** Recurring Tasks

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

****Preconditions:****

* A user account is created and logged in.

****Test Data:****

* Task title: "Monthly Report"

* Task description: "Generate a monthly report on project progress."

* Due date: First Friday of next month

****Test Steps:****

1. Navigate to the "Create New Task" page.
2. Enter the task title, description, and due date.
3. Select "Monthly" from the recurrence frequency dropdown menu.
4. Choose the specific day of the month for recurrence (e.g., the first Friday).
5. Click the "Create Task" button.
6. Verify that a new recurring task is created with monthly recurrence on the selected day.
7. Check the task calendar view for upcoming monthly instances of the task.

****Postconditions:****

* The recurring task is active.

****Expected Result:****

The system successfully creates and schedules monthly recurring instances of the task on the chosen day of the month, with notifications sent for each instance.

****Severity:** Blocker**

****Priority:** High**

****Type of Testing:** Functional Testing, System Testing**

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

Test Case 4: Verify that users can modify the recurrence pattern of an existing task.

****Test Case Description:** This test case verifies the ability to modify the recurrence pattern of an existing recurring task.**

****Test Suite:** Recurring Tasks**

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

****Preconditions:****

* A user account is created and logged in.

* A recurring task with daily recurrence is already created.

****Test Data:****

* Existing task: "Daily Report" (recurring daily)

* Modified recurrence: Weekly (every Monday)

****Test Steps:****

1. Navigate to the task list and open the "Daily Report" task.
2. Click on the "Edit Recurrence" button.
3. Select "Weekly" from the recurrence frequency dropdown menu.
4. Choose "Monday" as the day of the week for recurrence.
5. Click "Save Changes".
6. Verify that the task's recurrence pattern is updated to weekly on Mondays.
7. Check the task calendar view for upcoming weekly instances of the task.

****Postconditions:****

* The task's recurrence settings are updated.

****Expected Result:**** The system successfully modifies the recurrence pattern of the task, changing it from daily to weekly on Mondays. Notifications should reflect the updated recurrence.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, System Testing

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

Test Case 5: Verify that users can cancel the recurrence of a task.

****Test Case Description:**** This test case verifies the ability to cancel the recurrence of a task, making it a regular, non-recurring task.

****Test Suite:**** Recurring Tasks

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

****Preconditions:****

- * A user account is created and logged in.
- * A recurring task with weekly recurrence is already created.

****Test Data:****

- * Recurring task: "Weekly Meeting" (recurring weekly)

****Test Steps:****

1. Navigate to the task list and open the "Weekly Meeting" task.
2. Click on the "Edit Recurrence" button.
3. Uncheck the "Recur" checkbox.
4. Click "Save Changes".
5. Verify that the task's recurrence pattern is canceled.
6. Check the task calendar view for no further instances of the recurring task.

****Postconditions:****

- * The task is no longer recurring.

****Expected Result:**** The system successfully cancels the recurrence of the task. Future recurring instances disappear from the task calendar and notifications.

,

Data - 30

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tags and Labels

Feature description -

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

Userstory title -

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

Acceptance criteria -

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

Generated Test Cases:

Test Case Generation for Task Management Tool - Feature: Task Tags and Labels

****User Story:**** As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

****Test Cases:****

****Test Case ID:**** TCMT-TAG-001

****Test Case Title:**** Verify that users can add tags to tasks.

****Test Case Description:**** This test case verifies the ability of users to add tags to tasks. It will test different tag formats and character limits.

****Test Suite:**** User Story: Task Tags and Labels

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

****Preconditions:****

* A user is logged into the Task Management Tool.

* A new task is created.

****Test Data:**** Various tag names (short, long, with special characters, numbers)

****Test Steps:****

1. Create a new task.
2. Go to the task form and locate the "Tags" field.
3. Enter different tag names into the "Tags" field, varying in length and format.
4. Observe that the tags are added and displayed correctly associated with the task.
5. Save the task.

****Postconditions:****

* The task with the added tags is saved successfully.

****Expected Result:**** Users can successfully add tags to tasks with no error messages. Added tags should be displayed correctly alongside the task.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

****Test Case ID:**** TCMT-TAG-002

****Test Case Title:**** Verify that users can edit existing tags on tasks.

****Test Case Description:**** This test case verifies the ability of users to edit existing tags assigned to a task.

****Test Suite:**** User Story: Task Tags and Labels

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

****Preconditions:****

* A user is logged into the Task Management Tool.

* A task with at least one tag already exists.

****Test Data:**** Existing tag names, modified tag names

****Test Steps:****

1. Locate the task with existing tags.
2. Access the task form and locate the "Tags" field.
3. Select an existing tag from the list.
4. Edit the selected tag and observe that the name is updated in the list.
5. Save the task.

****Postconditions:****

* The task with updated tags is saved successfully.

****Expected Result:**** Users can successfully edit existing tags on tasks with no error messages.

****Severity:**** Major

****Priority:** High**

****Type of Testing:** Functional Testing, User Acceptance Testing**

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

****Test Case ID:** TCMT-TAG-003**

****Test Case Title:** Verify that users can remove tags from tasks.**

****Test Case Description:** This test case verifies the ability of users to remove tags assigned to a task.**

****Test Suite:** User Story: Task Tags and Labels**

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

****Preconditions:****

* A user is logged into the Task Management Tool.

* A task with at least one tag exists.

****Test Data:** Existing tag names**

****Test Steps:****

1. Locate the task with existing tags.
2. Access the task form and locate the "Tags" field.

3. Select the tag to be removed from the list.
4. Click the "Remove" button for the selected tag.
5. Verify that the selected tag is successfully removed from the list associated with the task.
6. Save the task.

****Postconditions:****

* The task with removed tag is saved successfully.

****Expected Result:**** Users can successfully remove tags from tasks with no error messages.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

******(Add remaining test cases for filtering, sorting, and edge cases as required)******

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:

Task Management Tool: Test Cases for Task Status Tracking

****User Story:**** As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

****Test Case 1:****

* **Test Case ID:** TC_STT_001

* **Test Case Title:** Verify that users can update the status of tasks.

* **Test Case Description:** This test case aims to verify that users can successfully change the status of tasks to "To Do", "In Progress", and "Completed" within the task management tool.

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

* **Test Priority:** High

* **Preconditions:**

- * A user is logged into the task management tool.

- * A new task exists with an initial status of "To Do".

* **Test Data:**

- * Task ID: 123

- * Updated Status Choices: "In Progress", "Completed"

* **Test Steps:**

1. Navigate to the task details page for task ID 123.
2. Select the "In Progress" status from the task status dropdown menu.
3. Click the "Update Status" button.
4. Verify that the task status is updated to "In Progress" on the task details page.
5. Repeat steps 2-4, selecting "Completed" as the status.
6. Verify that the task status is updated to "Completed" on the task details page.

* **Postconditions:**

- * The task remains accessible with its updated status.

* **Expected Result:** The user is able to successfully update the task status to both "In Progress" and "Completed" without encountering errors.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 2:****

*** **Test Case ID:** TC_STT_002**

*** **Test Case Title:** Verify that the system prominently displays task status in the task list.**

*** **Test Case Description:** This test case aims to verify that the system displays the status of each task clearly and visibly within the task list view.**

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

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

*** **Preconditions:****

- * A user is logged into the task management tool.

- * There are multiple tasks with different statuses ("To Do", "In Progress", "Completed") in the task list.

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

- * Multiple tasks with different statuses.

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

1. Navigate to the task list view.

2. Observe the status of each task displayed within the list.

*** **Postconditions:****

- * No changes to the task list or its structure are necessary.

*** **Expected Result:****

- * Each task's status is clearly displayed next to its title within the task list, using distinct visual cues (e.g., color-coding or icons) to differentiate each status.

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

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

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

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

****Test Case 3:****

*** **Test Case ID:** TC_STT_003**

*** **Test Case Title:** Verify that the system displays task progress percentage.**

*** **Test Case Description:** This test case verifies that the system accurately calculates and displays the progress percentage for tasks with defined progress increments.**

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

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

*** **Preconditions:****

*** A user is logged into the task management tool.**

*** A task exists with a set percentage progress (e.g., 50%).**

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

*** Task ID: 456**

*** Expected Progress Percentage: 50%**

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

1. Navigate to the task details page for task ID 456.

2. Observe the progress percentage displayed for the task (e.g., 50%).

*** **Postconditions:****

*** No changes necessary.**

*** **Expected Result:** The displayed progress percentage for the task accurately reflects the set percentage progress.**

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

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

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

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

****Test Case 4:****

* **Test Case ID:** TC_STT_004

* **Test Case Title:** Verify that the system provides notifications for status changes.

* **Test Case Description:** This test case verifies that users receive timely notifications when the status of tasks they are assigned to or are following changes.

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

* **Test Priority:** High

* **Preconditions:**

- * A user is logged into the task management tool.

- * A task exists with a user assigned to it.

* **Test Data:**

- * Task ID: 789

- * Assigned User email: testuser@example.com

- * Expected Notification Content: New status update on task 789.

* **Test Steps:**

1. Log in as the assigned user.

2. In the task details for task 789, manually change the status to "In Progress."

3. Check the user's email for a notification within a reasonable timeframe (e.g., within 5 minutes).

* **Postconditions:**

- * No changes to the task or notification settings are necessary.

* **Expected Result:** The assigned user receives a notification email containing the details of the status change for task 789.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, Security Testing

* **Test Case Approach:** Positive

****Test Case 5:****

*** **Test Case ID:** TC_STT_005**

*** **Test Case Title:** Verify that task status can be filtered and sorted.**

*** **Test Case Description:** This test case ensures that users can filter and sort tasks based on their status within the task list view.**

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

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

*** **Preconditions:****

*** A user is logged into the task management tool.**

*** A mix of tasks with different statuses ("To Do", "In Progress", "Completed") exists.**

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

*** Multiple tasks with different statuses.**

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

1. Navigate to the task list view.
2. Filter the task list to display only "Completed" tasks.
3. Verify that only completed tasks are displayed.
4. Clear the filter.
5. Sort the task list by "Status" in ascending order (e.g., "To Do" to "Completed").
6. Verify that tasks are listed from "To Do" to "Completed."
7. Sort the task list by "Status" in descending order.
8. Verify that tasks are listed from "Completed" to "To Do."

*** **Postconditions:****

*** The filter and sorting functions are applied correctly and efficiently.**

*** **Expected Result:** Users can successfully filter and sort tasks based on their status within the task list view.**

* **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:

Test Cases for Task Filtering and Sorting Feature

****Test Case ID:**** TC_TASK_FILTER_01

****Test Case Title:**** Verify that tasks can be filtered by due date.

****Test Case Description:**** This test case verifies the functionality of filtering tasks based on due date.

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

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

****Preconditions:****

- * A user account is logged in.
- * At least 3 tasks with different due dates exist in the system.

****Test Data:****

- * Different due dates for existing tasks (e.g., today, tomorrow, next week, next month)

****Test Steps:****

1. Navigate to the task list view.

2. Select the "Due Date" filter option.

3. Choose a specific date range or select an individual due date from the dropdown.

4. Verify that only tasks with the selected due date(s) are displayed in the task list.

****Postconditions:****

* The filter should be cleared automatically after the test is completed.

****Expected Result:**** Tasks are successfully filtered based on the selected due date(s).

****Severity:**** Critical

****Priority:**** High

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

****Test Case ID:**** TC_TASK_FILTER_02

****Test Case Title:**** Verify that tasks can be filtered by priority.

****Test Case Description:**** This test case verifies the functionality of filtering tasks based on priority.

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

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

****Preconditions:****

- * A user account is logged in.
- * At least 3 tasks with different priorities exist in the system.

****Test Data:****

- * Tasks assigned different priority levels (e.g., High, Medium, Low).

****Test Steps:****

1. Navigate to the task list view.
2. Select the "Priority" filter option.
3. Choose a specific priority level (e.g., High) from the dropdown.
4. Verify that only tasks with the selected priority are displayed in the task list.

****Postconditions:****

- * The filter should be cleared automatically after the test is completed.

****Expected Result:**** Tasks are successfully filtered based on the selected priority level.

****Severity:**** Critical

****Priority:**** High

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

****Test Case ID:** TC_TASK_FILTER_03**

****Test Case Title:** Verify that tasks can be filtered by assignee.**

****Test Case Description:** This test case verifies the functionality of filtering tasks based on assigned users.**

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

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

****Preconditions:****

- * A user account is logged in.
- * At least 3 tasks are assigned to different users in the system.

****Test Data:****

- * Multiple user accounts with distinct task assignments.

****Test Steps:****

1. Navigate to the task list view.
2. Select the "Assignee" filter option.
3. Choose a specific user from the dropdown.
4. Verify that only tasks assigned to the selected user are displayed in the task list.

****Postconditions:****

- * The filter should be cleared automatically after the test is completed.

****Expected Result:**** Tasks are successfully filtered based on the selected assignee.

****Severity:**** Critical

****Priority:**** High

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

****Test Case ID:**** TC_TASK_FILTER_04

****Test Case Title:**** Verify that tasks can be filtered by project.

****Test Case Description:**** This test case verifies the functionality of filtering tasks based on project.

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

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

****Preconditions:****

- * A user account is logged in.

- * At least 3 tasks are assigned to different projects in the system.

****Test Data:****

- * Projects with distinct names and assigned tasks.

****Test Steps:****

1. Navigate to the task list view.
2. Select the "Project" filter option.
3. Choose a specific project from the dropdown.
4. Verify that only tasks belonging to the selected project are displayed in the task list.

****Postconditions:****

- * The filter should be cleared automatically after the test is completed.

****Expected Result:**** Tasks are successfully filtered based on the selected project.

****Severity:**** Critical

****Priority:**** High

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

****Test Case ID:**** TC_TASK_SORT_01

****Test Case Title:**** Verify that tasks can be sorted by due date.

****Test Case Description:**** This test case verifies the functionality of sorting tasks by due date.

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

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

****Preconditions:****

* A user account is logged in.

* At least 3 tasks with different due dates exist in the system.

****Test Data:****

* Tasks with various due dates.

****Test Steps:****

1. Navigate to the task list view.
2. Click on the "Due Date" column header.
3. Verify that the tasks are sorted by due date in ascending order (oldest first)
4. Click on the "Due Date" column header again.
5. Verify that the tasks are sorted by due date in descending order (newest first).

****Postconditions:****

* The sorting order should be maintained even when navigating to other views and returning to the task list.

****Expected Result:**** The task list successfully sorts by due date in both ascending and descending order.

****Severity:**** Critical

****Priority:** High**

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

Data - 33

Project name -

Task Management Tool

Project description -

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

Feature name -

Time Tracking

Feature description -

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

Userstory title -

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Acceptance criteria -

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

Generated Test Cases:

##Test Cases for Time Tracking Feature

Project Name: Task Management Tool

Feature Name: Time Tracking

User Story: As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Test Case 1

Test Case ID: TC_TT_001

Test Case Title: Verify that users can start and stop a timer for each task.

Test Case Description: This test case verifies the functionality of starting and stopping a timer for a specific task. It will ensure that the timer accurately records the time spent on the task.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- * A new task is created with a title and assigned to a user.
- * The user is logged into the application.

Test Data: Task ID, Timer duration

****Test Steps:****

1. Navigate to the task details page.
2. Click the "Start Timer" button.
3. Perform some work on the task for a defined duration (e.g., 5 minutes).
4. Click the "Stop Timer" button.
5. Verify that the total time recorded for the task accurately reflects the time spent.

****Postconditions:****

- * The timer is stopped.

****Expected Result:**** The system should accurately capture and display the time spent on the task from the start to the stop time.

****Severity:**** Critical

****Priority:**** High

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

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

Test Case 2

****Test Case ID:**** TC_TT_002

****Test Case Title:**** Verify that the system records and displays total time spent on tasks.

****Test Case Description:**** This test case verifies the display of the total time spent on a task, including multiple start and stop instances.

****Test Suite:**** Time Tracking

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

****Preconditions:****

* A new task is created with a title and assigned to a user.

* The user is logged into the application.

****Test Data:**** Task ID, Multiple timer durations

****Test Steps:****

1. Navigate to the task details page.
2. Start the timer for a specific duration (e.g., 1 minute).
3. Stop the timer, leaving the task open.
4. Start the timer again after a brief period (e.g., 30 seconds).
5. Stop the timer again.
6. Repeat steps 4 and 5 multiple times.
7. Verify that the system correctly sums the duration of all timer sessions and displays the total time spent on the task.

****Postconditions:****

* The task is open and the timer is stopped.

****Expected Result:**** The displayed total time for the task should accurately reflect the cumulative duration of all tracked timer sessions.

****Severity:**** Critical

****Priority:**** High

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

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

Test Case 3

****Test Case ID:**** TC_TT_003

****Test Case Title:**** Verify that users can manually enter or adjust time entries.

****Test Case Description:**** This test case verifies the ability to manually input or modify time entries for tasks, allowing for flexible time tracking.

****Test Suite:**** Time Tracking

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

****Preconditions:****

* A new task is created with a title and assigned to a user.

* The user is logged into the application.

****Test Data:**** Task ID, Manual time entries

****Test Steps:****

1. Navigate to the task details page.
2. Manually enter a specific time duration for the task.
3. Verify that the entered time is accurately saved and displayed.
4. Adjust the manually entered time duration.
5. Verify that the updated time reflects the changes.

****Postconditions:****

* Time entries for the task are updated.

****Expected Result:**** The system should allow users to manually input, view, and modify time entries for tasks without issues.

****Severity:**** Major

****Priority:**** Medium

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

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

Test Case 4

****Test Case ID:** TC_TT_004**

****Test Case Title:** Verify that time tracking data is stored and can be exported.**

****Test Case Description:** This test case verifies the persistence and export capability of time tracking data, ensuring its availability for future reference and analysis.**

****Test Suite:** Time Tracking**

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

****Preconditions:****

* A user has been tracking time on tasks.

* The user is logged into the application.

****Test Data:** Task ID, Time tracking data**

****Test Steps:****

1. Manually input or track time on tasks for a defined period.
2. Navigate to the export options for time tracking data.
3. Select a desired export format (e.g., CSV, Excel).
4. Initiate the export process.
5. Verify that the exported file contains accurate time tracking data for the selected tasks and time period.

****Postconditions:****

* The exported file is successfully downloaded.

****Expected Result:**** The system should be able to store time tracking data persistently and export it in a usable format.

****Severity:**** Critical

****Priority:**** High

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

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

Test Case 5

****Test Case ID:**** TC_TT_005

****Test Case Title:**** Verify that the system provides reports on time spent per task or project.

****Test Case Description:**** This test case verifies the functionality of generating reports that display time spent on tasks or projects, enabling users to analyze their time usage.

****Test Suite:**** Time Tracking

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

****Preconditions:****

* Users have tracked time on multiple tasks within different projects.

* The user is logged into the application.

****Test Data:**** Time tracking data for multiple tasks and projects

****Test Steps:****

1. Navigate to the reporting section for time tracking.
2. Select a desired reporting option (e.g., time spent per task, time spent per project).
3. Specify the time period for the report (e.g., last week, last month).
4. Generate the report.
5. Verify that the report accurately displays the time spent on the selected tasks or projects within the specified timeframe.

****Postconditions:****

- * The report is generated and displayed.

****Expected Result:**** The system should generate comprehensive reports on time spent per task or project, allowing for insightful analysis.

****Severity:**** Major

****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:

Test Cases for Task Management Tool - Task Templates Feature

****User Story:**** As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

****Test Case 1:****

*** **Test Case ID:** TC_TM_001**

*** **Test Case Title:** Verify that users can create and save task templates.**

*** **Test Case Description:** This test case aims to ensure that users have the functionality to define and save new task templates with predefined structures, descriptions, assignees, and due dates.**

*** **Test Suite:** Task Management - Templates**

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

*** **Preconditions:****

- * User is logged into the Task Management Tool
- * The user has sufficient permissions to create templates

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

- * Template name: "Project Kickoff"
- * Template description: "Predefined tasks for project initiation"
- * Template tasks: "Define Scope", "Identify Resources", "Set Budget", "Assign Roles" (with specific deadlines and assignees)

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

1. Navigate to the Task Templates section.
2. Click the "Create New Template" button.
3. Enter the template name, description, and define the tasks with details (deadline, assignee, description).
4. Save the template.
5. Verify that the "Project Kickoff" template is listed in the available templates.

*** **Postconditions:****

- * The "Project Kickoff" template is saved and accessible.

*** **Expected Result:****

The test should successfully create and save the "Project Kickoff" template in the system, making it available for usage.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TC_TM_002

* **Test Case Title:** Verify that users can apply templates to new tasks or projects.

* **Test Case Description:** This test case checks the functionality of applying an existing template to create new tasks or projects.

* **Test Suite:** Task Management - Templates

* **Test Priority:** High

* **Preconditions:**

- * User is logged into the Task Management Tool

- * The previously created "Project Kickoff" template is available.

* **Test Data:**

- * New Project Name: "Software Development Project"

* **Test Steps:**

1. Navigate to the "Projects" section.

2. Click the "Create New Project" button.

3. Select the "Project Kickoff" template during project creation.

4. Verify that the new project "Software Development Project" automatically includes the predefined tasks from the template ("Define Scope", "Identify Resources", "Set Budget", "Assign Roles") with their respective deadlines and assignees.

* **Postconditions:**

- * A new project with the "Project Kickoff" template is created.

* **Expected Result:**

The new project inherits the task structure, descriptions, due dates, and assignees defined in the "Project Kickoff" template.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_TM_003

* **Test Case Title:** Verify that templates can be edited or deleted as needed.

* **Test Case Description:** This test case verifies the user's ability to modify and remove existing task templates.

* **Test Suite:** Task Management - Templates

* **Test Priority:** Medium

* **Preconditions:**

- * User is logged into the Task Management Tool

- * The user has sufficient permissions to edit/delete templates

* **Test Data:**

- * The existing "Project Kickoff" template.

* **Test Steps:**

1. Navigate to the Task Templates section.
2. Select the "Project Kickoff" template.
3. Click the "Edit" button.
4. Modify the description of "Project Kickoff" template.
5. Save the changes.
6. Click the "Delete" button.
7. Confirm the deletion of "Project Kickoff" template.

* **Postconditions:**

- * The "Project Kickoff" template is removed from the list.

* **Expected Result:**

- * The "Project Kickoff" template description is updated.

- * The "Project Kickoff" template is successfully deleted from the system.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

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

****Test Case 4:****

* **Test Case ID:** TC_TM_004

* **Test Case Title:** Verify that the system displays a list of available templates for selection.

* **Test Case Description:** This test case ensures that the user interface properly displays a list of available task templates for selection when creating new tasks.

* **Test Suite:** Task Management - Templates

* **Test Priority:** Medium

* **Preconditions:**

- * User is logged into the Task Management Tool

* **Test Data:** (No additional test data required)

* **Test Steps:**

1. Navigate to the "Tasks" section.
2. Click the "Create New Task" button.
3. Verify that a "Choose Template" dropdown menu is displayed.
4. Ensure that all available task templates are listed in the dropdown menu.

* **Postconditions:**

- * The "Choose Template" dropdown is closed.

* **Expected Result:**

The drop-down menu displays a relevant list of available task templates for the user to select from.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_TM_005

* **Test Case Title:** Verify that deleting a template does not affect existing tasks that were created using it.

* **Test Case Description:** This test case explores the scenario where a user wants to delete a template while ensuring that previously created tasks that used that template remain unaffected.

* **Test Suite:** Task Management - Templates

* **Test Priority:** Medium

* **Preconditions:**

* A project was created using a template named "Team Onboarding".

* The user has sufficient permissions to delete templates.

* **Test Data:**

* "Team Onboarding" template

* **Test Steps:**

1. Navigate to the Task Templates section.
2. Select the "Team Onboarding" template.
3. Click the "Delete" button.
4. Confirm the deletion of the template.
5. Navigate to the project that used the "Team Onboarding" template
6. Verify that all tasks within the project are still present and accessible.

* **Postconditions:**

* The "Team Onboarding" template is removed from the system.

* **Expected Result:**

The project and its tasks remain unaffected after deleting the template. Deleting a template should not cause data loss or issues for existing tasks.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

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

Data - 35

Project name -

Task Management Tool

Project description -

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

Feature name -

Calendar Integration

Feature description -

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments

alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

Userstory title -

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

Acceptance criteria -

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

Generated Test Cases:

Task Management Tool - Calendar Integration Test Cases

Here are 5 detailed test cases for the "Calendar Integration" feature:

****Test Case ID:**** TCC-CAL-001

****Test Case Title:**** Verify that tasks are synced with Google Calendar

****Test Case Description:**** This test case verifies the ability to sync tasks with an external calendar (Google Calendar) and ensures task deadlines are displayed correctly in the user's calendar view.

****Test Suite:**** Calendar Integration

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

****Preconditions:****

- * User has a Google account.
- * User has granted the Task Management Tool app access to their Google Calendar.

****Test Data:****

* A new task with a specific deadline is created in the Task Management Tool.

****Test Steps:****

1. Log in to the Task Management Tool as a user.
2. Create a new task with a specific deadline.
3. Navigate to the "Calendar Sync" settings within the app.
4. Select Google Calendar as the sync target.
5. Authorize the app to access the user's Google Calendar.
6. Access the user's Google Calendar view.

****Postconditions:****

* The task with its corresponding deadline should be visible in the user's Google Calendar.

****Expected Result:**** The task is displayed in the user's Google Calendar with the accurate deadline.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TCC-CAL-002

****Test Case Title:**** Verify that users can set reminders for tasks via the calendar

****Test Case Description:**** This test case verifies the functionality of setting reminders for tasks directly from the calendar view.

****Test Suite:**** Calendar Integration

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

****Preconditions:****

* User has a Google Calendar account with reminders functionality enabled.

* User has synced their tasks with Google Calendar (as per TCC-CAL-001).

****Test Data:****

* A task with an upcoming deadline in the Task Management Tool.

****Test Steps:****

1. Log in to the Task Management Tool as a user.
2. Navigate to the "Calendar Sync" settings within the app.
3. Ensure Google Calendar is selected as the sync target.
4. Access the user's Google Calendar view.
5. Locate the task with the upcoming deadline.
6. Set a reminder for the task via the Google Calendar interface.

****Postconditions:****

* The user should receive a notification confirming the reminder has been set.

****Expected Result:**** A reminder is successfully set for the task in the user's Google Calendar.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TCC-CAL-003

****Test Case Title:**** Verify that calendar events update automatically when tasks are changed

****Test Case Description:**** This test case confirms that updates to task deadlines or other relevant information in the Task Management Tool are reflected automatically in the synced calendar events.

****Test Suite:**** Calendar Integration

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

****Preconditions:****

* User has synced their tasks with their Google Calendar (as per TCC-CAL-001).

* A task with a specific deadline exists in the Task Management Tool and is synced with the user's Google Calendar.

****Test Data:****

* New deadline for an existing task

****Test Steps:****

1. Log in to the Task Management Tool as a user.
2. Find the task that is synced with Google Calendar.
3. Modify the deadline for the task.
4. Observe the Google Calendar view within a defined timeframe (e.g., 5 minutes).

****Postconditions:****

* No need for manual refresh of the calendar.

****Expected Result:**** The deadline for the task in the Google Calendar is updated automatically to reflect the change made in the Task Management Tool.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TCC-CAL-004

****Test Case Title:**** Verify that users can manage calendar sync settings within the app

****Test Case Description:**** This test case verifies the functionality of allowing users to manage their calendar sync settings, including choosing sync targets and disabling synchronization.

****Test Suite:**** Calendar Integration

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

****Preconditions:****

* User has a Google Calendar account.

****Test Data:****

* User accounts for testing with different calendar providers (Google, Outlook)

****Test Steps:****

1. Log in to the Task Management Tool as a user.
2. Navigate to the "Calendar Sync" settings within the app.
3. Select Google Calendar as the sync target.
4. Verify successful authorization and sync with Google Calendar.
5. Remove authorization for Google Calendar.
6. Attempt to access Google Calendar sync functions; they should be disabled.
7. Repeat steps 3-6 with Outlook Calendar (if supported).
8. Access the "Sync & Settings" section, then "Calendar Notifications," and ensure users can personalize notification frequency and method.

****Postconditions:****

* See the sync status updates within the Calendar settings.

****Expected Result:**** Users can successfully manage their calendar sync settings, choosing different providers, enabling/disabling sync, and customizing notification preferences.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** System Testing, Functional Testing, Integration Testing, regression testing

****Test Case Approach:**** Positive, Negative

****Test Case ID:**** TCC-CAL-005

****Test Case Title:**** Verify that a conflict is displayed when scheduling a task that overlaps with an existing calendar event

****Test Case Description:**** This test case verifies the system's ability to prevent overlapping scheduling by displaying a conflict message when a task deadline is set during a pre-existing calendar event.

****Test Suite:**** Calendar Integration

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

****Preconditions:****

* User has a Google Calendar account.

* User has synced their tasks with Google Calendar.

* A pre-existing event is scheduled in the user's Google Calendar with a specific time slot.

****Test Data:****

- * New task with a deadline falling within the time slot of the existing calendar event.

****Test Steps:****

1. Log in to the Task Management Tool as a user.
2. Access the calendar view within the app, ensuring it reflects the synced Google Calendar.
3. Identify a pre-existing calendar event with a specific time slot.
4. Attempt to create a new task with a deadline that falls within the time slot of the existing calendar event.

****Postconditions:****

- * Feedback should indicate the potential conflict with no permanent task creation.

****Expected Result:**** The app displays a warning message indicating a scheduling conflict with the existing calendar event. The user cannot save the task with the overlapping deadline.

****Severity:**** Critical

****Priority:**** High

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

****Test Case Approach:**** Negative

Data - 36

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Duplication

Feature description -

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

Userstory title -

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

Acceptance criteria -

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

Generated Test Cases:

Test Cases for Task Duplication Feature

Test Case ID: TC_TASK_DUP_001

****Test Case Title:**** Verify that users can duplicate an existing task.

****Test Case Description:**** This test case verifies the functionality of duplicating an existing task and confirms that the duplicated task retains all original details.

****Test Suite:**** Task Management Tool - UI Functionality

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

****Preconditions:****

- * A user is logged into the Task Management Tool.
- * An existing task with details such as description, assignee, and deadline is created.

****Test Data:**** Existing task with populated fields

****Test Steps:****

1. Navigate to the task list view.
2. Select the existing task to be duplicated.
3. Click on the "Duplicate Task" button.
4. Verify that a new task is created with the same details as the original task.
5. Check the duplicated task's title, description, assignee, and deadline.

****Postconditions:****

- * The duplicated task remains visible in the task list.

****Expected Result:****

The duplicated task preserves all original details and is successfully added to the task list.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing, System Testing**

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

****Test Case ID:** TC_TASK_DUP_002**

****Test Case Title:** Verify that users can modify the duplicated task.**

****Test Case Description:** This test case verifies that a user can modify the details of a duplicated task after creation.**

****Test Suite:** Task Management Tool - UI Functionality**

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

****Preconditions:****

* The duplicated task from TC_TASK_DUP_001 is created and accessible.

****Test Data:**** Existing task details

****Test Steps:****

1. Click on the duplicated task to open its details.
2. Modify any field of the duplicated task (e.g., title, description, assignee, deadline).
3. Save the changes.
4. Verify that the modified details are reflected in the task details.

****Postconditions:****

* The modified task is saved and displayed in the task list.

****Expected Result:****

The user successfully modifies the duplicated task, and the changes are reflected in the task details.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

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

****Test Case ID:** TC_TASK_DUP_003**

****Test Case Title:** Verify that a clear indicator is provided when a task is duplicated.**

****Test Case Description:** This test case checks if the system provides visual or textual feedback to the user after duplicating a task.**

****Test Suite:** Task Management Tool - UI Feedback**

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

****Preconditions:****

* A user is logged into the Task Management Tool.

* An existing task is created.

****Test Data:** Existing task**

****Test Steps:****

1. Navigate to the task list view and select the existing task.
2. Click on the "Duplicate Task" button.

3. Observe the displayed UI element or message.

****Postconditions:****

****Expected Result:****

A clear indicator, such as a pop-up message or a visual cue, confirms the successful duplication of the task.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** System Testing, Usability Testing

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

****Test Case ID:**** TC_TASK_DUP_004

****Test Case Title:**** Verify that duplicated tasks are saved and displayed in the task list.

****Test Case Description:**** This test case ensures that duplicated tasks are persisted in the system and appear in the user's task list.

****Test Suite:**** Task Management Tool - Data Persistence

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

****Preconditions:****

* A user is logged into the Task Management Tool.

* An existing task is created.

****Test Data:**** Existing task

****Test Steps:****

1. Navigate to the task list view and select the existing task.
2. Click on the "Duplicate Task" button.
3. Verify that the duplicated task appears in the task list.
4. Check if the duplicated task persists after refreshing the page or logging out and back in.

****Postconditions:****

* The duplicated task remains in the task list.

****Expected Result:****

The duplicated task is saved and displayed in the task list, and the data persists across page refreshes and login

sessions.

****Severity:** Medium**

****Priority:** Medium**

****Type of Testing:** System Testing, Functional Testing**

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

****Test Case ID:** TC_TASK_DUP_005**

****Test Case Title:** Verify handling of duplication when a task is already assigned to multiple users**

****Test Case Description:** This test case simulates the scenario of duplicating a task that is already assigned to multiple users and checks for correct handling of assignments in the duplicated task.**

****Test Suite:** Task Management Tool - User Assignment**

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

****Preconditions:****

- * A user is logged into the Task Management Tool.
- * An existing task with multiple users assigned is created.

****Test Data:**** Existing task with multiple assignees.

****Test Steps:****

1. Navigate to the task list view and select the existing task with multiple assignees.
2. Click on the "Duplicate Task" button.
3. Verify that the duplicated task has retained the multiple user assignments.

****Postconditions:****

- * The duplicated task appears in the task list.

****Expected Result:****

The duplicated task correctly preserves and maintains the list of multiple user assignments from the original task.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, System Testing

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

These test cases provide a starting point for comprehensive testing of the Task Duplication feature in the Task Management Tool. More test cases can be added to cover various edge cases, error scenarios, and user flows specific to the application.

Data - 37

Project name -

Task Management Tool

Project description -

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

Feature name -

Task History

Feature description -

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating

accountability, and transparency in task management.

Userstory title -

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

Acceptance criteria -

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

Generated Test Cases:

Test Cases for Task History Feature

****Test Case ID:** TCH-001**

****Test Case Title:** Verify that task history displays detailed changes for modifications.**

****Test Case Description:** This test case verifies that the task history accurately reflects all modifications made to a task, including changes to its status, description, due date, and assigned user.**

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

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

****Preconditions:****

* A new task is created with initial information (title, description, due date, assigned user).

* A logged-in user has permission to edit the task.

* The user is viewing the task details page.

****Test Data:** Sample task data with initial values and modified values for each field**

****Test Steps:****

1. Open the task details page for the newly created task.
2. Modify the task's status, description, due date, and assigned user.
3. Click the "Save" button to apply the changes.
4. Navigate to the "Task History" section of the task details page.
5. Verify that the history entries accurately reflect each modification made to the task, including the specific field changed and the new value.

****Postconditions:**** Task is saved with modified values; User is still viewing the task details page.

****Expected Result:**** The task history section displays detailed entries for each modification made to the task, including the field changed, the old value, the new value, and the timestamp of the change.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TCH-002

****Test Case Title:**** Verify that task history includes comments added to tasks.

****Test Case Description:**** This test case verifies that any comments added to a task are recorded and displayed in the task history.

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

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

****Preconditions:****

* A new task is created with initial information (title, description, due date).

* A logged-in user has permission to view and comment on tasks.

****Test Data:**** Sample task data; placeholder comment text

****Test Steps:****

1. Open the task details page for the newly created task.
2. Enter a comment about the task in the designated comment field.
3. Click the "Add Comment" button to save the comment.
4. Navigate to the "Task History" section of the task details page.
5. Verify that the comment is displayed in the history, including the commenter's name (if applicable) and the timestamp of the comment.

****Postconditions:**** Comment is saved on the task; User is still viewing the task details page.

****Expected Result:**** The task history accurately reflects the added comment, including the comment text, commenter, and timestamp.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TCH-003

****Test Case Title:**** Verify that task history can be filtered by change type.

****Test Case Description:**** This test case verifies the functionality of filtering the task history by the type of change made, such as "Modified by", "Comment Added", or "Deadline Changed."

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

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

****Preconditions:****

* A task is created with initial information (title, description, due date).

* Multiple changes are made to the task, including modifications, comments, and a change to the due date.

****Test Data:**** Task data with diverse changes (status modifications, assigned user changes, comment additions, due date adjustments).

****Test Steps:****

1. Open the task details page for the task.
2. Navigate to the "Task History" section.
3. Utilize the available filter options (e.g., drop-down menu or checkbox) to select specific change types.
4. Verify that the displayed history entries only include changes matching the selected filter criteria.
5. Repeat steps 3 and 4 with different filter selections.

****Postconditions:**** User is still viewing the task details page; Filters applied/changed within the "Task History" section.

****Expected Result:**** The task history correctly filters based on the selected change type, presenting only relevant entries.

****Severity:**** Major

****Priority:**** Medium

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

****Test Case ID:** TCH-004**

****Test Case Title:** Verify that task history can be filtered by date.**

****Test Case Description:** This test case verifies the functionality of filtering the task history by date, allowing users to view changes within a specific timeframe.**

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

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

****Preconditions:****

* Multiple tasks with differing update times and creation dates are available.

* The task history section is accessible.

****Test Data:** Tasks with timestamps across multiple days and weeks.**

****Test Steps:****

1. Navigate to the "Task History" section of a task.
2. Utilize the date filtering options (e.g., calendar selection or date range input) to choose a specific date or date range.
3. Verify that the displayed history entries align with the selected date filter criteria.
4. Repeat steps 2 and 3 with different date selections (e.g., yesterday, this week, last month).

****Postconditions:** Date filter applied within the "Task History" section.**

****Expected Result:** The task history accurately filters entries based on the chosen date, displaying only changes within the selected timeframe.**

****Severity:** Major**

****Priority:** Medium**

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

****Test Case Approach:** Positive & Negative (Ensure negative date ranges or invalid input are handled appropriately)**

****Test Case ID:** TCH-005**

****Test Case Title:** Verify that task history timestamps are accurate and consistent.**

****Test Case Description:** This test case ensures that the timestamps recorded in the task history reflect the actual time of each change, and are displayed consistently across different browsers and devices.**

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

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

****Preconditions:****

- * A task is created with initial information.
- * Users can add comments and edit task details.
- * Time synchronization is ensured across testing environment and devices.

****Test Data:** Different timezones (Consider testing across different time zones to verify timezone accuracy)**

****Test Steps:****

1. Open the task details page and perform several modifications (status change, comment addition, description update).
2. Observe the timestamp for each change recorded in the task history.

3. Repeat the process from different devices and browsers, with varying time zones set.
4. Cross-reference the recorded timestamps with the system clock and local time on each device.

****Postconditions:**** Task history is displayed with timestamps.

****Expected Result:**** All timestamps in the task history are accurate, reflecting the time of each change in the respective user's time zone.

Data - 38

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

Userstory title -

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task

progress.

Acceptance criteria -

['Users receive notifications for task updates, comments, mentions, and other relevant activities.',

'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.',

'The system logs notification history for future reference.']

Generated Test Cases:

Test Cases for Task Management Tool: Task Notifications

****User Story:**** As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

Test Case 1:

****Test Case ID:**** TC_TASK_NOTIF_001

****Test Case Title:**** Verify that users receive email notifications for task updates.

****Test Case Description:**** This test case verifies that users receive email notifications when a task assigned to them is updated by another user.

****Test Suite:**** Notification System

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

****Preconditions:****

- * A user account with email notifications enabled.
- * A task created and assigned to the user.
- * A valid email address associated with the user account.

****Test Data:**** A sample task with updated details

****Test Steps:****

1. Log in as the user.
2. Create a new task and assign it to the user.
3. Update the task details (e.g., status, due date) from another user account.
4. Check the user's email inbox for a notification regarding the task update.

****Postconditions:****

* Email notification received by the user should contain the relevant task details.

****Expected Result:**** The user receives an email notification confirming the task update.

****Severity:**** Critical

****Priority:**** High

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

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

Test Case 2:

****Test Case ID:**** TC_TASK_NOTIF_002

****Test Case Title:**** Verify that users can customize notification settings.

****Test Case Description:**** This test case verifies that users can configure their desired notification methods and frequencies for task updates.

****Test Suite:**** User Settings

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

****Preconditions:****

* A user account logged in.

****Test Data:****

* Various notification settings (email, push, in-app, frequency)

****Test Steps:****

1. Navigate to the user profile settings.
2. Locate the notification settings section.
3. Select different notification methods and frequencies.

4. Save the changes to the notification settings.
5. Simulate a task update (e.g., update status) and check the chosen notification channels.

****Postconditions:****

- * Notification settings should be saved and applied correctly.

****Expected Result:**** The user receives notifications through the selected methods and frequency.

****Severity:**** Major

****Priority:**** Medium

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

****Test Case Approach:**** Positive/Negative

Test Case 3:

****Test Case ID:**** TC_TASK_NOTIF_003

****Test Case Title:**** Verify that users receive push notifications for task updates.

****Test Case Description:**** This test case verifies if users receive push notifications on their devices when a task assigned to them is updated.

****Test Suite:**** Mobile App Functionality

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

****Preconditions:****

- * A user account logged in on a mobile device with push notifications enabled.

- * A task created and assigned to the user via the mobile app.

****Test Data:**** A sample task with updated details

****Test Steps:****

1. Log in to the mobile app as the user.
2. Create a new task and assign it to the user.
3. Update the task details from a different device (web or another mobile app).
4. Verify if the user receives a push notification on their device.

****Postconditions:****

* Device notification should appear promptly after the task update.

****Expected Result:**** The user receives a push notification confirming the task update.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing & Performance Testing

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

Test Case 4:

****Test Case ID:**** TC_TASK_NOTIF_004

****Test Case Title:**** Verify that users receive in-app alerts for task updates.

****Test Case Description:**** This test case verifies whether in-app alerts effectively notify users about task updates when using the application.

****Test Suite:**** In-App User Interface

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

****Preconditions:****

* A user account logged in within the application.

* A task assigned to the user.

****Test Data:**** A sample task with updated details

****Test Steps:****

1. Log in to the application as the user.
2. Create a new task and assign it to the user.
3. Simulate a task update from another user or device.
4. Verify if an in-app alert appears notifying the user about the update.

****Postconditions:****

* The alert should be non-intrusive and easily noticeable.

****Expected Result:**** The user receives an in-app alert confirming the task update.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing & Usability Testing**

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

Test Case 5:

****Test Case ID:** TC_TASK_NOTIF_005**

****Test Case Title:** Verify that notifications include detailed task information.**

****Test Case Description:** This test case ensures that notifications provide all necessary details about the task update for users to understand the context effectively.**

****Test Suite:** Notification Content**

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

****Preconditions:****

- * A user account logged in.
- * A task with specific details (description, assigned user, category, etc.)

****Test Data:** A sample task with various details**

****Test Steps:****

1. Create a new task with detailed information.
2. Update the task from another account.
3. Check the notification received by the original user.
4. Verify if the notification includes:
 - * Task name
 - * Updated details (e.g., status, due date)
 - * Details of the updating user
 - * Link to the task view

****Postconditions:****

- * The notification should be informative and actionable.

****Expected Result:**** The notification accurately reflects the task updates with all relevant information.

****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:

Task Management Tool: Task Archiving Test Cases

****Test Case ID:** TC_TASK_ARCHIVING_001**

****Test Case Title:** Verify that completed tasks can be archived**

****Test Case Description:**** This test case verifies the functionality of archiving completed tasks. A user will mark a task as complete and then attempt to archive it. The system should successfully move the task to the archived section while retaining all original details.

****Test Suite:** Task Archiving**

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

****Preconditions:****

- * A user account is logged in with permission to create and manage tasks.
- * A new task is created with a relevant description and deadline.
- * The task is marked as complete.

****Test Data:****

* Task title: "Test Archive Task"

* Task description: "A test task for archiving."

* Task deadline: "2023-12-31"

****Test Steps:****

1. Navigate to the task list.
2. Locate the created task and mark it as "Complete."
3. Locate the archive option for the task and click it.
4. Verify that the task is successfully moved from the active task list to the archived section.
5. Click on the archived task to view its details.
6. Confirm that all original details (title, description, deadline) are retained.

****Postconditions:****

- * The archived task remains accessible in the designated archived section.

****Expected Result:**** The completed task is successfully archived and retains all original details.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

****Test Case ID:**** TC_TASK_ARCHIVING_002

****Test Case Title:**** Verify that archived tasks can be searched and viewed

****Test Case Description:**** This test case verifies the functionality of searching and viewing archived tasks. A user will search for an archived task by its title and confirm its presence in the search results.

****Test Suite:**** Task Archiving

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

****Preconditions:****

- * A user account is logged in with permission to access archived tasks.
- * A test task is archived in the previous test case.

****Test Data:****

- * Archived task title: "Test Archive Task"

****Test Steps:****

1. Navigate to the archived tasks section.
2. Enter the archived task title ("Test Archive Task") in the search bar and click "Search."
3. Verify that the archived task appears in the search results.
4. Click on the archived task to view its details.
5. Confirm all original details are present and accurate.

****Postconditions:****

- * The archived task is found and displayed in the search results.

****Expected Result:**** The archived task is successfully located and displayed in the search results.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

****Test Case ID:**** TC_TASK_ARCHIVING_003

****Test Case Title:**** Verify that users can restore archived tasks

****Test Case Description:**** This test case verifies the functionality of restoring archived tasks. A user will locate an archived task and use the restore option to move it back to the active task list.

****Test Suite:**** Task Archiving

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

****Preconditions:****

- * A user account is logged in with permission to restore archived tasks.
- * A task is archived in a previous test case (TC_TASK_ARCHIVING_001).

****Test Data:****

- * Archived task title: "Test Archive Task"

****Test Steps:****

1. Navigate to the archived tasks section.
2. Locate the archived task ("Test Archive Task").
3. Click on the "Restore" button associated with the task.
4. Verify that the task is successfully moved back to the active task list.
5. Confirm the task now appears in the active task list with the same details as before archiving.

****Postconditions:****

- * The restored task is active and visible in the active task list.

Expected Result:****** The archived task is successfully restored and appears in the active task list with all its original information.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

****Test Case ID:**** TC_TASK_ARCHIVING_004

****Test Case Title:**** Verify that archived tasks retain all original details and history

****Test Case Description:**** This test case verifies whether archived tasks preserve their history and details.

****Test Suite:**** Task Archiving

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

****Preconditions:****

- * User is logged in with access to archived tasks.

- * A task is created, completed, and archived with details.

****Test Data:****

- * Task title: "Test Archive Details"

- * Task description: "This task includes detailed information."

* Task deadline: "2023-11-30"

* Attachments: 2-3 relevant images.

****Test Steps:****

1. Navigate to the archived tasks section.
2. Locate the archived task ("Test Archive Details").
3. View the details of the task.
4. Examine the following aspects:

- * Task title and description

- * Deadline date and time

- * Attached files

- * Any relevant comments or updates associated with the task from the active state.

****Postconditions:****

- * Archived task details remain intact.

****Expected Result:**** All original details and history of the task, including attachments and comments, are retained within the archived task.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

****Test Case ID:** TC_TASK_ARCHIVING_005**

****Test Case Title:** Verify that the "Archive" and "Restore" actions are unavailable for certain user roles**

****Test Case Description:** This test case verifies that users with guest or read-only access are prevented from archiving or restoring tasks.**

****Test Suite:** Task Archiving**

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

****Preconditions:****

* Two user accounts: Admin (full permissions) and Guest (read-only).

****Test Data:****

* A task is created.

****Test Steps:****

1. ****Admin user:****

- * Navigates to the task.
- * Clicks the "Archive" button.
- * Verifies that the archive action is successfully performed.

2. ****Guest user:****

- * Navigates to the task.
- * Attempts to click the "Archive" button.
- * Verifies they are prevented from archiving the task and an appropriate error message is displayed (e.g.,

"Insufficient permissions").

- * Attempts to restore an archived task.

* Verifies they are prevented from restoring the task and an appropriate error message is displayed.

****Postconditions:****

* Admin user successfully archives and restores tasks.

* Guest user cannot archive or restore tasks.

****Expected Result:**** Only users with appropriate permissions can perform archiving and restoring operations on tasks.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Security Testing

****Test Case Approach:**** Positive, Negative

Data - 40

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users

should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Guest Access

Feature description -

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

Userstory title -

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

Acceptance criteria -

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

Generated Test Cases:

Test Cases for Guest Access Feature

****Project Name:** Task Management Tool**

****Feature Name:** Guest Access**

****User Story:** As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.**

Test Case 1

****Test Case ID:** TC_GUEST_001**

****Test Case Title:** Verify that users can invite external collaborators as guests.**

****Test Case Description:** This test case verifies the functionality of inviting external collaborators as guests to access specific tasks.**

****Test Suite:** Guest Access**

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

****Preconditions:****

* A logged-in user with administrative privileges.

* At least one task exists within a project.

****Test Data:****

* Valid email address of an external collaborator.

****Test Steps:****

1. Navigate to the project containing the task.
2. Select the task to be shared with a guest.
3. Click on the "Invite Guest" button.
4. Enter the external collaborator's email address.
5. Click "Send Invitation".

****Postconditions:****

* An invitation email is sent to the specified email address.

* The invited collaborator is added as a guest to the task.

****Expected Result:** The system successfully sends an invitation email and grants guest access to the specified task.**

****Severity:** Major**

****Priority:** High**

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

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

Test Case 2

****Test Case ID:** TC_GUEST_002**

****Test Case Title:** Verify that guests can view task details.**

****Test Case Description:** This test case verifies that guests can access the details of tasks they have been invited to.**

****Test Suite:** Guest Access**

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

****Preconditions:****

* An invited external collaborator has accepted the invitation.

* A task exists within a project.

****Test Data:****

* N/A

****Test Steps:****

1. Log in as the invited guest.

2. Access the project containing the task.

3. Open the chosen task.

****Postconditions:****

* No changes are made to the task.

****Expected Result:** The guest user can view all the details of the task, including description, attachments, and assigned users.**

****Severity:** Major**

****Priority:** High**

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

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

Test Case 3

****Test Case ID:** TC_GUEST_003**

****Test Case Title:** Verify that guests can participate in task discussions.**

****Test Case Description:** This test case verifies that guests can contribute to task discussions.**

****Test Suite:** Guest Access**

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

****Preconditions:****

* An invited external collaborator has accepted the invitation.

* A task exists within a project.

****Test Data:****

* Sample comment to be added by the guest user.

****Test Steps:****

1. Log in as the invited guest.
2. Access the project containing the task.
3. Open the chosen task.
4. Type a comment in the designated discussion area.
5. Click "Post Comment".

****Postconditions:****

* No changes are made to the task.

****Expected Result:** The guest user's comment is successfully added to the task discussion thread.**

****Severity:** Major**

****Priority:** High**

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

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

Test Case 4

****Test Case ID:** TC_GUEST_004**

****Test Case Title:** Verify that guests cannot access sensitive areas of the platform.**

****Test Case Description:** This test case verifies that guests are restricted from accessing sensitive areas of the platform.**

****Test Suite:** Guest Access**

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

****Preconditions:****

* Guest user is logged in.

* The platform has defined sensitive areas (e.g., user profile settings, billing information).

****Test Data:****

* N/A

****Test Steps:****

1. Access a sensitive area (e.g., User Profile Settings).

****Expected Result:** The guest user encounters a permission denial message or is redirected to a login screen, preventing access to sensitive areas.**

****Severity:** Blocker**

****Priority:** High**

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

****Test Case Approach:** Negative**

Test Case 5

****Test Case ID:** TC_GUEST_005**

****Test Case Title:** Verify that users can manage and revoke guest access.**

****Test Case Description:** This test case verifies the functionality of managing and revoking guest access to tasks.**

****Test Suite:** Guest Access**

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

****Preconditions:****

* A logged-in user with administrative privileges has invited a guest collaborator.

****Test Data:****

* The email address of a guest collaborator.

****Test Steps:****

1. Navigate to the project containing the task shared with the guest.
2. Find the guest user's listing.
3. Select the option to revoke guest access (e.g., "Remove Guest").
4. Confirm the revocation action.

****Postconditions:****

* The guest user is no longer listed as a collaborator on the task.

****Expected Result:**** The system successfully removes the guest collaborator from the task, revoking their access.

****Severity:**** Major

****Priority:**** Medium

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

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

Task Management Tool: Task Import/Export Test Cases

****Test Suite:**** Task Management

****Project Name:**** Task Management Tool

****User Story:**** As a user, I want to import and export tasks so that I can migrate data between different task management tools.

Test Case 1: Verify that Importing a CSV Task List Successfully Populates the System

****Test Case ID:**** TC_IMP001

****Test Case Title:**** Verify that importing a CSV task list successfully populates the system.

****Test Case Description:**** This test case focuses on the successful import of a CSV file containing task data into the task management tool.

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

****Preconditions:****

- * A valid CSV file containing task data (title, description, due date, assigned user) is created.
- * The user is logged into the task management tool with appropriate permissions.

****Test Data:** Valid CSV file containing task data.**

****Test Steps:****

1. Navigate to the import section within the task management tool.
2. Select the CSV file from the local system.
3. Review the field mapping provided.
4. Click the "Import" button.
5. Review the imported tasks list for accuracy.

****Postconditions:****

- * The imported tasks display correctly in the task management tool.

****Expected Result:** All tasks from the CSV file are successfully imported, with fields correctly mapped.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** System Testing, Functional Testing**

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

Test Case 2: Verify that Importing a CSV with Incorrect Formatting Results in an Error

****Test Case ID:** TC_IMP002**

****Test Case Title:** Verify that importing a CSV with incorrect formatting results in an error.**

****Test Case Description:** This test case aims to ensure the system handles invalid CSV formatting gracefully and provides informative error messages.**

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

****Preconditions:****

* A CSV file with intentional formatting errors (e.g., missing header, inconsistent data types) is created.

* The user is logged into the task management tool with appropriate permissions.

****Test Data:** Invalid CSV file with intentional formatting errors.**

****Test Steps:****

1. Navigate to the import section within the task management tool.
2. Select the invalid CSV file from the local system.
3. Attempt to import the file.

****Postconditions:****

* The system displays a clear error message indicating the issue with the CSV format.

****Expected Result:** The import process fails, and the system displays a comprehensive error message explaining the reason for the failure.**

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** System Testing, Functional Testing**

****Test Case Approach:** Negative**

Test Case 3: Verify that Mapping Fields Correctly During Import

****Test Case ID:** TC_IMP003**

****Test Case Title:** Verify that mapping fields correctly during import.**

****Test Case Description:** This test case confirms that the user can map imported fields from the CSV to the corresponding fields in the system, ensuring data integrity.**

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

****Preconditions:****

- * A CSV file is created with tasks in a slightly different field order than the system.
- * The user is logged into the task management tool with appropriate permissions.

****Test Data:** CSV file with tasks in a different field order than the system**

****Test Steps:****

1. Navigate to the import section within the task management tool.
2. Select the CSV file from the local system.
3. Drag and drop the CSV fields onto the corresponding fields in the system mapping panel.
4. Review the field mapping to ensure accuracy and click "Import."

****Postconditions:****

* Imported tasks are displayed correctly in the task management tool with mapped fields.

****Expected Result:**** The imported tasks populate the system correctly, with data mapped to the appropriate fields according to the user's selections.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** System Testing, Functional Testing

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

Test Case 4: Verify that Exporting Tasks to CSV Includes All Relevant Fields

****Test Case ID:**** TC_EXP001

****Test Case Title:**** Verify that exporting tasks to CSV includes all relevant fields.

****Test Case Description:**** This test case examines whether exporting tasks to CSV captures all essential data points.

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

****Preconditions:****

* Tasks with various properties (title, description, due date, assignee, category) are created in the system.

* The user is logged into the task management tool with appropriate permissions.

****Test Data:**** A set of tasks with diverse attributes in the system

****Test Steps:****

1. Navigate to the export section within the task management tool.
2. Select "CSV" as the export format.
3. Ensure all tasks are selected for export.
4. Click the "Export" button.

****Postconditions:****

* A CSV file is downloaded containing the exported tasks.

****Expected Result:**** The exported CSV file contains all relevant data fields associated with the selected tasks, including title, description, due date, assignee, and category.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** System Testing, Functional Testing

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

Test Case 5: Verify that Importing and Exporting Tasks Logs Activities

****Test Case ID:**** TC_LOG001

****Test Case Title:**** Verify that importing and exporting tasks logs activities.

****Test Case Description:**** This test case ensures the system maintains a record of import and export actions for audit and tracking purposes.

****Test Priority:**** Low

****Preconditions:****

* The user has permission to view system logs.

* The "Task Management Tool" is running.

Test Data: No test data needed.

Test Steps:

1. Import a CSV file into the task management tool.
2. Export a set of tasks as a CSV file.
3. Access the system logs.

Postconditions:

* Logs record both the import and export actions.

Expected Result: The system logs clearly record the import and export operations, including timestamps, user information, file names, and potentially other relevant details.

Severity: Minor

Priority: Low

Type of Testing: System Testing, Regression Testing

Test Case Approach: Positive

Data - 42

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.

Userstory title -

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

Acceptance criteria -

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.', 'Priority changes are logged and tracked.']

Generated Test Cases:

Test Cases for Task Prioritization Feature in Task Management Tool

****Test Case ID:**** TC_TaskPrioritization_01

****Test Case Title:**** Verify that users can set initial task priorities.

****Test Case Description:**** This test case verifies the functionality of setting the initial priority of a new task. It will cover different priority levels and ensure correct data persistence.

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

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

****Preconditions:****

* A valid user account is logged in.

* A new task is created.

* The system is running properly.

****Test Data:****

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

****Test Steps:****

1. Navigate to the task creation form.

2. Enter essential task details (title, description, deadline).

3. From the priority dropdown menu, select a specific priority level (e.g., High).

4. Click on the "Create Task" button.

5. Verify that the newly created task displays the selected priority level.

****Postconditions:****

* The task should be saved in the system.

****Expected Result:**** The task should be created with the selected priority level reflected in its display.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TC_TaskPrioritization_02

****Test Case Title:**** Verify that users can change existing task priorities.

****Test Case Description:**** This test case verifies the functionality of modifying the priority of an already existing task. It will involve updating a task's priority and ensuring the change persists.

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

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

****Preconditions:****

* A valid user account is logged in.

* A task with an initial priority is created.

****Test Data:****

* Existing task with a specific priority level (e.g., Medium)

****Test Steps:****

1. Open the existing task.
2. Locate the priority dropdown menu.
3. Select a different priority level (e.g., High).
4. Click on the "Update Task" button.
5. Verify that the task's priority has changed to the newly selected level.

****Postconditions:****

* The task should be updated in the system.

****Expected Result:**** The task's priority will be successfully modified and reflect the new selection.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:** TC_TaskPrioritization_03**

****Test Case Title:** Verify that tasks can be sorted and filtered by priority levels.**

****Test Case Description:** This test case verifies the system's ability to sort and filter tasks based on their priority levels.**

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

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

****Preconditions:****

- * A valid user account is logged in.
- * At least two tasks with different priority levels are created.

****Test Data:****

- * Tasks with varying priority levels (e.g., High, Medium, Low)

****Test Steps:****

1. Navigate to the task list view.
2. Sort the tasks by priority (ascending or descending) using the available sorting options.
3. Verify that the tasks are displayed in the correct order based on the selected priority level.
4. Apply a filter to display only tasks with a specific priority level (e.g., High).
5. Verify that only tasks with the selected priority are displayed.

****Postconditions:** The task list remains sorted or filtered according to the user's selections.**

****Expected Result:**** Tasks will be sorted and filtered accurately based on the selected priority levels.

****Severity:**** Major

****Priority:**** High

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

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

****Test Case ID:**** TC_TaskPrioritization_04

****Test Case Title:**** Verify that priority changes are logged and trackable.

****Test Case Description:**** This test case verifies if the system accurately logs and displays changes in task priorities.

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

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

****Preconditions:****

* A valid user account is logged in.

* A task is created with a specific initial priority.

****Test Data:****

* A task with an initial priority level.

****Test Steps:****

1. Open the task details.
2. Change the task priority level to a different one.

3. Save the changes.
4. Access the task history or audit log.
5. Verify that a record of the priority change is documented, including the original and new priority levels, and the timestamp of the change.

****Postconditions:**** The task history reflects the priority change.

****Expected Result:**** The system accurately logs the priority change, providing a clear record of modifications.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** functional Testing, Regression Testing

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

****Test Case ID:**** TC_TaskPrioritization_05

****Test Case Title:**** Verify system behavior when attempting to set invalid priority levels.

****Test Case Description:**** This test case verifies the system's response to invalid priority level inputs.

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

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

****Preconditions:****

* A valid user account is logged in.

* A task creation form is open.

****Test Data:****

- * Invalid priority levels (e.g., "None", "Extra High", "Very Urgent")

****Test Steps:****

1. Enter necessary task information.
2. Attempt to select an invalid priority level from the dropdown menu.

****Expected Result:****

- * The system should either display an error message indicating the invalid input, or prevent the selection of an invalid priority level altogether. The task creation process should not proceed with an invalid priority.

****Severity:**** Critical

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Boundary Value Analysis

****Test Case Approach:**** Negative

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:

Task Management Tool: Task Collaboration Test Cases

Here are five detailed test cases designed to validate the "Task Collaboration" feature of the Task Management

Tool, ensuring all acceptance criteria are met.

****Test Case 1:****

****Test Case ID:**** TCTM-001

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

****Test Case Description:**** This test case verifies the functionality of assigning tasks to specific team members within the system. It will test the process of selecting a team member from a list, assigning them to a task, and confirming the assignment is reflected in the task details.

****Test Suite:**** User Management

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

****Preconditions:****

- * A user account with administrator privileges is logged in.
- * At least two user accounts with different roles exist in the system.
- * Tasks have been created in the system.

****Test Data:****

- * Task ID: T-001
- * Assignee user: JohnDoe

****Test Steps:****

1. Navigate to the task details page for task T-001.
2. Click the "Assign to" dropdown menu.
3. Select "JohnDoe" from the list of available team members.
4. Click the "Save" or "Assign" button to confirm the assignment.
5. Verify that JohnDoe is listed as the assignee for task T-001 in the task details page.

****Postconditions:****

- * JohnDoe should receive a notification about the task assignment.

****Expected Result:**** Task T-001 is successfully assigned to JohnDoe.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 2:****

****Test Case ID:**** TCTM-002

****Test Case Title:**** Verify that users can share updates and communicate through comments or messages.

****Test Case Description:**** This test case confirms the ability to leave comments and messages on tasks and ensures they are visible to appropriate team members.

****Test Suite:**** Task Management

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

****Preconditions:****

- * A user account with the " contributor" role is logged in.
- * A task has been created and assigned to a team member.

****Test Data:****

* Task ID: T-002

* Comment text: "Working on this task, will update by Friday."

****Test Steps:****

1. Navigate to the task details page for task T-002.
2. Click the "Add Comment" or "Add Message" button.

3. Type the comment text "Working on this task, will update by Friday."
4. Click the "Submit" or "Post" button.
5. Verify that the comment appears in the task history, visible to the assigned team member and other collaborators.

****Postconditions:****

* The comment creator should receive a notification when the comment is added.

****Expected Result:**** The user successfully adds a comment to task T-002, which is visible to the assigned team member and other collaborating users.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 3:****

****Test Case ID:**** TCTM-003

****Test Case Title:**** Verify that task updates and comments are visible to assigned team members.

****Test Case Description:**** This test case ensures that only authorized users (assigned team members and collaborators) can see task updates and comments.

****Test Suite:**** User Permissions

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

****Preconditions:****

* Two user accounts with different roles (one assigned to a task and one not assigned) exist.

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

****Test Data:****

* Task ID: T-003

* Comment text: "Task is moving forward well, expect completion by next week."

****Test Steps:****

1. Login as the assigned team member.

2. Verify that the task details page for task T-003 displays all previous comments and updates.

3. Login as a user not assigned to the task.

4. Navigate to the task details page for task T-003.

5. Verify that the task details page does not display any comments or updates made by the assigned team member.

****Postconditions:****

* N/A

****Expected Result:**** The assigned team member can view all task updates and comments, while users not assigned to the task cannot.

****Severity:**** Major

****Priority:**** Medium

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

****Test Case Approach:**** Positive/Negative

****Test Case 4:****

****Test Case ID:**** TCTM-004

****Test Case Title:**** Verify that users receive notifications for task updates and comments.

****Test Case Description:**** This test case validates that users receive timely notifications for new task updates and comments, ensuring they stay informed about changes and discussions.

****Test Suite:**** Notifications

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

****Preconditions:****

- * A user account is logged in.
- * A task has been created and assigned to them.
- * Notification settings are enabled for the user account.

****Test Data:****

- * Task ID: T-004
- * Update text: "Milestone 1 achieved, now moving to development phase."

****Test Steps:****

1. Manually trigger a task update for task T-004.
2. Login as the user assigned to task T-004.
3. Verify that the user receives a notification about the task update.
4. Optionally, verify that the notification indicates the specific change made ("Milestone 1 achieved, now moving to development phase.").

****Postconditions:****

- * The user should be able to acknowledge or dismiss the notification.

****Expected Result:**** The user receives a notification promptly upon any updates or comments added to their assigned tasks.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 5:****

****Test Case ID:**** TCTM-005

****Test Case Title:**** Verify that the system provides a clear view of task collaboration history.

****Test Case Description:**** This test case ensures that the system effectively logs and displays the entire history of task collaborations, including updates, comments, and assignments.

****Test Suite:**** Task History

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

****Preconditions:****

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

* Multiple comments, updates, and assignments have been made to the task.

****Test Data:****

* Task ID: T-005

* Task creation date: 2023-10-26

****Test Steps:****

1. Navigate to the task details page for task T-005.
2. Verify that a chronological list of all task updates, comments, and assignment changes is displayed.
3. Verify the list includes the author, timestamp, and content of each update, comment, or assignment.
4. Check if the sorting is chronological, with the most recent changes appearing first.

****Postconditions:****

* N/A

****Expected Result:**** The task history page provides a clear and comprehensive record of all collaboration activities performed on task T-005.

****Severity:**** Major

****Priority:**** Medium

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

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

These test cases provide a starting point for ensuring the quality and functionality of the "Task Collaboration" feature in the Task Management Tool.

Data - 44

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tracking

Feature description -

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims

to provide visibility into task progress and ensure timely completion of tasks.

Userstory title -

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

Acceptance criteria -

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.', 'System provides visual indicators for task progress.', '']

Generated Test Cases:

Here are 5 detailed test cases for the "Task Tracking" feature of the Task Management Tool, based on the provided acceptance criteria and test case template:

****Test Case 1****

* **Test Case ID:** TC_TASK_TRACK_01

* **Test Case Title:** Verify that users can view task statuses correctly.

* **Test Case Description:** This test case verifies that the system accurately displays the status of tasks (e.g., Not Started, In Progress, Completed, Blocked) to the user.

* **Test Suite:** Task Tracking

* **Test Priority:** High

* **Preconditions:**

- * A user is logged into the system.

- * There are at least three tasks in the system with different statuses.

* **Test Data:** Existing tasks with varying statuses (Not Started, In Progress, Completed, Blocked)

* **Test Steps:**

1. Navigate to the task list view.
2. Identify a task with each of the different statuses.
3. Verify that the displayed status for each task is accurate.

* **Postconditions:** No specific postconditions

* **Expected Result:** The system should correctly display the status of each task as defined in the test data.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, System Testing, Regression Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TC_TASK_TRACK_02

* **Test Case Title:** Verify that users can track task progress visually.

* **Test Case Description:** This test case checks if the system uses visual indicators (e.g., progress bars, color coding) to represent the progress of tasks effectively.

* **Test Suite:** Task Tracking

* **Test Priority:** High

* **Preconditions:**

- * A user is logged into the system.

- * A task with progress below 100% is available.

* **Test Data:** A task with a defined progress percentage (e.g., 60%)

* **Test Steps:**

1. Open the task details.
2. Observe the visual indicator for task progress.
3. Verify that the visual indicator accurately reflects the progress percentage of the task.

* **Postconditions:** No specific postconditions.

* **Expected Result:** The visual indicator should accurately depict the task's progress percentage.

* **Severity:** Critical

* **Priority:** High

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

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TC_TASK_TRACK_03

* **Test Case Title:** Verify that tasks are updated in real-time.

* **Test Case Description:** This test case validates the real-time update functionality for tasks. When a task's status or progress is changed, the system should reflect the update instantly.

* **Test Suite:** Task Tracking

* **Test Priority:** High

* **Preconditions:**

- * A user is logged into the system.

- * A task is available in a specific status (e.g., "In Progress").

* **Test Data:** A task with a starting status of "In Progress"

* **Test Steps:**

1. Open the task details.
2. Manually change the task status to "Completed" (or another relevant status) through the user interface.
3. Observe the system for immediate updates.
4. Verify that the task status has been updated in real-time across all views related to the task.

* **Postconditions:** No specific postconditions

* **Expected Result:** The task status should change instantly to the updated value in all relevant locations within the system.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, System Testing, Performance Testing, Regression Testing

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TC_TASK_TRACK_04

* **Test Case Title:** Verify task progress tracking for tasks with no set deadline.

* **Test Case Description:** This test case ensures that the system correctly tracks the progress of tasks that do not have a deadline assigned.

* **Test Suite:** Task Tracking

* **Test Priority:** Medium

* **Preconditions:**

- * A user is logged into the system.

- * A task is created without a deadline set.

* **Test Data:** A new task without a due date.

* **Test Steps:**

1. Create a new task without setting a deadline.
2. Progress the task through various stages (e.g., 20%, 50%, 80%, 100%).
3. Observe the visual progress indicator.
4. Verify that the progress indicator accurately reflects the tasks' progression despite the lack of a deadline.

* **Postconditions:** No specific postconditions

* **Expected Result:** The progress indicator should update correctly and reflect the task's progress accurately.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, System Testing, Regression Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC_TASK_TRACK_05

* **Test Case Title:** Verify that task status can be changed to "Blocked" and identify the reason.

* **Test Case Description:** This test case ensures the functionality of the "Blocked" status and its association with a reason for blockage.

* **Test Suite:** Task Tracking

* **Test Priority:** Medium

* **Preconditions:**

- * A user is logged into the system.

- * A task is present in the "In Progress" status.

* **Test Data:** A task with a status of "In Progress"

* **Test Steps:**

1. Open the task details.
2. Change the task status to "Blocked."
3. Enter a reason for the blockage (e.g., "Awaiting feedback").
4. Observe the display of the "Blocked" status and the entered reason.

* **Postconditions:** No specific postconditions

* **Expected Result:** The task status should change to "Blocked," and the entered reason for blockage should be displayed clearly on the task details page.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, System Testing, Regression 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:

Test Cases for Task Notifications Feature

Project: Task Management Tool

Feature: Task Notifications

User Story: As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

Test Case 1

Test Case ID: TC_NTF_001

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

Test Case Description: This test case verifies that notifications are sent to users when a new task is created and assigned to them.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- * A user account is created and logged in.
- * An administrator/existing user has the permission to create tasks.

****Test Data:**** A new task with assigned user.

****Test Steps:****

1. An administrator/existing user creates a new task and assigns it to the test user.
2. Wait for the notification to be delivered to the test user (e.g., via email, in-app notification).
3. The test user should see the notification clearly indicating a new task has been assigned to them.
4. Click on the notification and verify that it directs the test user to the newly created task.

****Postconditions:****

* Notification is acknowledged by the user or dismissed.

* The task is accessible to the assigned user.

****Expected Result:**** The test user receives a notification for the newly created task.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 2****

****Test Case ID:**** TC_NTF_002

****Test Case Title:**** Verify that users receive notifications for task updates and comments.

****Test Case Description:**** This test case verifies that notifications are sent to users when a task assigned to them is updated or a comment is added.

****Test Suite:**** Task Notifications

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

****Preconditions:****

- * A user account is created and logged in.
- * An existing task is assigned to the test user.

****Test Data:**** Task updates and comments.

****Test Steps:****

1. Log in as the test user.
2. Navigate to the assigned task.
3. Update the task status or deadline and add a comment on the task.
4. Wait for the notification to be delivered to the test user.
5. The test user should see a notification indicating the task has been updated/a comment has been added.

****Postconditions:****

- * Notification is acknowledged by the user or dismissed.

****Expected Result:**** The test user receives a notification for the task update or added comment.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 3****

****Test Case ID:**** TC_NTF_003

****Test Case Title:**** Verify that users receive reminders for upcoming deadlines.

****Test Case Description:**** This test case verifies that notifications are sent to users for tasks with upcoming deadlines.

****Test Suite:**** Task Notifications

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

****Preconditions:****

* A user account is created and logged in.

* A task with a set deadline is created and assigned to the test user.

****Test Data:**** A task with a future deadline.

****Test Steps:****

1. Set a specific deadline for a task and assign it to the test user.
2. Wait for the reminder notification to be delivered to the test user (date/time close to the deadline).
3. Verify that the notification contains the relevant task details and the approaching deadline.

****Postconditions:****

* Notification is acknowledged by the user or dismissed.

****Expected Result:**** The test user receives a reminder notification for the task's upcoming deadline.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case 4****

****Test Case ID:** TC_NTF_004**

****Test Case Title:**** Verify that notification settings can be customized by users.

****Test Case Description:**** This test case verifies that users can adjust their notification preferences (e.g., notification frequency, channel, types of events).

****Test Suite:**** Task Notifications

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

****Preconditions:****

* A user account is created and logged in.

****Test Data:**** Different notification settings (email, in-app, daily digest, silence for weekends, etc.).

****Test Steps:****

1. Navigate to the user settings or notification preferences section.
2. Change the notification settings according to various test scenarios.
3. Verify that the changes in settings are reflected in the notification options.
4. Create a new task and simulate events accordingly.
5. Check if notifications are received as per the customized settings.

****Postconditions:****

* Changes are saved and reflected in the user's profile.

****Expected Result:**** Users can successfully customize their notification settings, and notifications are delivered according to their chosen preferences.

****Severity:**** Major

****Priority:**** Medium

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

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

****Test Case 5****

****Test Case ID:** TC_NTF_005**

****Test Case Title:**** Verify handling of notification errors.

****Test Case Description:**** This test case verifies the system's response to potential errors in notification delivery (e.g., invalid email address, network connectivity issues).

****Test Suite:**** Task Notifications

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

****Preconditions:****

* User accounts with different email addresses.

****Test Data:****

* An invalid email address.

* Simulated network connectivity issues.

****Test Steps:****

1. Create a user account with an invalid email address.
2. Attempt to send a notification to this user.
3. Log the error message and its severity.
4. Simulate network issues and attempt to send notifications.
5. Observe the system's behavior and any error logs generated.

****Postconditions:****

* Error messages are logged with appropriate severity levels.

* The system gracefully handles connectivity issues and prevents notification failures.

****Expected Result:****

The system gracefully handles notification errors, logs them appropriately, and attempts to recover or inform the user about the issue.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Error Handling 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:

Task Management Tool: Task Categorization Test Cases

****Test Case ID:**** TC_TaskCategorization_01

****Test Case Title:**** Verify that users can create new categories.

****Test Case Description:**** This test case verifies the ability of users to create new categories for task organization.

****Test Suite:**** User Interface - Task Management

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

****Preconditions:****

* User is logged into the system.

****Test Data:****

* New category name: "Project X"

****Test Steps:****

1. Navigate to the "Categories" section within the task management tool.
2. Click on the "New Category" button.
3. Enter "Project X" as the name for the new category.
4. Click the "Create" button.
5. Verify that the new category "Project X" appears in the list of categories.

****Postconditions:****

* Category "Project X" is saved in the system.

****Expected Result:**** A new category named "Project X" is successfully created and displayed in the list.

****Severity:**** High

****Priority:**** High

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

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

****Test Case ID:**** TC_TaskCategorization_02

****Test Case Title:**** Verify that users can assign tasks to existing categories.

****Test Case Description:**** This test case verifies the ability of users to assign existing tasks to specific categories.

****Test Suite:**** User Interface - Task Management

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

****Preconditions:****

* User is logged into the system.

* An existing task is in the task list.

****Test Data:****

* Existing task ID: "Task 123"

* Category for assignment: "Project Y"

****Test Steps:****

1. Navigate to the "Tasks" section within the task management tool.
2. Select the task with ID "Task 123".
3. In the task details, locate the "Category" field.
4. Select "Project Y" as the category for task "Task 123".
5. Click the "Save" button.

****Postconditions:****

* Task "Task 123" is updated with the new category.

****Expected Result:**** The selected task "Task 123" is successfully assigned to the "Project Y" category.

****Severity:**** High

****Priority:**** High

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

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

... ****Continue adding test cases in this format to cover the remaining acceptance criteria****

****Example Test Cases:****

* ****TC_TaskCategorization_03:**** Verify that users can view tasks grouped by categories.

* ****TC_TaskCategorization_04:**** Verify that tasks can be moved between categories.

* ****TC_TaskCategorization_05:**** Verify that category information is accurately reflected in task views.

(Consider including various filter options and sorting)

Data - 47

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Search

Feature description -

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

Userstory title -

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search

results by category, priority, or due date.', 'Search results are accurate and relevant.', "]

Generated Test Cases:

Here are 5 test cases for the "Task Search" feature of the Task Management Tool, following the provided template:

Test Case 1

* **Test Case ID:** TC-TMS-SEARCH-01

* **Test Case Title:** Verify that users can search for tasks by name.

* **Test Case Description:** This test case will verify the functionality of searching for tasks by their name.

It will involve creating tasks with distinct names and then searching for them using the search bar.

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

* **Test Priority:** High

* **Preconditions:**

- * A user account is logged in.

- * At least three tasks with unique names are created.

* **Test Data:**

- * Task names: "Project A", "Meeting Notes", "Grocery Shopping"

* **Test Steps:**

1. Navigate to the task list page.
2. Enter "Project A" in the search bar.
3. Click the search button.
4. Verify that the "Project A" task appears in the search results.
5. Repeat steps 2-4 with "Meeting Notes" and "Grocery Shopping."

* **Postconditions:**

- * Search bar is cleared after each search.

- * The task list returns to its original state.

- * **Expected Result:** The search results accurately display tasks matching the entered name.

- * **Severity:** Major

- * **Priority:** High

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

- * **Test Case Approach:** Positive

Test Case 2

- * **Test Case ID:** TC-TMS-SEARCH-02

- * **Test Case Title:** Verify that users can search for tasks by description.

- * **Test Case Description:** This test case verifies the search functionality when using a task's description as the search criteria.

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

- * **Test Priority:** High

- * **Preconditions:**

- * A user account is logged in.

- * A task is created with a descriptive title and content.

- * **Test Data:**

- * Task description: "Prepare presentation for client meeting with detailed financial analysis."

- * **Test Steps:**

- 1. Navigate to the task list page.

- 2. Enter "financial analysis" in the search bar.

- 3. Click the search button.

4. Verify that the task with the description containing "financial analysis" appears in the results.

* **Postconditions:**

* Search bar is cleared.

* The task list returns to its original state.

* **Expected Result:** The search result accurately displays the task containing the specified description phrase.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TC-TMS-SEARCH-03

* **Test Case Title:** Verify that users can filter search results by category.

* **Test Case Description:** This test case verifies the functionality of filtering search results based on task categories.

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

* **Test Priority:** Medium

* **Preconditions:**

* A user account is logged in.

* At least two tasks are created in different categories (e.g., "Work", "Personal").

* **Test Data:**

* Task 1: Category = "Work", status = "In progress"

* Task 2: Category = "Personal", status = "Completed"

* **Test Steps:**

1. Navigate to the task list page.
2. Enter any term in the search bar.
3. Click on the "Category" filter drop-down.
4. Select the "Work" category.
5. Verify that only Task 1 (matching the "Work" category) is displayed in the search results.

* **Postconditions:**

- * The category filter is reset.
- * The task list returns to original state.

* **Expected Result:** The search results are accurately filtered to show tasks belonging to the selected category.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TC-TMS-SEARCH-04

* **Test Case Title:** Verify that users can filter search results by priority.

* **Test Case Description:** This test case tests the ability to filter search results based on task priority

levels.

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

* **Test Priority:** Medium

* **Preconditions:**

- * A user account is logged in.

- * At least two tasks are created with different priority levels (e.g., "High", "Low").

* **Test Data:**

- * Task 1: Priority = "High"

- * Task 2: Priority = "Low"

* **Test Steps:**

1. Navigate to the task list page.

2. Enter any term in the search bar.

3. Click on the "Priority" filter drop-down.

4. Select "High" priority.

5. Verify that only Task 1 (matching the "High" priority) is displayed in the search results.

* **Postconditions:**

- * The priority filter is reset.

- * The task list returns to the original state.

* **Expected Result:** The search results are accurately filtered to show tasks of the selected priority level.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC-TMS-SEARCH-05

* **Test Case Title:** Verify that users can filter search results by due date.

* **Test Case Description:** This test case verifies the functionality of filtering search results based on a task's due date.

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

* **Test Priority:** Medium

* **Preconditions:**

- * A user account is logged in.

- * At least two tasks are created with different due dates.

* **Test Data:**

- * Task 1: Due date = "2024-03-15"

- * Task 2: Due date = "2024-04-01"

* **Test Steps:**

1. Navigate to the task list page.
2. Click on the "Due Date" filter drop-down.
3. Select "Before" and input "2024-03-10."
4. Verify that only Task 2 (due after the specified date) is displayed in the search results.

* **Postconditions:**

* The due date filter is reset.

* The task list returns to its original state.

* **Expected Result:** The search results are accurately filtered to show tasks whose due date falls within the specified range.

* **Severity:** Minor

* **Priority:** Medium

* **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:

Test Cases for Task Management Tool - Task Collaboration Feature

****User Story:**** 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.

****Test Case 1:****

****Test Case ID:**** TCTM-CC-001

****Test Case Title:**** Verify that users can add comments to tasks.

****Test Case Description:**** This test case verifies the functionality of adding comments to tasks. It will test adding basic comments, adding comments with special characters, and adding long comments.

****Test Suite:**** Task Collaboration

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

****Preconditions:****

- A user is logged into the application.
- A task exists in the system.

****Test Data:****

- Sample comment: "This is a test comment."
- Sample comment with special characters: "This is a comment with !@#\$%^&*()_+."
- Sample long comment: "This is a test comment that is longer than 140 characters to ensure the comment field accommodates longer inputs."

****Test Steps:****

1. Navigate to the task page.
2. Enter the sample comment into the comment field.
3. Click the "Add Comment" button.
4. Verify that the comment is displayed below the task details.
5. Repeat steps 2-4 with the special characters and long comment samples.

****Postconditions:****

- All added comments should be visible to other authorized users.

****Expected Result:****

The comments should be added successfully and displayed on the task page.

****Severity:** Major**

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

****Test Case 2:****

****Test Case ID:** TCTM-CC-002**

****Test Case Title:** Verify that users can attach files to tasks.**

****Test Case Description:**** This test case verifies the functionality of attaching files to tasks. It will test attaching different file types, files of varying sizes, and ensure the attached files are visible to other authorized users.

****Test Suite:** Task Collaboration**

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

****Preconditions:****

- A user is logged into the application.
- A task exists in the system.

****Test Data:****

- Text document (e.g., .txt)
- Image file (e.g., .jpg or .png)
- Spreadsheet (e.g., .xlsx)

****Test Steps:****

1. Navigate to the task page.
2. Click the "Attach File" button.
3. Select each file type from your local storage.
4. Verify that the file is uploaded successfully and displayed in the attached files section.

5. Repeat steps 2-4 with various file sizes (small, medium, large).

****Postconditions:****

- All attached files should be accessible to other authorized users.

****Expected Result:****

Files should be uploaded successfully and displayed on the task page.

****Severity:** Major**

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

****Test Case 3:****

****Test Case ID:** TCTM-CC-003**

****Test Case Title:** Verify that users can share tasks with other users.**

****Test Case Description:**** This test case verifies the functionality of sharing tasks with other users. It will test sharing a task with a valid user, sharing with multiple users, and verifying that shared tasks are accessible to the recipients.

****Test Suite:** Task Collaboration**

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

****Preconditions:****

- A user is logged into the application.
- A task exists in the system.

- A second user account exists (the recipient)

****Test Data:****

- Invalid email address (e.g., "testuserinvalid@email")

****Test Steps:****

1. Navigate to the task page.
2. Click the "Share Task" button.
3. Enter the recipient's valid email address.
4. Click "Share".
5. Verify that the recipient receives a notification about the shared task.
6. Sign in with the recipient's account.
7. Navigate to the shared task.
8. Verify that the recipient can access and view the task details.
9. Repeat steps 3-8 sharing the task with multiple users.
10. Try sharing with an invalid email address and confirm appropriate error message is displayed.

****Postconditions:****

- Shared task is properly accessible to all recipients.

****Expected Result:****

The task should be successfully shared with the recipient(s), and they should have access to the task details.

Invalid email addresses should not result in successful sharing and should display an error message.

****Severity:** Major**

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

****Test Case 4:****

****Test Case ID:** TCTM-CC-004**

****Test Case Title:** Verify that task collaboration features enhance team communication.**

****Test Case Description:** This test case assesses the user experience of collaboration features, focusing on intuitiveness, clarity, and real-time communication aspects.**

****Test Suite:** Task Collaboration**

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

****Preconditions:****

- A team of at least two users are logged into the application.
- A task exists in the system.

****Test Data:****

- Multiple comments related to the task, showcasing effective use of comment threading.

****Test Steps:****

1. Assign the task to two team members.
2. Have one team member leave several related comments on the task, including a comment-topic starter and follow-up comments.
3. Observe how the other team member accesses these comments, the organization of threaded conversation, and the ease of adding replies.

4. Have both team members add attachments (e.g., documents, images) relevant to the task and discussion.

****Postconditions:****

- The team should have successfully used comments and attachments to communicate effectively regarding the task.

****Expected Result:****

Users should find the comment threading and attachment functionality intuitive and helpful for facilitating clear and organized communication about the task.

****Severity:**** Major

****Type of Testing:**** Usability Testing

****Test Case 5:****

****Test Case ID:**** TCTM-CC-005

****Test Case Title:**** Verify that task sharing respects permissions.

****Test Case Description:**** This test case verifies that task sharing adheres to user permissions and access levels. It will ensure that tasks are shared only with authorized users and that unauthorized users cannot access shared tasks.

****Test Suite:**** Task Collaboration

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

****Preconditions:****

- Two user accounts with different permission levels exist.
- A task exists in the system.

****Test Data:****

- One owner user with full access to tasks
- One editor user with limited access to tasks (can view and edit assigned tasks)

****Test Steps:****

1. The owner user shares the task with the editor user.
 2. The editor user logs in and attempts to access the task.
 3. Verify that the editor user can successfully view and edit the task as per their permissions.
 4. Create a third, non-permitted user account (e.g., "reader").
-
5. Attempt to share the task with the reader user.

****Postconditions:****

- The shared task will only be accessible to users with the appropriate permission level.

****Expected Result:****

The editor user, with limited access permissions, will be able to access and edit the shared task. The reader

user, without permissions, will not be able to access the shared task. Any attempt to share with unauthorized users will generate an appropriate error message.

****Severity:**** Major

****Type of Testing:**** Security Testing

Data - 49

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Course Creation Tools

Feature description -

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to

structure content effectively, ensuring an engaging and informative learning experience for students.

Userstory title -

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

Acceptance criteria -

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

Generated Test Cases:

Test Cases for Online Learning Platform - Course Creation Tools

****User Story:**** As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

****Test Case 1:****

*** **Test Case ID:**** TC_CC_01

*** **Test Case Title:**** Verify that instructors can add videos to a course.

*** **Test Case Description:**** This test case verifies the functionality of adding videos to a course by an instructor. It will check if a video can be uploaded successfully, appears within the course, and is playable.

*** **Test Suite:**** Course Creation Tools

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

*** **Preconditions:****

- * A logged-in instructor account

- * Access to a new or existing course

- * A video file in a supported format is available

*** **Test Data:**** A sample video file

****Test Steps:****

1. Navigate to the course being created/edited.
2. locate the video upload section within the course creation tools.
3. Select the sample video file to upload.
4. Verify that the video uploads successfully and displays with appropriate controls.
5. Attempt to play the video and ensure it functions correctly.

****Postconditions:****

- * The uploaded video remains visible within the course

****Expected Result**:** The video should successfully upload and be playable within the course.

****Severity:**** Critical

****Priority:**** High

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

****Test Case 2:****

****Test Case ID:**** TC_CC_02

****Test Case Title:**** Verify that instructors can add presentations to a course.

****Test Case Description:**** This test case verifies the functionality of adding presentations to a course. It will check if a presentation file can be uploaded successfully, appears within the course, and displays correctly.

****Test Suite:**** Course Creation Tools

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

****Preconditions:****

- * A logged-in instructor account
- * Access to a new or existing course
- * A presentation file in a supported format is available (e.g., PowerPoint, PDF)

****Test Data**:** A sample presentation file

****Test Steps:****

1. Navigate to the course being created/edited.
2. Locate the presentation upload section within the course creation tools.
3. Select the sample presentation file to upload.
4. Verify that the presentation uploads successfully and displays with appropriate controls.
5. Navigate through the presentation slides and ensure they load and display correctly.

****Postconditions:****

- * The uploaded presentation remains visible within the course

****Expected Result**:** The presentation should successfully upload and be viewable within the course.

****Severity:**** Critical

****Priority:**** High

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

****Test Case 3:****

****Test Case ID:**** TC_CC_03

****Test Case Title:**** Verify that instructors can add and edit quizzes to a course.

****Test Case Description:**** This test case verifies the functionality of adding and editing quizzes within a course. It will test adding multiple-choice, true/false, and short-answer questions, along with the ability to adjust question order, answer options, and scoring.

****Test Suite:**** Course Creation Tools

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

****Preconditions:****

- * A logged-in instructor account

- * Access to a new or existing course

****Test Data**:** Sample quiz questions, answer options, scoring criteria

****Test Steps:****

1. Navigate to the course being created/edited.
2. Access the quiz creation section within the course creation tools.
3. Add multiple-choice, true/false, and short-answer questions.
4. Specify correct answers, point values, and answer feedback.
5. Reorder questions within the quiz.
6. Edit existing questions and answer options.

* **Postconditions:**

* The newly created or edited quizzes are saved within the course.

* **Expected Result**: Instructors can add and edit different types of questions within a quiz, with the ability to adjust question order, scoring, and feedback.

* **Severity**: Critical

* **Priority**: High

* **Type of Testing**: Functional Testing

Test Case 4:

* **Test Case ID**: TC_CC_04

* **Test Case Title**: Verify that instructors can preview a course before publishing.

* **Test Case Description**: This test case verifies the functionality of the course preview feature. It will ensure that instructors can see how the published course will appear to students.

* **Test Suite**: Course Creation Tools

* **Test Priority**: Medium

* **Preconditions:**

* A logged-in instructor account

* A complete course with content (video, presentation, quiz, assignment)

* Course status is set to "Draft"

* **Test Data**: Content within the course

****Test Steps:****

1. Navigate to the course in "Draft" status.
2. Access the course preview function.
3. Verify that all content is displayed correctly within the preview mode.
4. Navigate through the course structure and ensure navigation functions as expected.

****Postconditions**:** The course remains in "Draft" status.

****Expected Result**:** The preview accurately displays the course content and structure as it will appear to students after publishing.

****Severity:**** Major

****Priority:**** Medium

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

****Test Case 5:****

****Test Case ID:**** TC_CC_05

****Test Case Title:**** Verify that courses can be edited after publishing.

****Test Case Description:**** This test case ensures that instructors can modify their published courses. It will verify the editing capabilities for content, quizzes, and assignments.

****Test Suite:**** Course Creation Tools

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

****Preconditions:****

* A logged-in instructor account

* A published course

****Test Data**:** Sample modifications to course content

****Test Steps:****

1. Navigate to the published course.
2. Access the "Edit" function for the course.

3. Modify existing content within the course (video, presentation, text).
4. Edit existing quizzes and assignments.
5. Save the modifications.

* **Postconditions**:

* The course is updated with the made changes.

* **Expected Result**:

Instructors can successfully edit and update published courses, reflecting the changes in the live version of the course.

* **Severity**:

Major

* **Priority**:

Medium

* **Type of Testing**:

Functional Testing

Data - 50

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Interactive Lessons

Feature description -

Lessons are thoughtfully designed to foster learner engagement through interactive components such as quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these

interactive elements enhance comprehension and retention of course material.

Userstory title -

As a learner, I want interactive lessons so that I can engage with the content more effectively.

Acceptance criteria -

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided after each interactive component.', 'Interactive lessons can be accessed on various devices.']

Generated Test Cases:

Test Cases for Interactive Lessons Feature

Project Context: Online Learning Platform

User Story: As a learner, I want interactive lessons so that I can engage with the content more effectively.

Test Case 1

Test Case ID: ILC-001

Test Case Title: Verify that quizzes are included in lessons.

Test Case Description:

This test case verifies the presence and functionality of quizzes within the interactive lessons.

It checks if learners can access, attempt, and receive results for quizzes.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

* User has access to the platform.

* A lesson containing a quiz is selected.

* **Test Data:** Sample quiz questions and answers.

* **Test Steps:**

1. Navigate to the selected lesson.
2. Identify and click on the quiz section.
3. Attempt all quiz questions.
4. Submit the quiz and review the results.

* **Postconditions:**

- * Quiz results are displayed to the learner.
- * Feedback is provided for each answer (correct/incorrect).

* **Expected Result:** The quiz launches correctly, learners can complete questions, and receive feedback and results.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** ILC-002

* **Test Case Title:** Verify that polls are integrated seamlessly into lessons.

* **Test Case Description:** This test case assesses the seamless integration of polls within the lesson content and tests the functionality of participating in a poll.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:**

- * User has access to the platform.

- * A lesson containing a poll is selected.

* **Test Data:** Sample poll questions with multiple choice options.

* **Test Steps:**

1. Navigate to the selected lesson.
2. Identify and interact with the poll section.
3. Choose an option and submit the poll.
4. Observe the poll results displayed.

* **Postconditions:**

- * Poll results are displayed to the learner.

* **Expected Result:** The poll launches correctly, learners can select options and submit their votes, and the result is displayed.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** ILC-003

* **Test Case Title:** Verify that learners can track their progress through interactive activities.

* **Test Case Description:** This test case verifies the functionality of tracking progress through interactive elements like quizzes, polls, simulations, and discussions.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:**

- * User has access to the platform.

- * A lesson containing various interactive activities is selected.

* **Test Data:** Sample quiz attempts, poll votes, simulation completion, and discussion forum participation.

* **Test Steps:**

1. Navigate to the selected lesson.
2. Complete various interactive activities within the lesson.
3. Access the progress tracking section (e.g., within the lesson itself or the user profile).
4. Verify the displayed progress for each completed activity.

* **Postconditions:**

- * Progress in interactive activities is accurately reflected.

* **Expected Result:** Learners can see their progress through each interactive element within the lesson.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4****

*** **Test Case ID:** ILC-004**

*** **Test Case Title:** Verify that feedback is provided after completing each interactive component**

*** **Test Case Description:** This test case confirms that appropriate feedback is given to the learner after completing each interactive component (quizzes, polls, simulations, discussion posts).**

*** **Test Suite:** Interactive Lessons**

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

*** **Preconditions:****

- * User has access to the platform.**

- * A lesson containing interactive components is selected.**

*** **Test Data:** Sample quiz attempts, poll votes, simulation completion, and discussion forum posts.**

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

- 1. Navigate to the selected lesson**
- 2. Complete an interactive component (e.g., quiz, simulation).**
- 3. Observe the feedback provided for the completed activity.**

*** **Postconditions:****

- * Feedback is clear and concise.**

*** **Expected Result:** Learners receive specific feedback on their performance after each interactive activity.**

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

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

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

* **Test Case Approach:** Positive, Negative (attempting to bypass feedback mechanisms)

Test Case 5

* **Test Case ID:** ILC-005

* **Test Case Title:** Verify that interactive lessons are accessible on various devices.

* **Test Case Description:** This test case ensures the responsiveness and usability of interactive lessons across different devices (desktop, mobile, tablet).

* **Test Suite:** Interactive Lessons

* **Test Priority:** Medium

* **Preconditions:**

- * User has access to the platform.

- * A lesson with interactive activities is selected.

* **Test Data:** Multiple devices with different screen sizes and operating systems.

* **Test Steps:**

1. Access the platform on each designated device.
2. Navigate to the selected lesson.
3. Interact with various interactive elements (quizzes, polls, simulations) on each device.
4. Verify that the functionalities work correctly and the user interface is responsive.

* **Postconditions:**

- * No significant layout issues or functional discrepancies across devices.

* **Expected Result:** Interactive lessons function seamlessly and are user-friendly on different devices.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Compatibility Testing, Usability Testing

* **Test Case Approach:** Positive

Please Note: These test cases provide a starting point. They may need further refinement and expansion based on the specific design and functionalities of the interactive lessons within the Online Learning Platform.

Data - 51

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Personalized Learning Paths

Feature description -

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

Userstory title -

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at

my own pace.

Acceptance criteria -

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

Generated Test Cases:

Test Cases for Personalized Learning Paths Feature

Here are 5 test cases for the "Personalized Learning Paths" feature in the Online Learning Platform:

****Test Case 1****

* **Test Case ID:** OLP-PLP-001

* **Test Case Title:** Verify that learners can choose courses based on their interests.

* **Test Case Description:** This test case verifies the learner's ability to select courses based on their preferred subjects or topics using the filters and search functionalities.

* **Test Suite:** User Stories

* **Test Priority:** High

* **Preconditions:**

- * A learner account is created.

- * The platform has a diverse catalog of courses across various subjects.

- * Filters and search functionalities are accessible to learners.

* **Test Data:** List of courses categorized by subject, keywords for specific topics of interest.

* **Test Steps:**

1. Log in to the platform as a learner.

2. Navigate to the course catalog.
3. Apply filters based on specific subjects or topics of interest.
4. Use the search functionality to find courses related to specific keywords.
5. Verify that the displayed courses align with the selected filters and search query.

* **Postconditions:** No specific postconditions required.

* **Expected Result:** The learner should be able to successfully filter and search for courses based on their interests, displaying relevant course offerings.

* **Severity:** Critical

* **Priority:** High

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

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** OLP-PLP-002

* **Test Case Title:** Verify that personalized learning paths adjust dynamically based on learner progress.

* **Test Case Description:** This test case assesses the platform's ability to modify the recommended learning path based on the learner's completed modules and courses.

* **Test Suite:** User Stories

* **Test Priority:** High

* **Preconditions:**

* A learner account is created and logged in.

* **Test Data:** A predetermined set of courses and modules with specific prerequisite relationships.

* **Test Steps:**

1. Start a personalized learning path with a chosen set of introductory courses.
2. Complete several modules within the chosen path.
3. Observe the adjustments made to the recommended learning path based on the completed modules.

4. Verify that new recommendations align with the learner's progress and knowledge acquired.

* **Postconditions:** No specific postconditions required.

* **Expected Result:** The platform should dynamically adjust the recommended learning path, suggesting relevant courses and modules based on the learner's completed progress.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** OLP-PLP-003

* **Test Case Title:** Verify that learners receive relevant recommendations for new courses and modules.

* **Test Case Description:** This test case evaluates the accuracy and relevance of course and module recommendations provided to learners based on their learning history and interests.

* **Test Suite:** User Stories

* **Test Priority:** High

* **Preconditions:**

- * A learner account is created and logged in.

- * The learner has interacted with specific courses and modules.

* **Test Data:** A predefined set of courses and modules, categorized by subject and topic, to simulate learner interests and history.

* **Test Steps:**

1. Log in as a learner.
2. Observe the initial recommendations presented on the learning path dashboard.
3. Intentionally select and complete a few courses/modules from a specific subject area.
4. Refresh the dashboard to observe updated recommendations.

5. Verify that the new recommendations are relevant to the learner's recently explored subject and demonstrate a personalized approach.

* **Postconditions:** No specific postconditions required.

* **Expected Result:** The platform should provide tailored recommendations for new courses and modules aligned with the learner's progressively explored interests and learning trajectory.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive, Positive/Negative

****Test Case 4****

* **Test Case ID:** OLP-PLP-004

* **Test Case Title:** Verify that progress tracking is personalized to each learner's path.

* **Test Case Description:** This test case assesses the accuracy and uniqueness of progress tracking for each learner.

* **Test Suite:** User Stories

* **Test Priority:** Medium

* **Preconditions:**

- * Multiple learner accounts are created.

- * Each learner starts a personalized learning path.

* **Test Data:** Courses/Modules with varying complexity and assignments.

* **Test Steps:**

1. Have each learner complete different sets of modules within their chosen path.

2. Log in as each learner individually.

3. Review the personalized progress tracking dashboard.

4. Verify that the progress bars and completion status reflect the specific modules completed by each

learner.

* **Postconditions:** No specific postconditions required.

* **Expected Result:** Each learner's progress tracking dashboard should uniquely display their individual course and module completions, accurately reflecting their personalized learning journey.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** OLP-PLP-005

* **Test Case Title:** Verify that learners can adjust their learning paths at any time.

* **Test Case Description:** This test case examines the learner's ability to modify their chosen learning path, add or remove courses and modules as needed.

* **Test Suite:** User Stories

* **Test Priority:** Medium

* **Preconditions:**

- * A learner account is created and logged in.

- * A personalized learning path has been initialized.

* **Test Data:**

- * List of courses and modules within the existing learning path.

- * Additional courses and modules from different subject areas.

* **Test Steps:**

1. View the learner's personalized learning path dashboard.

2. Use the interface to add new courses or modules from different subject areas.

3. Utilize the interface to remove previously selected modules from the path.

4. Verify that the changes are reflected in the revised learning path.

* **Postconditions:** No specific postconditions required.

* **Expected Result:** Learners should be able to freely adjust their learning paths by adding or removing courses and modules as needed, enabling flexibility and customization.

* **Severity:** Minor

* **Priority:** Medium

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

* **Test Case Approach:** Positive

Data - 52

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Progress Tracking

Feature description -

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

Userstory title -

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Acceptance criteria -

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

Generated Test Cases:

Test Cases for Online Learning Platform: Progress Tracking Feature

****Project:**** Online Learning Platform

****Feature:**** Progress Tracking

****User Story:**** As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Test Case ID: TC_PT_001

****Test Case Title:**** Verify that completed activities are accurately reflected in the progress tracker.

****Test Case Description:**** This test case aims to confirm that the progress tracker correctly displays the status of completed activities within a course, including lectures, quizzes, and assignments.

****Test Suite:**** User Interface Testing

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

****Preconditions:****

- * A user is logged into the platform.

- * A course with multiple activities (lectures, quizzes, assignments) is available to the user.

* The user has completed a set of activities within the course.

****Test Data:****

* Course ID: [Course ID]

* User login credentials: [Valid Username & Password]

****Test Steps:****

1. Navigate to the course dashboard.
2. Verify that the progress tracker accurately displays the completed status of each activity.
3. Click on individual activities to confirm their completion status and details.
4. Compare the progress tracker with the activity completion records within the course content.
5. Repeat steps 2-4 for different types of activities (lectures, quizzes, assignments).

****Postconditions:****

* The user remains logged in.

****Expected Result:****

All completed activities within the course are accurately reflected as completed in the progress tracker.

****Severity:** Critical**

****Priority:** High**

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

Test Case ID: TC_PT_002

****Test Case Title:** Verify that grades are correctly displayed in the progress tracker.**

****Test Case Description:**** This test case verifies that the progress tracker accurately reflects the grades earned by the user on quizzes and assignments within a course.

****Test Suite:**** User Interface Testing

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

****Preconditions:****

- * A user is logged into the platform.
- * A course with quizzes and assignments is available to the user.
- * The user has completed quizzes and assignments and received grades.

****Test Data:****

- * Course ID: [Course ID]
- * User login credentials: [Valid Username & Password]
- * Quiz and Assignment scores: [Record of completed quizzes and assignment scores]

****Test Steps:****

1. Navigate to the course dashboard.
2. Review the progress tracker for recorded grades on completed quizzes and assignments.
3. Click on individual quizzes and assignments to verify the displayed grades against the expected scores.
4. Compare the progress tracker grades with the gradebook within the course content.

****Postconditions:****

- * The user remains logged in.

****Expected Result:****

All grades earned on quizzes and assignments are displayed accurately in the progress tracker.

****Severity:** Critical**

****Priority:** High**

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

Test Case ID: TC_PT_003

****Test Case Title:** Verify that achievements are highlighted in the progress tracker.**

****Test Case Description:** This test case verifies that the progress tracker visually highlights achievements earned by the user within a course or module.**

****Test Suite:** User Interface Testing**

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

****Preconditions:****

- * A user is logged into the platform.
- * A course with achievements embedded within modules or activities is available to the user.
- * The user has earned at least one achievement within the course.

****Test Data:****

- * Course ID: [Course ID]
- * User login credentials: [Valid Username & Password]
- * Achievement criteria: [List of achieved achievements and criteria]

****Test Steps:****

1. Navigate to the course dashboard.
2. Review the progress tracker for visual cues indicating achieved achievements.

3. Click on highlighted achievements to view details and confirmation of completion criteria.
4. Check if the visual cue (e.g., badge, icon, color) distinguishes achieved achievements from pending or unattained ones.

****Postconditions:****

- * The user remains logged in.

****Expected Result:****

Achieved achievements are clearly highlighted in the progress tracker with distinct visual cues.

****Severity:** Major**

****Priority:** Medium**

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

Test Case ID: TC_PT_004

****Test Case Title:** Verify that learners receive notifications about progress updates.**

****Test Case Description:** This test case verifies that the platform sends notifications to learners about their progress updates, such as completing activities, earning grades, or achieving milestones.**

****Test Suite:** System Functionality Testing**

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

****Preconditions:****

- * A user is logged into the platform.
- * The user has enabled notifications within their account settings.

* A course with activities and assignments is available to the user.

****Test Data:****

* User login credentials: [Valid Username & Password]

* Notification settings: Enabled for progress updates

****Test Steps:****

1. Complete a learning activity in the specified course (e.g., watching a lecture, taking a quiz).
2. Verify that a notification is received regarding the completion of the activity.
3. Earn a grade on an assignment within the course.
4. Confirm the receipt of a notification about the earned grade.
5. Achieve a milestone within the course (e.g., completing a module).
6. Verify that a notification is received regarding the milestone achievement.

****Postconditions:****

* The user remains logged in.

* Notifications are properly displayed within the platform's notification center.

****Expected Result:****

Learners receive timely and relevant notifications about their progress updates, including completion of activities, grades earned, and milestone achievements.

****Severity:** Critical**

****Priority:** High**

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

Test Case ID: TC_PT_005

****Test Case Title:**** Verify the ability to export progress reports.

****Test Case Description:**** This test case verifies the functionality to export progress tracking data as a report in a usable format (e.g., PDF, CSV).

****Test Suite:**** User Interface Testing

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

****Preconditions:****

* A user is logged into the platform.

* The user has completed activities and earned grades within at least one course.

****Test Data:****

* Course ID: [Course ID]

* User login credentials: [Valid Username & Password]

****Test Steps:****

1. Navigate to the progress tracking section of the user dashboard.
2. Locate the option to export the progress report.
3. Select the desired export format (e.g., PDF, CSV).
4. Download the exported report to the user's device.

5. Open the downloaded report and verify that it contains the expected data:

- * Course name
- * User name
- * Completed activities
- * Grades earned
- * Date of completion

****Postconditions:****

- * The user remains logged in.

****Expected Result:****

A formatted progress report (according to the selected format) containing the user's progress data is successfully downloaded.

****Severity:** Major**

****Priority:** Medium**

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

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:

Online Learning Platform: Assessment & Grading Test Cases

****User Story:**** As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

****Test Case ID:**** TC_Assessment_01

****Test Case Title:**** Verify that instructors can create quizzes.

****Test Case Description:**** This test case verifies the functionality of creating quizzes with multiple choice, true/false, and short answer question types.

****Test Suite:**** Assessment Module

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

****Preconditions:****

* Instructor account with administrative privileges is logged in.

****Test Data:****

* Course content related to a specific topic.

* Multiple choice, true/false, and short answer question content.

****Test Steps:****

1. Navigate to the "Create Assessment" page within the course.
2. Select "Quiz" as the assessment type.
3. Add multiple choice, true/false, and short answer questions with relevant content and scoring criteria.
4. Set the quiz duration and availability.
5. Publish the quiz.

****Postconditions:****

- * A new quiz is created and visible to students.
- * Quiz details, question types, and scoring criteria are accurate.
- * Instructors can access the quiz for editing or deleting.

****Expected Result:**** The quiz is successfully created and meets all the specified criteria.

****Severity:**** Critical

****Priority:** High**

****Type of Testing:** Functional Testing, User Acceptance Testing**

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

****Test Case ID:** TC_Assessment_02**

****Test Case Title:** Verify that instructors can create tests.**

****Test Case Description:** This test case verifies the functionality of creating tests with similar question types and features as quizzes.**

****Test Suite:** Assessment Module**

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

****Preconditions:****

* Instructor account with administrative privileges is logged in.

****Test Data:****

* Course content related to a specific topic.

* Multiple choice, true/false, and short answer question content.

****Test Steps:****

1. Navigate to the "Create Assessment" page within the course.
2. Select "Test" as the assessment type.
3. Add multiple choice, true/false, and short answer questions with relevant content and scoring criteria.
4. Set the test duration and availability.
5. Publish the test.

****Postconditions:****

- * A new test is created and visible to students.
- * Test details, question types, and scoring criteria are accurate.
- * Instructors can access the test for editing or deleting.

****Expected Result:**** The test is successfully created and meets all the specified criteria.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

****Test Case ID:**** TC_Assessment_03

****Test Case Title:**** Verify that instructors can create assignments.

****Test Case Description:**** This test case verifies the functionality of creating assignments that allow for subjective responses.

****Test Suite:**** Assessment Module

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

****Preconditions:****

- * Instructor account with administrative privileges is logged in.

****Test Data:****

- * Course content related to a specific topic.
- * Essay-type question content with clear prompt and grading criteria.

****Test Steps:****

1. Navigate to the "Create Assessment" page within the course.

2. Select "Assignment" as the assessment type.
3. Add an essay-type question with clear instructions and grading criteria.
4. Set the assignment deadline and submission guidelines.
5. Publish the assignment.

****Postconditions:****

- * A new assignment is created and visible to students.
- * Assignment details, instructions, and grading criteria are accurate.
- * Instructors can access submitted assignments for review and grading.

****Expected Result:**** The assignment is successfully created and meets all the specified criteria.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

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

****Test Case ID:**** TC_Assessment_04

****Test Case Title:**** Verify automatic grading for objective questions.

****Test Case Description:**** This test case verifies that objective questions, such as multiple-choice and true/false, are automatically graded.

****Test Suite:**** Assessment Module

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

****Preconditions:****

- * Instructor account with administrative privileges is logged in.
- * A quiz with multiple-choice and true/false questions has been created.

****Test Data:****

- * Correct answers for each multiple-choice and true/false question.

****Test Steps:****

1. Launch the pre-created quiz.
2. Answer all multiple-choice and true/false questions with the correct answers.
3. Submit the quiz.
4. Review the grade report.

****Postconditions:****

- * The quiz submission is recorded.
- * An accurate grade is displayed based on the correct answers.

****Expected Result:**** The quiz is graded automatically with accurate results for objective questions.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Unit Testing

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

****Test Case ID:**** TC_Assessment_05

****Test Case Title:**** Verify manual grading for subjective questions.

****Test Case Description:**** This test case verifies that instructors can manually grade subjective questions, such as essay type.

****Test Suite:**** Assessment Module

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

****Preconditions:****

- * Instructor account with administrative privileges is logged in.
- * An assignment with essay-type questions has been created and submissions have been made.

****Test Data:****

- * Sample essay responses for grading.

****Test Steps:****

1. Access the submitted assignments from the assignment page.
2. Review an essay submission.
3. Manually grade the essay based on the predefined criteria.
4. Provide feedback to the student.
5. Save the graded submission.

****Postconditions:****

- * The essay submission is marked as graded.
- * Feedback is provided to the student.
- * The instructor's grading is recorded in the system.

****Expected Result:**** The instructor can successfully manually grade the subjective questions and provide feedback.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, User Acceptance 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:

Test Cases for Online Learning Platform - Discussion Forums Feature

Here are 5 detailed test cases for the "Discussion Forums" feature of the Online Learning Platform, based on the provided user story and acceptance criteria:

****Test Case 1:**** Verify that learners can post questions and responses in forums.

*** **Test Case Description:**** This test case verifies the functionality of posting questions and responses within the discussion forum. It will involve logging in as a learner, navigating to a forum, composing a new post, and ensuring it is successfully displayed within the forum.

*** **Test Suite:**** User Interface Testing

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

*** **Preconditions:****

- * A User account with learner privileges is created.

- * A relevant course with an active discussion forum is accessible.

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

- * Sample question and response text

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

1. Log in as a learner.
2. Navigate to the course with the discussion forum.
3. Click on the "New Post" button.
4. Compose a question or response in the designated text field.
5. Select "Post" or equivalent button to submit the post.
6. Verify that the posted question or response is displayed in the forum.

*** **Postconditions:****

- * No changes expected after test completion.

*** **Expected Result:**** The learner will be able to successfully post a question or response within the designated forum.

* **Severity:** High

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2: Verify that forums support threaded discussions.

* **Test Case Description:** This test case verifies the functionality of threaded discussions, allowing for replies and sub-replies within a conversation.

* **Test Suite:** User Interface Testing

* **Test Priority:** High

* **Preconditions:**

- * A User account with learner privileges is created.

- * A relevant course with an active discussion forum is accessible.

- * A sample question has been posted in the forum.

* **Test Data:**

- * Sample reply text for the initial question.

- * Further sub-reply text to create a thread.

* **Test Steps:**

1. Log in as a learner.

2. Navigate to the course with the discussion forum.

3. Locate the sample question previously posted.

4. Reply to the sample question.

5. Submit the reply.

6. Post another reply to the initial reply, creating a sub-thread.

* **Postconditions:**

- * No changes expected after test completion.

* **Expected Result:** The replies and sub-replies will form a clear threaded conversation, allowing for structured discussion.

* **Severity:** High

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3: Verify that instructors can moderate discussions.

* **Test Case Description:** This test case verifies the instructor's ability to moderate discussions by deleting posts, editing posts, and marking posts as solutions.

* **Test Suite:** User Interface Testing/Administrative Testing

* **Test Priority:** High

* **Preconditions:**

- * An instructor account is created with appropriate permissions.

- * A relevant course with an active discussion forum is accessible.

- * At least two learner posts are present in the forum.

* **Test Data:**

- * Specific learners posts to analyze moderation features.

* **Test Steps:**

1. Log in as an instructor.
2. Navigate to the course with the discussion forum.
3. Select a learner post to check deletion functionality.
4. Access the moderation options for the chosen post.
5. Delete the selected post.
6. Verify the post is removed from the forum.
7. Repeat steps 3-6 for editing a post.

8. Repeat steps 3-6 for marking a post as a solution.

* **Postconditions:**

- * Deleted post should no longer be visible in the forum.

- * Edited post reflects the changes made by the instructor.

- * Marked post should be explicitly indicated as a solution.

* **Expected Result:** The instructor will be able to successfully delete, edit, and mark posts as solutions within the forum.

* **Severity:** High

* **Priority:** High

* **Type of Testing:** Functional Testing/Authentication Testing

* **Test Case Approach:** Positive

Test Case 4: Verify that learners receive notifications for new posts.

* **Test Case Description:** This test case verifies the functionality of notifications for new posts in a discussion forum.

* **Test Suite:** User Interface Testing/Notification Testing

* **Test Priority:** High

* **Preconditions:**

- * A learner account with notification preferences enabled is created.

- * A relevant course with an active discussion forum is accessible.

- * A post has already been made in the forum.

* **Test Data:**

* A simulated notification triggering event (a new post being made).

* **Test Steps:**

1. Log in as a learner.
2. Navigate to the course with the discussion forum.
3. Initiate the simulated notification (e.g., another user posts a new message).
4. Verify a notification message is received, indicating a new post.
5. Click on the notification.
6. Verify the redirect to the new post within the forum.

* **Postconditions:**

* No changes expected after test completion.

* **Expected Result:** The learner will receive a notification and be able to access the new post with ease.

* **Severity:** High

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5: Verify that forums are accessible on all devices.

* **Test Case Description:** This test case verifies the responsiveness and accessibility of discussion forums across different devices (desktop, mobile, tablet).

* **Test Suite:** Cross-Browser Testing/Cross-Device Testing

* **Test Priority:** Medium

* **Preconditions:**

* Access to different devices (desktop, mobile, tablet) with varying screen sizes. Each device must have an

active internet connection.

****Test Data:****

- * The user account and login credentials used in previous test cases.

****Test Steps:****

1. Access the online learning platform through the web browser on each device.
2. Log in as a learner.
3. Navigate to the course with an active discussion forum on each device.
4. Verify that the forum is displayed correctly and all functionalities (posting, replying, viewing threads) are accessible.
5. Test scrolling through long threads, accessing profile pictures, and navigating between different sections of the forum on each device.

****Postconditions:****

- * No changes expected after test completion.

****Expected Result:**** Learners should be able to access and utilize the forum features seamlessly across all tested devices, with a consistent user experience.

****Severity:**** Medium

****Priority:**** High

****Type of Testing:**** Cross-Browser Testing/Cross-Device 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 -

Live Classes and Webinars

Feature description -

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

Userstory title -

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

Acceptance criteria -

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

Generated Test Cases:

Test Cases for Live Classes and Webinars Feature

****Project Context:**** Online Learning Platform

****User Story Title:**** As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

****Test Case 1:****

*** **Test Case ID:** LC1**

*** **Test Case Title:** Verify that learners can join a live class using video conferencing tools.**

*** **Test Case Description:**** This test case verifies the functionality of joining a live class through integrated video conferencing tools. It ensures learners can access the session, view the instructor, and participate in the real-time audio/video interaction.

*** **Test Suite:** Live Class Functionality**

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

*** **Preconditions:****

- * A scheduled live class is available.

- * The learner has an active account on the platform.

- * Video conferencing software is installed and integrated with the platform.

*** **Test Data:**** Class schedule details, learner credentials

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

1. Log in to the platform using valid learner credentials.
2. Navigate to the "Live Classes" section.
3. Select the scheduled live class from the list.
4. Click the "Join Class" button to launch the video conferencing session.
5. Verify successful connection to the video conference and ability to see and hear the instructor.

*** **Postconditions:**** Exit the video conferencing session.

*** **Expected Result:**** The learner successfully connects to the live class using the integrated video conferencing tools and can view and hear the instructor.

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

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

*** **Type of Testing:**** Functional Testing, System Testing

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

****Test Case 2:****

*** **Test Case ID:** LC2**

*** **Test Case Title:**** Verify that learners can ask questions during a live class.

*** **Test Case Description:**** This test case ensures learners can submit questions in real-time during a live class and that these questions are received by the instructor.

*** **Test Suite:**** Live Class Functionality

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

*** **Preconditions:****

- * A scheduled live class is in progress.

- * The learner has an active account on the platform.

*** **Test Data:**** Question text

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

1. Join a live class using the video conferencing tools (see Test Case LC1).
2. Type a question relevant to the class content in the designated chat or Q&A section.
3. Submit the question.

*** **Postconditions:**** No specific follow-up action required.

*** **Expected Result:**** The question submitted by the learner is successfully sent to the instructor and is visible within the designated chat or Q&A section.

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

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

*** **Type of Testing:**** Functional Testing, System Testing

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

****Test Case 3:****

*** **Test Case ID:**** LC3

*** **Test Case Title:**** Verify that session recordings are available for later viewing.

*** **Test Case Description:**** This test case ensures that recordings of live sessions are generated and

accessible to learners after the session ends.

* **Test Suite:** Live Class Accessibility

* **Test Priority:** Medium

* **Preconditions:**

- * A live class has been conducted.

- * The "Record Session" option is enabled in the platform settings.

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

* **Test Steps:**

1. Navigate to the "Archived Live Classes" section on the platform.
2. Locate the recording of the previously conducted live class.
3. Click the "Play" button to initiate playback of the recording.

* **Postconditions:** Close the playback window.

* **Expected Result:** The recorded live class video is accessible and playable. The content displayed is consistent with the live session.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** LC4

* **Test Case Title:** Verify that instructors can share screen and resources during a live session.

* **Test Case Description:** This test case verifies the functionality allowing instructors to share their screen and various resources with learners during a live interactive session.

* **Test Suite:** Live Class Functionality

* **Test Priority:** Medium

* **Preconditions:**

* A scheduled live class is in progress.

* An instructor account is logged in to the platform.

* **Test Data:** Instructor can prepare slides, a document or a webpage to share.

* **Test Steps:**

1. Join the live class using the instructor account (see Test Case LC1).

2. Locate the screen sharing or resource sharing option within the video conferencing interface.

3. Select the desired resource (e.g., document, slides, webpage) to share.

* **Postconditions:** Stop screen sharing or resource sharing.

* **Expected Result:** The instructor successfully shares the chosen resource with all learners in the live session. Learners can clearly see and interact with the shared content.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** LC5

* **Test Case Title:** Verify that notifications are sent before a live class starts. *

* **Test Case Description:** This test case ensures learners receive timely notifications about upcoming live classes.

* **Test Suite:** Live Class Notifications

* **Test Priority:** Medium

* **Preconditions:**

* A scheduled live class is registered on the platform.

* Learners have enabled notifications in their account settings.

* **Test Data:** User account with notification preferences enabled

* **Test Steps:**

1. Log in to the platform using a test learner account.
2. Navigate to the "Upcoming Live Classes" section (or any similar relevant area).
3. View the notification settings within the individual learner profile.
4. Verify notifications are enabled for the specific event type "Live Classes" or "Webinars".
5. Wait for the scheduled time before the live class begins.
6. Check if a notification is received on the chosen notification platform (email, push notification, etc.)

* **Postconditions:** Dismiss the notification.

* **Expected Result:** A notification is successfully sent to the learner's designated notification platform before the live class start time.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive

Data - 56

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Mobile Accessibility

Feature description -

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

Userstory title -

As a learner, I want mobile access to the platform so that I can learn on the go.

Acceptance criteria -

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

Generated Test Cases:

Test Cases for Mobile Accessibility Feature

****Test Case ID:**** TC-01

****Test Case Title:**** Verify that the platform is accessible on smartphones

****Test Case Description:**** This test case verifies the platform's responsiveness and functionality on a smartphone device, ensuring learners can access and interact with its core features.

****Test Suite:**** Mobile Accessibility

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

****Preconditions:****

- A smartphone device with a stable internet connection.
- The platform application installed on the smartphone.

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

****Test Steps:****

1. Launch the platform application on the smartphone.
2. Navigate to the home screen.
3. Access different sections like course catalog, profile, forum, and notifications.
4. Attempt to enroll in a course and complete an introductory quiz.
5. Download a course resource and verify its accessibility offline.

****Postconditions:****

- Close all platform application windows and exit.

****Expected Result:**** The platform should load and function flawlessly on the smartphone, displaying all content and features correctly with optimized touch navigation.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** System Testing, Functional Testing, Compatibility Testing**

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

****Test Case ID:** TC-02**

****Test Case Title:** Verify that the mobile UI is optimized for touch navigation**

****Test Case Description:**** This test cases verifies the usability and intuitiveness of the platform's interface on a touch screen device.

****Test Suite:** Mobile Accessibility**

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

****Preconditions:****

- A tablet device with a stable internet connection.
- The platform application installed on the tablet.

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

****Test Steps:****

1. Launch the platform application on the tablet.
2. Navigate through menus and sub-menus using touch gestures.
3. Tap on interactive elements like buttons, links, and icons.
4. Scroll through course content and resource lists.
5. Zoom in and out on text and images.

****Postconditions:****

- Close all platform application windows and exit.

****Expected Result:** All touch interactions should be responsive, smooth, and intuitive. The UI should adapt seamlessly to touch input, providing a user-friendly experience.**

****Severity:** Critical**

****Priority:** High**

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

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

****Test Case ID:** TC-03**

****Test Case Title:** Verify that course materials are accessible and readable on mobile devices**

****Test Case Description:** This test case ensures that all course content is formatted and displayed appropriately for mobile screens, enhancing readability and learning experience.**

****Test Suite:** Mobile Accessibility**

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

****Preconditions:****

- A smartphone device with a stable internet connection.
- The platform application installed on the smartphone.

****Test Data:**** Access a course with diverse content types (text, images, videos, interactive elements).

****Test Steps:****

1. Access the chosen course on the smartphone.
2. Read through course text materials, verifying font size, line spacing, and paragraph breaks.
3. View downloadable resources like PDFs, documents, and presentations.
4. Watch video lectures, checking for proper playback and resolution.
5. Interact with interactive elements like quizzes and simulations.

****Postconditions:****

- Close all platform application windows and exit.

****Expected Result:**** All course materials should be legible and accessible on the mobile device screen, with appropriate formatting and functionality for all content types.

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Functional Testing, Compatibility Testing**

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

****Test Case ID:** TC-04**

****Test Case Title:** Verify that notifications are sent to mobile devices**

****Test Case Description:**** This test case ensures the platform delivers timely and relevant notifications to users on their mobile devices.

****Test Suite:** Mobile Accessibility**

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

****Preconditions:****

- A smartphone device with a stable internet connection and cellular data or WiFi enabled notification settings.

- The platform application installed on the smartphone.

- A user account created on the platform.

****Test Data:**** User account details and notification settings preferences.

****Test Steps:****

1. Log into the platform application on the smartphone.

2. Enable different types of notifications (e.g., course updates, messages, activity notifications).

3. Trigger various activities within the platform that generate notifications (e.g., enroll in a new course, receive a private message, complete a quiz).

4. Verify that the respective notifications appear on the smartphone device within a reasonable timeframe.

5. Check the notification content for accuracy and clarity.

****Postconditions:****

- Disable all notifications in the application settings.

****Expected Result:**** The platform should successfully send notifications to the mobile device for the triggered activities, displaying accurate and relevant information.

****Severity:** Medium**

****Priority:** Medium**

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

****Test Case Approach:** Positive/Negative**

****Test Case ID:** TC-05**

****Test Case Title:**** Verify that the platform supports offline access to downloaded materials

****Test Case Description:**** This test case ensures the platform enables users to access downloaded course materials offline, allowing for continued learning regardless of connectivity.

****Test Suite:**** Mobile Accessibility

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

****Preconditions:****

- A smartphone device with a stable internet connection.
- The platform application installed on the smartphone.
- An active user account on the platform.

****Test Data:**** A course containing text materials, downloadable documents, or presentations.

****Test Steps:****

1. Access the chosen course on the smartphone and download selected materials (e.g., PDF document, PowerPoint presentation).
2. Turn off the internet connection on the smartphone.
3. Attempt to open the downloaded materials through the platform application.
4. Verify that the downloaded materials are accessible and viewable offline.
5. Attempt to access non-downloaded course materials, observing that the platform prompts a need for an internet connection.

****Postconditions:**** Restoring internet connection on the device.

****Expected Result:**** The platform should successfully allow users to access and view downloaded materials offline. The platform should also clearly indicate which content requires an internet connection.

****Severity:**** Medium

****Priority:**** Medium

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

****Test Case Approach:**** Positive/Negative

Data - 57

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Resource Library

Feature description -

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides

enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

Userstory title -

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

Acceptance criteria -

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

Generated Test Cases:

Online Learning Platform - Resource Library Test Cases

****User Story:**** As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

****Test Case ID:**** RL-001

****Test Case Title:**** Verify that the resource library includes e-books, articles, and reference guides.

****Test Case Description:**** This test case verifies the availability of diverse resource types within the resource library as specified in the acceptance criteria.

****Test Suite:**** Resource Library

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

****Preconditions:****

- * User is logged into the online learning platform as a learner.
- * The platform is connected to the internet.

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

****Test Steps:****

1. Navigate to the "Resource Library" section of the platform.
2. Verify that the library displays a mix of e-books, articles, and reference guides.
3. Select and open a sample resource from each category to confirm its accessibility.

****Postconditions:****

- * No changes made to the platform.
- * User remains logged in.

****Expected Result:**** The resource library should display a variety of e-books, articles, and reference guides, accessible to the learner.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing (UAT)

****Test Case ID:**** RL-002

****Test Case Title:**** Verify that learners can search for specific materials in the resource library.

****Test Case Description:**** This test case validates the functionality of the search feature within the resource library, ensuring learners can find specific materials.

****Test Suite:**** Resource Library

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

****Preconditions:****

- * User is logged into the online learning platform as a learner.
- * The platform is connected to the internet.
- * At least 10 resources exist in the library.

****Test Data:**** Search terms related to specific subjects or topics covered by the platform's courses.

****Test Steps:****

1. Navigate to the "Resource Library" section.
2. Enter a relevant search term in the search bar.
3. Click the "Search" button.
4. Verify that the search results display a list of relevant resources matching the search term.
5. Attempt to search for a non-existent term and verify an appropriate "No results found" message is displayed.

****Postconditions:****

- * No changes made to the platform.

* User remains logged in.

****Expected Result:**** The search functionality should return accurate and relevant resource results based on the entered search terms.

****Test Case ID:**** RL-003

****Test Case Title:**** Verify that resources are categorized and tagged for easy navigation.

****Test Case Description:**** This test case evaluates the organization and searchability of resources within the library through categorization and tagging.

****Test Suite:**** Resource Library

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

****Preconditions:****

* User is logged into the online learning platform as a learner.

* The platform is connected to the internet.

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

****Test Steps:****

1. Navigate to the "Resource Library" section.
2. Identify the available resource categories (e.g., by subject, type, difficulty level).
3. Select a category and verify a list of resources within that category displays.

4. Explore various resource tags and verify that clicking on a tag displays a filtered list of resources matching that tag.

****Postconditions:****

* No changes made to the platform.

* User remains logged in.

****Expected Result:**** Resources should be categorized and tagged effectively, allowing learners to easily find and browse relevant materials.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing (UAT)

****Test Case ID:**** RL-004

****Test Case Title:**** Verify that learners can download resources for offline access.

****Test Case Description:**** This test case confirms the functionality of downloading resources and their availability for offline use.

****Test Suite:**** Resource Library

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

****Preconditions:****

* User is logged into the online learning platform as a learner.

* The platform is connected to the internet.

****Test Data:**** A downloadable resource from the resource library (e.g., a PDF file)

****Test Steps:****

1. Navigate to the "Resource Library" section.
2. Locate a downloadable resource (e.g., an e-book or article in PDF format).
3. Click the download button associated with the resource.
4. Verify that the resource is downloaded successfully to the user's device.
5. Offline, attempt to open the downloaded resource and confirm its accessibility.

****Postconditions:****

* The downloaded resource should be accessible offline.

* No changes made to the user's account.

****Expected Result:**** Learners should be able to successfully download resources for offline access.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing (UAT)

****Test Case ID:**** RL-005

****Test Case Title:**** Verify that instructors can recommend resources to learners.

****Test Case Description:**** This test case validates the functionality for instructors to recommend resources to

learners within the platform.

****Test Suite:**** Resource Library

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

******(This may require backend access and administrative roles for testing purposes.)******

****Preconditions:****

- * Instructor account is logged into the platform.
- * One or more resources are already uploaded to the resource library.

****Test Data:**** A list of resource links or details that instructors can use for recommending.

****Test Steps:****

1. Instructor navigates to the "Resource Library" section.
2. Select a specific resource to recommend.
3. Utilize the "Recommend to Learners" or similar functionality to add the resource to a recommendation list.
4. From the learner's perspective, verify that the recommended resource is visible in a designated section (e.g., "Recommended Resources", "Instructor Suggestions").
5. Access the recommended resource and confirm its availability.

****Postconditions:****

- * No changes to the learner's account.

* No changes made to the user accounts used for the registration process.

****Expected Result:**** Instructors should be able to recommend specific resources to learners, providing personalized learning suggestions within the platform.

Data - 58

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Peer Review

Feature description -

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

Userstory title -

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

Acceptance criteria -

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers

remain anonymous to ensure unbiased feedback.']

Generated Test Cases:

Here are 5 detailed test cases for the "Peer Review" feature of the Online Learning Platform, following the provided template:

Test Case 1:

* **Test Case ID:** TC_PR_001

* **Test Case Title:** Verify that learners can submit assignments for peer review.

* **Test Case Description:** This test case verifies that a learner can successfully upload an assignment for peer review as intended within the specified course.

* **Test Suite:** User Story Feature Testing - Peer Review

* **Test Priority:** High

* **Preconditions:**

- * A learner is enrolled in a course offering peer review on assignments.

- * An assignment is available for submission within the course.

- * The learner has a valid assignment file (e.g., document, presentation).

* **Test Data:** A sample assignment document relevant to the course.

* **Test Steps:**

1. Login as a learner enrolled in the course.
2. Navigate to the relevant course module containing the peer review assignment.
3. Click on the "Submit for Peer Review" button.
4. Select the assignment file from their device and upload it.
5. Submit the assignment for review.

* **Postconditions:**

- * The assignment is visible in the learner's submitted work area.

- * The assignment appears in the designated peer review queue for reviewers.
- * **Expected Result:** The learner successfully submits the assignment for peer review without any errors.
- * **Severity:** Critical
- * **Priority:** High
- * **Type of Testing:** System Testing, Functional Testing
- * **Test Case Approach:** Positive

Test Case 2:

- * **Test Case ID:** TC_PR_002
- * **Test Case Title:** Verify that peers can provide feedback on submitted assignments.
- * **Test Case Description:** This test case verifies that a peer reviewer can access a submitted assignment and provide feedback according to the defined structure.
- * **Test Suite:** User Story Feature Testing - Peer Review
- * **Test Priority:** High
- * **Preconditions:**
 - * An assignment has been successfully submitted by a learner for peer review.
 - * A peer reviewer is assigned the task to review the assignment.
- * **Test Data:** A sample feedback comment following the platform's guidance on constructive feedback.
- * **Test Steps:**
 1. Login as a peer reviewer.
 2. Navigate to the peer review queue or dashboard.
 3. Select the assigned assignment for review.
 4. Provide feedback in designated areas (e.g., comments, rating scale).
 5. Submit the feedback.
- * **Postconditions:**
 - * The feedback is visible to the learner who submitted the assignment.

* **Expected Result:** The peer reviewer successfully submits constructive feedback on the assigned assignment.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** System Testing, Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_PR_003

* **Test Case Title:** Verify that a structured review process guides the feedback.

* **Test Case Description:** This test case ensures that the platform provides clear guidelines and structure for reviewers to provide helpful and relevant feedback to learners.

* **Test Suite:** User Story Feature Testing - Peer Review

* **Test Priority:** Medium

* **Preconditions:**

- * An assignment has been submitted for peer review.

- * A peer reviewer accesses the assignment for review.

* **Test Data:** N/A

* **Test Steps:**

1. Navigate to the assignment for review.
2. Review the platform's provided instructions or guidelines for feedback.
3. Analyze the designated areas for feedback (text boxes, rating scales).

* **Postconditions:**

- * The peer reviewer understands the expected format and content of feedback.

* **Expected Result:** The platform offers a structured and clear review process that guides reviewers in providing constructive feedback.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TC_PR_004

* **Test Case Title:** Verify that learners can view feedback on their work.

* **Test Case Description:** This test case verifies a learner can access and view feedback provided on their submitted assignment.

* **Test Suite:** User Story Feature Testing - Peer Review

* **Test Priority:** High

* **Preconditions:**

- * A learner has submitted an assignment for peer review.

- * At least one reviewer has provided feedback on the assignment.

* **Test Data:** N/A

* **Test Steps:**

1. Login as the learner who submitted the assignment.
2. Navigate to their Submissions or My Feedback section within the course.
3. Locate the reviewed assignment.
4. View the submitted feedback from reviewers.

* **Postconditions:**

- * The learner is aware of the feedback received on their work.

* **Expected Result:** The learner can successfully view all feedback provided by reviewers on their submitted assignment.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** System Testing, Functional Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_PR_005

* **Test Case Title:** Verify anonymous feedback to ensure unbiased reviews.

* **Test Case Description:** This test case ensures that reviewers' identities are concealed from learners, promoting honest and objective feedback.

* **Test Suite:** User Story Feature Testing - Peer Review

* **Test Priority:** High

* **Preconditions:**

- * A learner has submitted an assignment for peer review.

- * A peer reviewer accesses the assignment for review.

* **Test Data:** N/A

* **Test Steps:**

1. Login as the learner who submitted the assignment.
2. Navigate to the feedback section for the reviewed assignment.
3. Observe the feedback provided by reviewers.

* **Postconditions:**

- * The learner cannot identify the reviewers who provided feedback.

* **Expected Result:** The platform preserves the anonymity of reviewers, protecting them and encouraging honest feedback.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, Security Testing

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 for the "Certificates and Badges" feature of the Online Learning Platform, following your provided template:

Test Case 1

* **Test Case ID:** TC_CERT_001

* **Test Case Title:** Verify that learners earn a certificate upon successful course completion.

* **Test Case Description:** This test case verifies that a learner receives a certificate after successfully completing all modules and assessments within a specific course.

* **Test Suite:** User Stories

* **Test Priority:** High

* **Preconditions:**

- * A learner account is created.

- * A course with modules and assessments is available.

- * The learner enrolls in the course.

* **Test Data:** Course ID, Learner ID

* **Test Steps:**

1. The learner logs in to the platform.
2. The learner navigates to the enrolled course.
3. The learner completes all modules and assessments within the course, achieving a passing score.
4. The learner views their course completion status.
5. The learner accesses their profile and confirms the presence of the earned certificate.

* **Postconditions:**

- * The certificate is downloaded and saved by the learner.

* **Expected Result:** The learner should see a visible indication of course completion and be able to access

and download the certificate from their profile.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TC_BADGE_001

* **Test Case Title:** Verify that badges are awarded for specific achievements and milestones.

* **Test Case Description:** This test case verifies that badges are awarded to learners for predefined achievements and milestones within a course or the overall platform.

* **Test Suite:** User Stories

* **Test Priority:** High

* **Preconditions:**

- * A learner account is created.

- * A course with specific milestones and achievements is available.

* **Test Data:** Milestones/Achievements within a course, Badge ID

* **Test Steps:**

1. The learner logs in to the platform.
2. The learner enrolls in the course with predefined milestones/achievements.
3. The learner completes the specified milestones/achievements (e.g., completing a module, attending a live session, achieving a certain score in a quiz).
4. The learner checks their profile to see if they have received the corresponding badge(s) for the completed milestones/achievements.

* **Postconditions:**

- * The learner downloads the newly earned badge(s).
- * **Expected Result:** The learner should receive the appropriate badge(s) after completing the specific milestones/achievements.
- * **Severity:** Critical
- * **Priority:** High
- * **Type of Testing:** Functional Testing, System Testing
- * **Test Case Approach:** Positive

Test Case 3

- * **Test Case ID:** TC_CERT_ACCESS_001
- * **Test Case Title:** Verify that certificates and badges are accessible through learner profiles.
- * **Test Case Description:** This test case confirms that learners can readily access their earned certificates and badges from their individual profiles.
- * **Test Suite:** User Stories
- * **Test Priority:** High
- * **Preconditions:**
 - * A learner account is created.
 - * The learner earns at least one certificate and one badge through course completion or milestones.
- * **Test Data:** Learner ID
- * **Test Steps:**

1. The learner logs in to the platform.
2. The learner navigates to their profile page.
3. The learner verifies the presence of a dedicated section for certificates and badges.
4. The learner clicks on a certificate and confirms that it displays properly.
5. The learner clicks on a badge and confirms that it displays properly.

* **Postconditions:**

* None

* **Expected Result:** The learner profile should clearly display a dedicated section for certificates and badges, allowing them to access and view their earned credentials.

* **Severity:** High

* **Priority:** High

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TC_SOCIAL_SHARE_001

* **Test Case Title:** Verify that learners can share their certificates and badges on social media.

* **Test Case Description:** This test case confirms that learners can share their earned certificates and badges directly to their preferred social media platforms.

* **Test Suite:** User Stories

* **Test Priority:** Medium

* **Preconditions:**

* A learner account is created with connected social media accounts (e.g., Facebook, Twitter).

* The learner has earned at least one certificate or badge.

* **Test Data:** Learner ID, Certificate or Badge data, Social Media Platform

* **Test Steps:**

1. The learner logs in to the platform.
2. The learner navigates to their profile and views their certificates/badges.
3. The learner selects the "Share" option for a certificate or badge.
4. The learner chooses their desired social media platform from the available options.

5. The learner confirms the sharing process and verifies the successful posting of their certificate/badge on their selected social media platform.

* **Postconditions:**

- * The learner reviews the shared content on their social media account.

* **Expected Result:** The learner's certificate or badge should be shared successfully on the chosen social media platform, including relevant information like the course title and achievement details.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC_CUSTOM_BADGE_001

* **Test Case Title:** Verify that instructors can issue custom badges for special achievements.

* **Test Case Description:** This test case checks the functionality for instructors to create and award custom badges to learners for exceptional achievements beyond predefined milestones.

* **Test Suite:** User Stories

* **Test Priority:** Medium

* **Preconditions:**

- * An instructor account is created for the specific course.

- * The instructor has access to the badge management functionality.

* **Test Data:** New badge name, description, image, eligibility criteria

* **Test Steps:**

1. The instructor logs into the platform.
2. The instructor navigates to the "Badges" or "Achievements" section in the course management area.

3. The instructor creates a new custom badge, providing a name, description, image, and eligibility criteria for awarding.

4. The instructor selects a learner and awards the custom badge.

5. The learner logs in and checks their profile to see the newly awarded custom badge.

* **Postconditions:**

* The custom badge appears in the learner's badge collection.

* **Expected Result:** The instructor should be able to create and assign a custom badge to a learner based on predefined eligibility criteria. The learner should receive and view the custom badge in their profile.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, System Testing

* **Test Case Approach:** Positive / Flow

Data - 60

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Gamification Elements

Feature description -

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

Userstory title -

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

Acceptance criteria -

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

Generated Test Cases:

Test Cases for Gamification Elements Feature

Here are 5 detailed test cases for the "Gamification Elements" feature of the Online Learning Platform, following the provided template:

****Test Case 1:****

*** **Test Case ID:** TC_GAM_001**

*** **Test Case Title:** Verify that learners earn points for completing activities.**

*** **Test Case Description:**** This test case verifies that learners receive points when they successfully complete various learning activities on the platform, such as watching videos, taking quizzes, and participating in forum discussions.

*** **Test Suite:** Gamification**

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

* **Preconditions:**

- * A learner account is created.
- * The learner is enrolled in a course containing various activities.

* **Test Data:** List of activities with corresponding point values.

* **Test Steps:**

1. Log in to the platform as a learner.
2. Access a course containing the specified activities.
3. Complete each activity successfully.
4. Navigate to the learner's gamification dashboard.
5. Verify that the total points displayed match the accumulated points from completed activities.

* **Postconditions:** No specific postconditions.

* **Expected Result:** The learner's gamification dashboard accurately reflects the earned points for completed activities.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case 2:

* **Test Case ID:** TC_GAM_002

* **Test Case Title:** Verify that levels are achieved based on accumulated points.

* **Test Case Description:** This test case verifies the level progression system based on accumulated points.

It ensures that learners are correctly promoted to the next level upon reaching the required point threshold.

* **Test Suite:** Gamification

* **Test Priority:** High

* **Preconditions:**

- * A learner account is created.

* The learner has earned sufficient points to reach at least two levels.

* **Test Data:** Point thresholds for each level.

* **Test Steps:**

1. Login to the platform as a learner.
2. Complete activities to ensure sufficient points are earned.
3. Verify that the learner's level is accurately displayed on their gamification dashboard.
4. Continue accumulating points until the required threshold for the next level is reached.
5. Confirm that the learner's level is updated on their gamification dashboard.

* **Postconditions:** No specific postconditions.

* **Expected Result:** Learners progress through levels as they accumulate points, and the displayed level accurately reflects their progress.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Test Case 3:

* **Test Case ID:** TC_GAM_003

* **Test Case Title:** Verify that leaderboards display top-performing learners.

* **Test Case Description:** This test case verifies the functionality of the leaderboard display, ensuring that it accurately showcases the top-performing learners based on their accumulated points.

* **Test Suite:** Gamification

* **Test Priority:** Medium

* **Preconditions:**

* Multiple learner accounts are created.

* Learners have accumulated different point values through activities.

* **Test Data:** List of learner accounts with varying point totals.

* **Test Steps:**

1. Access the leaderboard functionality on the platform.
2. Verify that the leaderboard displays learner usernames sorted based on their accumulated points in descending order.
3. Scroll through the leaderboard to ensure all listed learners are accurately ranked.
4. Validate that the displayed points for each learner reflect their current standing on the leaderboard.

* **Postconditions:** No specific postconditions.

* **Expected Result:** The leaderboard accurately displays the top-performing learners based on their points, with consistent ranking and point representations.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

Test Case 4:

* **Test Case ID:** TC_GAM_004

* **Test Case Title:** Verify that badges are awarded for reaching milestones.

* **Test Case Description:** This test case verifies that the system correctly awards badges to learners upon reaching specific point milestones or completing certain achievements within the platform.

* **Test Suite:** Gamification

* **Test Priority:** Medium

* **Preconditions:**

- * A learner account is created.

* **Test Data:** List of milestone point values and corresponding badges.

* **Test Steps:**

1. Login to the platform as a learner.
2. Complete activities to accumulate points and reach designated milestones.

3. Verify that the corresponding badge is awarded and displayed on the learner's profile or gamification dashboard.

4. Repeat the process for multiple milestones to confirm badge allocation accuracy.

* **Postconditions:** No specific postconditions.

* **Expected Result:** Learners receive badges upon reaching predefined milestones, accurately reflecting their achievements.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

****Test Case 5:****

* **Test Case ID:** TC_GAM_005

* **Test Case Title:** Verify that learners can view their progress in a gamification dashboard.

* **Test Case Description:** This test case verifies the learner's ability to access and view their progress within the gamification dashboard. It ensures that the dashboard clearly displays key information such as points, levels, badges earned, and leaderboards ranking.

* **Test Suite:** Gamification

* **Test Priority:** High

* **Preconditions:**

* A learner account is created.

* The learner has accumulated points, earned badges, and participated in activities.

* **Test Data:** N/A

* **Test Steps:**

1. Login to the platform as a learner.

2. Access the gamification dashboard.

3. Verify that their current point total, level, and earned badges are clearly displayed on the dashboard.

4. Navigate to the leaderboard section within the dashboard.

5. Confirm that their ranking on the leaderboard is accurate and reflects their current point accumulation.

* **Postconditions:** No specific postconditions.

* **Expected Result:** The gamification dashboard provides a user-friendly and comprehensive overview of the learner's progress, including points, level, badges, and leaderboard position.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

Data - 61

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Multilingual Support

Feature description -

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and

facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

Userstory title -

As a learner, I want multilingual support so that I can access the platform in my preferred language.

Acceptance criteria -

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated help and support resources.', 'Instructors can provide content in multiple languages.']

Generated Test Cases:

Test Cases for Multilingual Support Feature

Project Name: Online Learning Platform

Feature Name: Multilingual Support

User Story Title: As a learner, I want multilingual support so that I can access the platform in my preferred language.

Test Case 1

Test Case ID: TC_MLS_001

Test Case Title: Verify that the platform interface is available in multiple languages.

****Test Case Description:**** This test case aims to verify that the online learning platform offers a user interface that can be switched between various languages as specified in the requirements.

****Test Suite:**** User Interface (UI) Testing

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

****Preconditions:****

- * The platform is accessible with a valid account.
- * The user is not signed in.

****Test Data:**** List of supported languages

****Test Steps:****

1. Navigate to the platform homepage.
2. Locate and click on the language selection dropdown menu.
3. Select each supported language from the dropdown menu.
4. Verify that the entire user interface (menus, buttons, text) is successfully translated to the selected language.

****Postconditions:****

- * The user should be able to freely switch back to the original language.

*

****Expected Result:****

All platform elements are accurately translated into each supported language.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing, User Interface (UI) Testing, Acceptance Testing**

****Test Case 2****

****Test Case ID:** TC_MLS_002**

****Test Case Title:** Verify that course materials are available in different languages.**

****Test Case Description:** This test case ensures that the learning content (text, videos, downloadable resources) is provided in the selected language by the learner.**

****Test Suite:** Content Accessibility Testing**

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

****Preconditions:****

* A user account is created and logged in.

* A course with multilingual content is available.

****Test Data:**** A list of courses with multilingual content and the target languages for each course.

****Test Steps:****

1. Navigate to the course page with multilingual content.
2. Select the desired language from the language selection dropdown menu.
3. Review the course materials (text, videos, downloadable resources) to verify they are accurately translated into the selected language.

****Postconditions:****

* The user can restore the original language setting.

*

****Expected Result:****

All course materials (including video transcripts, if applicable) are delivered in the chosen language.

****Severity:**** Critical

****Priority:**** High

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

****Test Case 3****

****Test Case ID:** TC_MLS_003**

****Test Case Title:** Verify that learners can select their preferred language at any time.**

****Test Case Description:** This test case ensures that learners can change their preferred language dynamically throughout their platform interaction.**

****Test Suite:** User Interface (UI) Testing**

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

****Preconditions:****

* A user account is created and logged in.

* The user language has been set to an initially selected language.

****Test Data:** Several test languages including the initial selected language.**

****Test Steps:****

1. Access any section of the platform (courses, forum, profile settings).
2. Locate and access the language selection dropdown menu.
3. Choose a different language from the dropdown menu.
4. Verify that the entire interface updates to the selected language.

****Postconditions:****

* The user can revert back to the previous language setting.

*

****Expected Result:****

The platform seamlessly adjusts its language based on the learner's chosen preference.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, User Interface (UI) Testing, Acceptance Testing

****Test Case 4****

****Test Case ID:**** TC_MLS_004

****Test Case Title:**** Verify that multilingual support includes translated help and support resources.

****Test Case Description:****

This test case checks if the help documentation, FAQs, and other support resources are translated into all supported languages.

****Test Suite:**** Help & Support Testing

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

****Preconditions:****

- * Access to the platform help center or support sections.
- * A list of supported languages.

****Test Data:**** Categories of help topics and FAQs.

****Test Steps:****

1. Navigate to the platform's help center or support section.
2. Select a different language from the language selection dropdown menu.
3. Search for specific help topics or FAQs in the translated language.
4. Verify that the translations are accurate and comprehensive.

****Postconditions:****

- * The user can switch back to the original language.

****Expected Result:****

Help and support resources are translated accurately to provide assistance to learners in their preferred language.

****Severity:** Major**

****Priority:** Medium**

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

****Test Case 5****

****Test Case ID:** TC_MLS_005**

****Test Case Title:** Verify that instructors can provide content in multiple languages.**

****Test Case Description:****

This test case confirms that instructors have the capability to create and upload course content in various languages.

****Test Suite:** Instructor Content Creation Testing**

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

****Preconditions:****

- * An instructor account with appropriate permissions is logged in.
- * Access to the course creation or content uploading interface.

****Test Data:****

Sample course content in different languages.

****Test Steps:****

1. Log in as an instructor.
2. Create a new course or navigate to an existing course.
3. Upload course materials (text, videos, downloadable resources) in different languages.
4. Verify that the language selection option is available for learners when accessing the course content.

****Postconditions:****

*

****Expected Result:****

Instructors can create and manage content in multiple languages, enabling learners to explore courses in their preferred languages.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Instructor Content Creation Testing, Acceptance Testing

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:

Online Learning Platform Test Cases: Accessibility Features

User Story: As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

Here are 5 detailed test cases based on the provided acceptance criteria:

Test Case ID: TC_ACC_001

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

****Test Case Description:**** This test case verifies if the platform's content is properly read aloud by screen readers, ensuring navigability and understandability for visually impaired users.

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

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

****Preconditions:****

- * A screen reader application is installed and running.

- * A user account is created.

- * The user is logged in.

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

****Test Steps:****

1. Navigate to the platform's homepage using the screen reader.
2. Interact with various elements on the homepage (e.g., navigation menu, course listings, search bar) using keyboard navigation and screen reader commands.
3. Verify that screen reader announcements accurately describe the content and structure of each element.
4. Access a sample course and navigate through its modules and sections using the screen reader.
5. Ensure that all interactive elements (e.g., buttons, links, form fields) are announced and navigable.

****Postconditions:****

- * No errors or unexpected behavior should be encountered during screen reader interaction.

****Expected Result:**** Interactive elements and content are properly announced and understood by the screen reader, enabling seamless navigation and access for visually impaired users.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TC_ACC_002

****Test Case Title:**** Verify that learners can adjust font sizes for better readability

****Test Case Description:**** This test case ensures that learners can modify the text size to suit their visual preferences and needs.

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

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

****Preconditions:****

* A user account is created.

* The user is logged in.

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

****Test Steps:****

1. Navigate to the platform's homepage.
2. Locate the font size adjustment option (e.g., dropdown menu, slider).
3. Select different font sizes from the available options.
4. Verify that the text on the page scales accordingly for each selected font size.
5. Access different content areas (e.g., course modules, discussion forums, quizzes) and confirm font size adjustments are applied consistently.

****Postconditions:****

* No content overlaps or clipping should occur due to font size changes.

****Expected Result:**** Learners can easily adjust font sizes to a comfortable level, enhancing readability and accessibility for users with visual impairments or preferences.

****Severity:**** Major

****Priority:**** High

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

****Test Case Approach:**** Positive, Edge Case

****Test Case ID:**** TC_ACC_003

****Test Case Title:**** Verify that color contrast options are available for improved visibility

****Test Case Description:**** This test case confirms the availability and functionality of color contrast options, ensuring sufficient contrast for users with low vision.

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

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

****Preconditions:****

* A user account is created.

* The user is logged in.

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

****Test Steps:****

1. Navigate to the platform's accessibility settings.
2. Locate and select different color contrast options (e.g., light theme, dark theme, high contrast).
3. Verify that the platform's appearance changes accordingly, applying the selected color scheme.
4. Assess the color contrast between text and background elements in various sections (e.g., header, navigation, course content).
5. Confirm that the chosen color contrast settings provide sufficient contrast for comfortable reading and visibility for users with low vision.

****Postconditions:**** The chosen color contrast setting should remain consistent across different pages and sections of the platform.

****Expected Result:**** Learners can effectively adjust color contrast settings to optimize visual clarity and accessibility for their needs.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:** TC_ACC_004**

****Test Case Title:** Verify that accessibility features are easy to activate and customize**

****Test Case Description:** This test case determines the ease of access and customization of accessibility features for learners.**

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

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

****Preconditions:****

* A user account is created.

* The user is logged in.

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

****Test Steps:****

1. Locate the platform's accessibility settings.
2. Assess the clarity and organization of the accessibility options (e.g., labeling, dropdown menus, checkbox).
3. Activate each accessibility feature (e.g., screen reader compatibility, font size adjustment, color contrast) within the settings.
4. Verify that the features are applied immediately and effectively.
5. Customize the settings to personal preferences (e.g., selecting a specific font size, applying a color contrast theme).

****Postconditions:****

* All accessibility features should be easily identifiable and manageable within the settings menu.

****Expected Result:**** Learners can readily find, activate, and customize accessibility features without encountering complexity or confusion.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Usability Testing, Accessibility Testing

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

****Test Case ID:**** TC_ACC_005

****Test Case Title:**** Verify accessibility resources are available to assist learners with disabilities

****Test Case Description:**** This test case ensures that the platform provides accessible support resources for learners with disabilities.

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

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

****Preconditions:****

* A user account is created.

* The user is logged in.

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

****Test Steps:****

1. Locate the platform's "Support" or "Help" section.
2. Search for information about accessibility features or specific accessibility assistance.
3. Verify that the platform provides clear and concise descriptions of its accessibility features.
4. Check if the platform offers links to external resources or guides for assistive technology.
5. Explore if contact options are available for seeking personalized assistance regarding accessibility.

****Postconditions:**** The support resources should be easily accessible and comprehensive for learners seeking help with accessibility features.

****Expected Result:**** Learners can find helpful and accessible information regarding the platform's accessibility features, along with options for personalized support if needed.

****Severity:**** Major

****Priority:**** Medium

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

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

Data - 63

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Moderation

Feature description -

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

Userstory title -

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

Acceptance criteria -

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

Generated Test Cases:

Test Cases for Discussion Moderation Feature

****Test Case ID:** TC_DISC_001**

****Test Case Title:** Verify that instructors can edit discussions posts.**

****Test Case Description:** This test case verifies the functionality for instructors to edit posts within discussion forums. It will test the editing functionality for basic text, formatting, and any other relevant post attributes.**

****Test Suite:** Discussion Moderation**

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

****Preconditions:****

* An instructor account is logged in.

* An existing discussion forum with at least one post is available.

****Test Data:** Test post content with basic text and formatting.**

****Test Steps:****

1. Navigate to the specified discussion forum.
2. Locate a post authored by a user other than the instructor.
3. Click on the edit button associated with the post.
4. Modify the post content by changing text, formatting, or other relevant attributes.
5. Click on the "Save Changes" button.

****Postconditions:****

* The post content is updated with the changed information.

****Expected Result:** The post is successfully edited, and the changes are reflected in the discussion forum.**

****Severity:** Critical**

****Priority:** High**

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

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

****Test Case ID:** TC_DISC_002**

****Test Case Title:** Verify that instructors can delete discussion posts.**

****Test Case Description:**** This test case verifies the functionality for instructors to delete posts from discussion forums. It will test the deletion functionality for various post types and verify the removal of associated data.

****Test Suite:** Discussion Moderation**

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

****Preconditions:****

- * An instructor account is logged in.
- * An existing discussion forum with at least one post is available.

****Test Data:** Test post content.**

****Test Steps:****

1. Navigate to the specified discussion forum.
2. Locate a post authored by a user other than the instructor.
3. Click on the delete button associated with the post.
4. Confirm the deletion by clicking on the "Delete" button in the pop-up window.

****Postconditions:****

- * The selected post is removed from the discussion forum.

****Expected Result:**** The post is successfully deleted, and it is no longer visible in the discussion forum.

****Severity:** Critical**

****Priority:** High**

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

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

****Test Case ID:** TC_DISC_003**

****Test Case Title:**** Verify that instructors can highlight discussion posts.

****Test Case Description:**** This test case verifies the functionality for instructors to highlight posts within discussion forums. It will test the highlighting functionality for various post types and verify the visual representation of highlighted posts.

****Test Suite:**** Discussion Moderation

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

****Preconditions:****

* An instructor account is logged in.

* An existing discussion forum with at least one post is available.

****Test Data:**** Test post content.

****Test Steps:****

1. Navigate to the specified discussion forum.
2. Locate a post authored by a user other than the instructor.
3. Click on the highlight button associated with the post.
4. Observe the visual representation of the highlighted post.

****Postconditions:****

* The post is visually highlighted, differentiating it from other posts.

****Expected Result:**** The post is successfully highlighted, and the change is visually apparent to other users.

****Severity:**** Major

****Priority:**** Medium

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

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

****Test Case ID:**** TC_DISC_004

****Test Case Title:**** Verify that instructors can mute disruptive users.

****Test Case Description:**** This test case verifies the functionality for instructors to mute disruptive users in discussion forums. It will test the ability to mute users and prevent them from posting new messages or participating in discussions.

****Test Suite:**** Discussion Moderation

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

****Preconditions:****

* An instructor account is logged in.

* An existing discussion forum with at least one disruptive user is available.

****Test Data:**** Test usernames for disruptive users.

****Test Steps:****

1. Locate the disruptive user within the discussion forum.
2. Click on the mute button associated with the disruptive user.
3. Verify that the user is muted and cannot post new messages.

****Postconditions:****

* The muted user is unable to participate in the discussion.

****Expected Result:**** The user is successfully muted, preventing them from posting new messages and participating in further discussions.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** TC_DISC_005

****Test Case Title:**** Verify that moderation actions are logged and reported.

****Test Case Title:**** This test case verifies whether moderation actions taken by instructors are properly logged and reported. It will involve simulating moderation actions (e.g., deleting posts, muting users) and then checking the logs for accurate recordings.

****Test Suite:**** Discussion Moderation

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

****Preconditions:****

* An instructor account is logged in.

* An existing discussion forum with at least one post and user is available.

****Test Data:**** Test post content, usernames for flagged users.

****Test Steps:****

1. Delete a test post under the instructor account.
2. Mute a user in the discussion forum.
3. Navigate to the moderation log or reporting section.
4. Verify the existence of entries for the deletion of the test post and the user being muted.

****Postconditions:****

* Moderation log or report contains records of the conducted actions (deleting a post and muting a user).

****Expected Result:**** The moderation log accurately reflects the date, time, and details of all moderation actions taken by instructors.

****Severity:**** Major

****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:

Test Cases for Learning Analytics Feature

****User Story:**** As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

Here are five detailed test cases generated to cover the provided acceptance criteria:

****Test Case 1:****

* ****Test Case ID:**** TC_LA_001

* ****Test Case Title:**** Verify that engagement metrics are displayed on the Learning Analytics dashboard.

* ****Test Case Description:**** This test case verifies that the Learning Analytics dashboard correctly displays key engagement metrics for the course, such as number of learners, average time spent per session, and completion rates.

* ****Test Suite:**** User Interface Testing

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

* ****Preconditions:****

- * Instructor successfully logs into the platform.
- * At least one course is assigned to the instructor.
- * Learners have interacted with the course content.

* ****Test Data:**** Course data with learner activities.

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

1. Navigate to the Learning Analytics dashboard for the specific course.
2. Verify the presence of engagement metrics such as:
 - * Number of learners enrolled

- * Average time spent per session
- * Completion rate for modules/lessons
- * Number of forum posts/comments

3. Compare the displayed metrics with the actual recorded data.

* **Postconditions:**

- * Instructor remains logged into the platform.
- * Dashboard page unchanged.

* **Expected Result:** All engagement metrics are displayed accurately and correspond to the actual learner activity data.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TC_LA_002

* **Test Case Title:** Verify that performance trends are visualized over time on the Learning Analytics dashboard.

* **Test Case Description:** This test case verifies the ability to visualize learner performance trends, such as quiz scores and assignment submissions, over different time intervals.

* **Test Suite:** User Interface Testing

* **Test Priority:** High

* **Preconditions:**

- * Instructor successfully logs into the platform.
- * Course with quizzes and assignments is assigned to the instructor.
- * Learners have submitted at least one quiz and one assignment.

* **Test Data:** Course data with learner performance records.

* **Test Steps:**

1. Navigate to the Learning Analytics dashboard for the specific course.
2. Select specific performance metric (e.g., average quiz score, assignment completion rate).
3. Choose a time interval to visualize the trend (e.g., weekly, monthly).
4. Observe the visualization (e.g., chart, graph) displaying performance trends over the selected time period.

* **Postconditions:**

* Instructor remains logged into the platform.

* Dashboard page unchanged.

* **Expected Result:** The selected performance metric is visualized accurately over the chosen time intervals, demonstrating observed trends in learner performance.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_LA_003

* **Test Case Title:** Verify that instructors can access detailed reports on learner progress.

* **Test Case Description:** This test case verifies the functionality of detailed learner progress reports, ensuring instructors can view individual learners' performance and track their progress.

* **Test Suite:** User Interface Testing

* **Test Priority:** High

* **Preconditions:**

* Instructor successfully logs into the platform.

* At least one course is assigned to the instructor.

- * Learners are enrolled in the course and have submitted assignments/completed quizzes.

- * **Test Data:** Course data with learner activity and submission records.

- * **Test Steps:**

1. Navigate to the Learning Analytics dashboard for the specific course.

2. Select the "Learner Progress Reports" option.

3. Choose a specific learner from the list.

4. Review the detailed report, including:

- * Course completion status

- * Assignment scores and submissions

- * Quiz results and performance

- * Forum participation

- * **Postconditions:**

- * Instructor remains logged into the platform.

- * Report page is displayed.

- * **Expected Result:** The system displays a comprehensive report on the selected learner's progress, capturing all relevant data points across different course activities.

- * **Severity:** Critical

- * **Priority:** High

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

- * **Test Case Approach:** Positive

****Test Case 4:****

- * **Test Case ID:** TC_LA_004

- * **Test Case Title:** Verify real-time data updates in the Learning Analytics dashboard.

- * **Test Case Description:** This test case ensures that analytics data on the dashboard is updated in real-time as learners interact with the course content.

* **Test Suite:** Performance Testing

* **Test Priority:** Medium

* **Preconditions:**

- * Instructor successfully logs into the platform.

- * Learners are actively engaged in the course (submitting assignments, taking quizzes, interacting in forums).

* **Test Data:** None

* **Test Steps:**

1. Open the Learning Analytics dashboard for the course.
2. Monitor key metrics (e.g., Number of learners online, Recent activity) for a specified time period (e.g., 10 minutes).
3. Initiate learner activity (e.g., submit assignment, post in forum).
4. Observe the dashboard metrics and verify the update reflection within a reasonable timeframe (e.g., within 1 minute).

* **Postconditions:**

- * Instructor remains logged into the platform.

- * Dashboard page unchanged.

* **Expected Result:** The dashboard metrics reflect the real-time changes in learner activity, with updates occurring promptly after the activity.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Performance Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_LA_005

* **Test Case Title:** Verify instructors can export analytics reports in various formats.

* **Test Case Description:** This test case ensures instructors can export detailed learning analytics reports in different formats for further analysis and sharing.

* **Test Suite:** User Functionality Testing

* **Test Priority:** Medium

* **Preconditions:**

- * Instructor successfully logs into the platform.

- * Relevant analytics data is available for the course.

* **Test Data:** Course data with enough activity to generate report content

* **Test Steps:**

1. Navigate to the Learning Analytics dashboard for the specific course.
2. Generate a detailed report (e.g., Learner Progress Report).
3. Click the "Export" button and select a desired format (e.g., PDF, CSV, Excel).
4. Observe the exported file and verify its content.

* **Postconditions:**

- * Instructor remains logged into the platform.

- * Exported file saved to the designated location.

* **Expected Result:**

The system successfully exports the selected learning analytics report in the chosen format, retaining all relevant data and formatting.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 65

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Learning Features

Feature description -

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

Userstory title -

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Acceptance criteria -

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

Generated Test Cases:

Test Cases for Online Learning Platform - Social Learning Features

****Test Case ID:**** OLP-SLF-001

****Test Case Title:**** Verify that learners can create a new group for a project.

****Test Case Description:**** This test case verifies the functionality of creating a new group for a project from a learner's perspective.

****Test Suite:**** Social Learning Features

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

****Preconditions:****

- * A logged-in learner account exists.
- * The platform is accessible.
- * No existing groups with the same name.

****Test Data:****

- * Group Name: "Project Management Team"
- * Group Description: "A group for collaborating on a project management course"

****Test Steps:****

1. Navigate to the "Groups" section of the platform.
2. Click on the "Create New Group" button.
3. Enter the group name and description.
4. Select appropriate privacy settings (public/private).
5. Click on the "Create Group" button.

****Postconditions:****

- * A new group with the specified name and description is created.
- * The learner is automatically added as a member of the new group.

****Expected Result:****

The learner successfully creates a new group for a project.

****Severity:** Critical**

****Priority:** High**

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

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

****Test Case ID:** OLP-SLF-002**

****Test Case Title:** Verify that learners can share resources within a group.**

****Test Case Description:** This test case verifies the functionality of sharing different types of resources (documents, links, videos) within a group forum.**

****Test Suite:** Social Learning Features**

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

****Preconditions:****

- * A logged-in learner account exists.
- * A group has been created.
- * A resource file (e.g., document, presentation) is available on the learner's device.

****Test Data:****

- * Resource File: ProjectProposal.docx
- * Group Name: "Project Management Team"

****Test Steps:****

1. Navigate to the group forum.

2. Click on the "Share Resource" button.
3. Select the resource file from the learner's device.
4. Add a brief description of the resource.
5. Click on the "Share" button.

****Postconditions:****

- * The shared resource file is visible to all group members in the forum.

****Expected Result:****

The learner successfully shares a resource within the group.

****Severity:** Critical**

****Priority:** High**

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

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

****Test Case ID:** OLP-SLF-003**

****Test Case Title:** Verify that learners can provide feedback on group projects.**

****Test Case Description:** This test case verifies the functionality of leaving feedback on a group project within a group forum.**

****Test Suite:** Social Learning Features**

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

****Preconditions:****

- * A logged-in learner account exists.
- * A group has been created with an assigned group project.

****Test Data:****

- * Project Name: "MERN Stack Web App Development"
- * Feedback Message: "Great work on the front-end design! Suggestion: consider optimizing the database queries for better performance."

****Test Steps:****

1. Navigate to the group forum relevant to the project.
2. Find the project discussion thread.
3. Click on the "Post Feedback" button or similar.
4. Enter the feedback message in the designated field.
5. Click on the "Submit Feedback" button.

****Postconditions:****

- * The feedback message is visible to all group members in the project discussion thread.

****Expected Result:****

The learner successfully provides feedback on the group project.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** OLP-SLF-004

****Test Case Title:**** Verify that instructors can assign group projects.

****Test Case Description:**** This test case verifies the functionality of assigning a group project to a specific group within a course.

****Test Suite:**** Social Learning Features

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

****Preconditions:****

- * A logged-in instructor account exists.
- * A course is created with available sections.
- * A group has been created within the course.

****Test Data:****

- * Course Name: "Data Science Fundamentals"
- * Group Name: "Data Wrangling Team"
- * Project Name: "Sales Data Analysis Report"

****Test Steps:****

1. Navigate to the "Course Management" section.
2. Select the relevant course.
3. Access the "Group Projects" tab.
4. Click on "Add New Project" button.
5. Select the target group "Data Wrangling Team".
6. Enter the project name and description.
7. Set due date and time for submission.
8. Click the "Assign Project" button.

****Postconditions:****

- * The assigned project is visible to members of the "Data Wrangling Team" within the group forum within the course.

****Expected Result:****

The instructor successfully assigns a group project to a specific group.

****Severity:**** Critical

****Priority:**** High

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

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

****Test Case ID:**** OLP-SLF-005

****Test Case Title:**** Verify that instructors can monitor group project progress.

****Test Case Description:**** This test case verifies the functionality allowing instructors to view the progress and submissions of group projects.

****Test Suite:**** Social Learning Features

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

****Preconditions:****

- * A logged-in instructor account exists.
- * A course is created with available sections.
- * A group project is assigned to a group.

****Test Data:****

- * Course Name: "Data Science Fundamentals"
- * Group Name: "Data Wrangling Team"
- * Project Name: "Sales Data Analysis Report"

****Test Steps:****

1. Navigate to the "Course Management" section.

2. Select the relevant course.
3. Access the "Group Projects" tab.
4. View the project entry for "Sales Data Analysis Report"

****Postconditions:****

- * The instructor can see the project submission status (submitted, pending, etc.) for the assigned group.

****Expected Result:****

The instructor can monitor group project progress.

****Severity:** Critical**

****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:

Test Cases for Content Authoring Tools:

****Test Case ID:**** TC_CA_001

****Test Case Title:**** Verify that instructors can create multimedia presentations.

****Test Case Description:**** This test case verifies the functionality of the content authoring tools to create multimedia presentations with embedded images, audio, and video.

****Test Suite:**** Content Authoring Tools

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

****Preconditions:****

* An instructor account is logged in.

* A new course is created.

****Test Data:****

* Images in various formats (jpg, png, gif).

* Audio files (mp3, wav).

* Video files (mp4, mov).

****Test Steps:****

1. Navigate to the content authoring tool within the course.

2. Select the "Presentation" content type.

3. Upload an image, audio file, and video file to the presentation.

4. Arrange the multimedia elements on the slide.

5. Preview the presentation to ensure all elements are displayed correctly.

****Postconditions:****

* The multimedia presentation is saved successfully.

****Expected Result:**** The multimedia presentation is created successfully with embedded images, audio, and video. All elements are displayed correctly in the preview.

****Severity:** Critical**

****Priority:** High**

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

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

****__****

****Test Case ID:** TC_CA_002**

****Test Case Title:** Verify that interactive modules can be developed and integrated.**

****Test Case Description:** This test case verifies the functionality of creating and integrating interactive modules, such as quizzes, polls, and drag-and-drop exercises.**

****Test Suite:** Content Authoring Tools**

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

****Preconditions:****

* An instructor account is logged in.

* A new course is created.

****Test Data:****

* Question text, answer choices, correct answers for a quiz.

* Instructions and choices for a poll.

****Test Steps:****

1. Navigate to the content authoring tool within the course.
2. Select the "Interactive Module" content type.
3. Choose a module type (e.g., quiz, poll, drag-and-drop).
4. Configure the module with relevant questions, answer choices, or instructions.
5. Preview the module to ensure functionality and correctness.

****Postconditions:****

* The interactive module is saved successfully.

****Expected Result:**** An interactive module is created and integrated into the course successfully. The module functions as expected based on the chosen type and configuration.

****Severity:**** Critical

****Priority:**** High

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

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

****__****

****Test Case ID:**** TC_CA_003

****Test Case Title:**** Verify that rich media resources are supported.

****Test Case Description:**** This test case verifies the support of various rich media resources, such as videos, animations, and embedded documents, within content authoring tools.

****Test Suite:**** Content Authoring Tools

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

****Preconditions:****

* An instructor account is logged in.

* A new course is created.

****Test Data:****

* Embed code for a YouTube video.

* Animated GIF file.

* PDF document.

****Test Steps:****

1. Navigate to the content authoring tool within the course.
2. Create a new content block or section.
3. Upload, embed, or paste the rich media resource (video, animation, or document).
4. Preview the content to check for proper display and functionality.

****Postconditions:****

* The rich media resource is successfully integrated into the content.

****Expected Result:**** The rich media resource is displayed correctly within the content. Features like video playback, animation, and document preview are functional.

****Severity:**** Critical

****Priority:**** High

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

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

****__****

****Test Case ID:**** TC_CA_004

****Test Case Title:**** Verify that content authoring tools are user-friendly and intuitive.

****Test Case Description:**** This test case evaluates the user-friendliness and intuitiveness of the content authoring tools based on ease of navigation, clear labeling, and logical workflow.

****Test Suite:**** Content Authoring Tools

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

****Preconditions:****

* An instructor account is logged in without prior familiarity with the tools.

****Test Data:****

* New course context.

****Test Steps:****

1. Open the content authoring tool for the first time within the course.
2. Explore the main toolbar and menu options.
3. Attempt to create a simple multimedia presentation using the drag-and-drop interface.
4. Access different content types (text, images, videos, etc.) and experiment with their respective features.
5. Navigate through the editing and preview functionalities.

****Postconditions:****

* Basic content is created and previewed successfully.

****Expected Result:**** The instructor can easily understand the layout and functionalities of the content authoring tool. Navigation is intuitive, and labeling is clear, allowing for a smooth experience in creating basic course content.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Usability Testing

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

****__****

****Test Case ID:** TC_CA_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 the created content and make necessary edits before making it available to learners.**

****Test Suite:** Content Authoring Tools**

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

****Preconditions:****

*** An instructor account is logged in.**

*** Content is created using the authoring tools.**

****Test Data:****

*** Content with text, images, and multimedia elements.**

****Test Steps:****

1. Create content within the course using the authoring tools.
2. Click the "Preview" button to view the content in a simulated learner environment.
3. Identify and make edits to the content based on the preview.
4. Save the edited content.

****Postconditions:****

*** Content is saved with the edits.**

****Expected Result:** The preview functionality displays the content accurately, and instructors can make**

edits before publishing the finalized version.

****Severity:**** Major

****Priority:**** High

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

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

Data - 67

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Paths and Curriculum

Feature description -

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence

and continuity in learners' educational journeys.

Userstory title -

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

Acceptance criteria -

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

Generated Test Cases:

Following the provided template, here are five detailed test cases for the "Learning Paths and Curriculum" feature:

****Test Case 1****

* **Test Case ID:** LP-001

* **Test Case Title:** Verify that a learning path includes sequential courses and modules.

* **Test Case Description:** This test case verifies that items within a learning path are presented in a logical, sequential order. Learners should progress through courses and modules in the intended sequence.

* **Test Suite:** Learning Paths Functionality

* **Test Priority:** High

* **Preconditions:**

- * A learning path with multiple courses and modules is created.

- * A learner account is created.

* **Test Data:**

* **Test Steps:**

1. Navigate to the learning path.

2. Access the first course in the path.
3. Successfully complete the first course.
4. Verify that the next course in the path is presented or accessible.

* **Postconditions:**

- * The learner remains within the learning path.

* **Expected Result:** The learner is able to progress through the courses and modules in the designated order.

****Test Case 2****

* **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 each course and module within a curriculum clearly connects to a defined learning outcome.

* **Test Suite:** Curriculum Content

* **Test Priority:** High

* **Preconditions:**

- * A learning path is created with multiple courses and modules.

- * Each course and module has associated learning objectives.

* **Test Data:** List of learning objectives for each course and module

* **Test Steps:**

1. Access the learning path and select a course.
2. Review the course description and objectives listed.
3. Complete the course modules.
4. Assess whether the completed modules facilitated the learners' attainment of the stated objectives.

* **Postconditions:**

* **Expected Result:** There is a clear and direct connection between the course/module content and the defined learning objectives.

****Test Case 3****

* **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 the learner interface effectively displays their progress through a learning path.

* **Test Suite:** Learner Dashboard/Tracking

* **Test Priority:** Medium

* **Preconditions:**

- * A learner account is created.

- * A learning path with multiple courses and modules is assigned to the learner.

* **Test Data:**

* **Test Steps:**

1. Login as the learner.
2. Navigate to the "My Learning Paths" section.
3. Select the active learning path.
4. Verify that the learner's progress is visible, e.g., completion percentage, modules marked as finished.

* **Postconditions:**

* **Expected Result:** The learner interface clearly and accurately shows their progress through the learning

path.

****Test Case 4****

* ****Test Case ID:**** LP-004

* ****Test Case Title:**** Verify that instructors can customize learning paths for different needs.

* ****Test Case Description:**** This test case confirms instructors have the ability to modify or tailor existing learning paths to address various learning styles, skill levels, or specific audience requirements.

* ****Test Suite:**** Instructor Tools

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

* ****Preconditions:****

- * An instructor account is created.

- * A learning path is available for editing.

* ****Test Data:**** Modified learning path structure or content

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

1. Login as the instructor.
2. Access the learning path editing interface.
3. Modify the learning path (e.g., reorder courses, add/remove modules, update content).
4. Save the changes.

* ****Postconditions:****

* ****Expected Result:**** The instructor is able to successfully customize the learning path to meet specific needs.

****Test Case 5****

* **Test Case ID:** LP-005

* **Test Case Title:** Verify that notifications are sent for upcoming courses or milestones within a learning path.

* **Test Case Description:** This test case ensures the notification system effectively alerts learners about approaching deadlines, new course releases, or milestone achievements within their learning paths. *

* **Test Suite:** Communication/Notifications

* **Test Priority:** Medium

* **Preconditions:**

* A learner account is created and enrolled in a learning path with upcoming courses or milestones.

* **Test Data:**

* **Test Steps:**

1. Log in as the learner.
2. Verify that notifications are received for upcoming courses or milestones as specified in the learning path.
3. Check the notification settings to ensure they are configured correctly.

* **Postconditions:**

* **Expected Result:** Learners receive timely notifications for relevant events within their learning paths.

Data - 68

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Attendance Tracking

Feature description -

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

Userstory title -

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Acceptance criteria -

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

Generated Test Cases:

Test Case Creation for Online Learning Platform: Attendance Tracking Feature

Here are 5 detailed test cases for the "Attendance Tracking" feature of the Online Learning Platform, based on the provided user story and acceptance criteria.

****Test Case 1****

*** **Test Case ID:** TC_ATT_01**

* **Test Case Title:** Verify that attendance is automatically recorded during live sessions.

* **Test Case Description:** This test case aims to validate the automatic recording of learner attendance during live sessions conducted on the platform. It will simulate a live session and confirm that attendance is accurately captured for participating learners.

* **Test Suite:** Attendance Tracking Feature

* **Test Priority:** High

* **Preconditions:**

- * A live session is scheduled and configured on the platform.
- * Learners are enrolled in the live session.

* **Test Data:** List of enrolled learners, session duration

* **Test Steps:**

1. Login as an Instructor and access the scheduled live session.
2. Start the live session.
3. Multiple learners join the live session.
4. Instructors interacts with learners during the session.
5. End the live session.
6. Navigate to the attendance report for the session.

* **Postconditions:**

- * Access the attendance report

* **Expected Result:** Attendance records for all learners who joined and actively participated in the live session should be accurately reflected in the attendance report.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** System Testing, Functional Testing

****Test Case 2****

* **Test Case ID:** TC_ATT_02

* **Test Case Title:** Verify that instructors can view attendance reports for each session.

* **Test Case Description:** This test case validates the instructor's ability to access and view detailed attendance reports for completed live sessions. It will check the report's content, including learner presence, duration, and any interactive activity.

* **Test Suite:** Attendance Tracking Feature

* **Test Priority:** High

* **Preconditions:**

- * A live session has been conducted and concluded successfully.

* **Test Data:** Session ID

* **Test Steps:**

1. Login as an instructor.
2. Navigate to the "Live Sessions" or "Attendance" section.
3. Select the completed live session from the list.
4. Open the attendance report for the selected session.

* **Postconditions:**

* **Expected Result:** The attendance report should display a comprehensive overview of learner attendance, including:

- * Learner names and their status (Present/Absent)
- * Session duration
- * Time of arrival and departure for each learner (if applicable)
- * Any other relevant participation metrics collected during the session.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** System Testing, Functional Testing

Test Case 3

* **Test Case ID:** TC_ATT_03

* **Test Case Title:** Verify that learners can see their attendance records.

* **Test Case Description:** This test case ensures learners can access their own attendance records for past sessions, promoting transparency and self-monitoring.

* **Test Suite:** Attendance Tracking Feature

* **Test Priority:** Medium

* **Preconditions:**

- * A learner has participated in at least one live session.

* **Test Data:** Learner username and password.

* **Test Steps:**

1. Login as a learner who attended a live session.
2. Navigate to the "My Profile" or "Course History" section.
3. Locate the attendance records section.
4. View the attendance records for past live sessions.

* **Postconditions:**

* **Expected Result:** The learner should be able to access their attendance records clearly displaying their presence or absence status for each session attended.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** System Testing, Functional Testing

Test Case 4

* **Test Case ID:** TC_ATT_04

* **Test Case Title:** Verify that attendance data is integrated with participation metrics.

* **Test Case Description:** This test case validates that attendance records are supplemented with relevant

participation metrics, providing instructors with a holistic view of learner engagement.

* **Test Suite:** Attendance Tracking Feature

* **Test Priority:** Medium

* **Preconditions:**

- * A live session has been conducted with learner participation.

* **Test Data:** Session ID

* **Test Steps:**

1. Login as an instructor.
2. Navigate to the attendance report for the live session.
3. Examine the attendance data for each learner.

* **Postconditions:**

* **Expected Result:** Attendance records should be accompanied by relevant participation metrics, such as:

- * Number of questions answered during Q&A sessions.
- * Time spent actively engaged in the live session.
- * Participation in chat discussions or collaborative activities.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** System Testing, Functional Testing

****Test Case 5****

* **Test Case ID:** TC_ATT_05

* **Test Case Title:** Verify that notifications are sent for missed sessions.

* **Test Case Description:** This test case confirms that learners receive notifications when they miss live sessions, promoting accountability and reducing missed opportunities.

* **Test Suite:** Attendance Tracking Feature

* **Test Priority:** Medium

* **Preconditions:**

* A future live session is scheduled and learners are enrolled.

* **Test Data:** Learner email address

* **Test Steps:**

1. Login as an instructor.
2. Schedule a live session and set a notification system for missed sessions.
3. Login as a learner enrolled in the session.
4. Intentionally miss the scheduled live session.

* **Postconditions:**

* **Expected Result:** The learner should receive a notification via email shortly after the session concludes, informing them that they missed the session.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** System Testing, Functional Testing

These test cases provide a comprehensive starting point for ensuring the quality and functionality of the attendance tracking feature within the Online Learning Platform.

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:

Test Cases for Content Licensing and Rights Management Feature

****User Story:**** As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

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

****Test Case Title:** Verify that content licenses are assigned to each resource.**

****Test Case Description:** This test case verifies that each resource (lecture video, tutorial, downloadable material) uploaded by a content creator is associated with a specific content license.**

****Test Suite:** Content Licensing and Rights Management**

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

****Preconditions:****

* A Content Creator account should be created and logged in.

* At least three different types of resources should be uploaded by the Content Creator.

****Test Data:****

* Sample video lecture, tutorial document, and downloadable quiz.

* Different types of content licenses (e.g., Creative Commons Attribution, Public Domain, Proprietary).

****Test Steps:****

1. Log in as a Content Creator.
2. Upload a video lecture and select a "Creative Commons Attribution" license.
3. Upload a tutorial document and select a "Public Domain" license.
4. Upload a downloadable quiz and select a "Proprietary" license.
5. Go to the uploaded resources section.
6. View the licensing details of each resource and confirm that the assigned license type is correct.

****Postconditions:****

* Uploaded resources should remain accessible.

****Expected Result:** All uploaded resources should display a clearly defined content license.**

****Severity:** Critical**

****Priority:** High**

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

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

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

****Test Case Title:** Verify that usage rights and permissions are clearly defined.**

****Test Case Description:** This test case validates that the platform clearly outlines the usage rights and permissions associated with each content license.**

****Test Suite:** Content Licensing and Rights Management**

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

****Preconditions:****

* A Content Creator account should be created and logged in.

* A content resource (e.g., video lecture) with a specific content license assigned.

****Test Data:****

N/A

****Test Steps:****

1. Log in as a Content Creator.
2. Upload a video lecture and select a "Creative Commons Attribution" license.
3. Navigate to the content licensing details of the uploaded video lecture.
4. Read the explicitly stated usage rights and permissions associated with the "Creative Commons Attribution" license.
5. Verify that the usage rights and permissions are understandable and unambiguous.

****Postconditions:****

- * No modifications are made to the uploaded resource.

****Expected Result:**** The platform clearly defines the usage rights and permissions granted by the selected content license.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, User Interface (UI) Testing

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

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

****Test Case Title:**** Verify content creators can set restrictions on usage and distribution.

****Test Case Description:**** This test case confirms the ability of content creators to impose limitations on how their content can be used and distributed.

****Test Suite:**** Content Licensing and Rights Management

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

****Preconditions:****

- * A Content Creator account should be created and logged in.

- * A template content resource (e.g., text document) should be available for manipulation.

****Test Data:****

- * Sample text document.

- * Specific usage restrictions (e.g., non-commercial use only, restricted geographical distribution).

****Test Steps:****

1. Log in as a Content Creator.
2. Using the platform interface, select the uploaded text document.
3. Access the content licensing settings.
4. Define specific usage restrictions for the document, such as "non-commercial use only" and "limited to users within North America."
5. Save the changes.
6. Attempt to download and distribute the document outside the defined usage restrictions (e.g., for commercial purposes or to users outside North America).

****Postconditions:****

- * Attempt to use the resource beyond allowed restrictions should result in an error message or access denial.

****Expected Result:**** The platform enforces the usage restrictions set by the content creator, preventing unauthorized access or distribution.

****Severity:**** Severe

****Priority:**** High

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

****Test Case Approach:**** Negative

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

****Test Case Title:**** Verify instructors and learners are informed of content licensing terms.

****Test Case Description:**** This test case ensures that instructors and learners are presented with clear and

accessible information about the content licensing terms before using or accessing protected materials.

****Test Suite:**** Content Licensing and Rights Management

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

****Preconditions:****

- * An Instructor account and a Learner account should be created.

- * A content resource with a specific content license assigned should be accessible through a course.

****Test Data:****

- * Content resource with a content license (e.g., "Creative Commons Attribution").

****Test Steps:****

1. Log in as an Instructor.
2. Add the content resource to a course.
3. Log out as the Instructor and log in as a Learner.
4. Enroll the Learner account in the course.
5. Access the course and the content resource.
6. Review the displayed information regarding the content license terms.

****Postconditions:****

- * No changes are made to the course content.

****Expected Result:**** Both Instructors and Learners are presented with clear and concise information about the content license associated with the resource within the course context.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, User Interface (UI) Testing

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

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

****Test Case Title:**** Verify notifications are sent for license expiration or renewal.

****Test Case Description:**** This test case confirms the functionality of the system to notify content creators about upcoming license expirations and renewals.

****Test Suite:**** Content Licensing and Rights Management

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

****Preconditions:****

* A Content Creator account should be created and logged in.

* A content resource with a specific content license has an expiration date set.

****Test Data:****

* Sample content resource with expiring content license.

* Valid email address associated with the Content Creator account.

****Test Steps:****

1. Log in as the Content Creator.
2. Upload a content resource with a defined content license expiration date in the future.
3. Ensure the platform is configured to send email notifications for license expirations.
4. Wait for the specified date (within a reasonable timeframe).

5. Check the email inbox associated with the Content Creator account for notifications regarding the license expiration.

****Postconditions:****

* License expiration date status is updated within the platform.

****Expected Result:**** The platform sends a timely notification email to the content creator, alerting them about the approaching license expiration.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Notification System 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 5 detailed test cases for the "Content Versioning" feature, following your provided template:

****Test Case 1****

* ****Test Case ID:**** TC_001

* ****Test Case Title:**** Verify that instructors can create multiple versions of content.

* ****Test Case Description:**** This test case verifies the functionality of creating new versions of existing course content. As instructors can have multiple editing iterations, this test case will ensure they can save different drafts and versions for comparison and selection.

* ****Test Suite:**** Content Management

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

* ****Preconditions:****

- * A logged-in instructor account with course ownership.
- * An existing module or lesson within the course.

* **Test Data:** Updated content for saving as a new version (e.g., revised text, added resources).

* **Test Steps:**

1. Navigate to the course module or lesson.
2. Make modifications to the existing content.
3. Utilize the "Save Version" or equivalent functionality to create a new version of the content.
4. Verify that the saved version appears in the version history with a distinct identifier and timestamp.

* **Postconditions:**

- * The original content remains unchanged.
- * The saved new version is accessible for viewing and editing.

* **Expected Result:** A new version of the content is successfully created and saved, with its own unique identifier and version history entry.

* **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 version history is maintained and accessible.

* **Test Case Description:** This test case checks that the platform accurately records and presents a comprehensive history of content changes. It ensures instructors can easily navigate through different versions and understand the evolution of their course materials.

* **Test Suite:** Content Management

* **Test Priority:** High

****Preconditions:****

- * A content item with multiple versions created (from Test Case 1).
- * A logged-in instructor account.

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

****Test Steps:****

1. Navigate to the module or lesson with multiple saved versions.
2. Access the version history feature (e.g., "view history," "version control").
3. Iterate through the displayed versions and confirm that each version has:
 - * Correct timestamp and identifier.
 - * Accurate representation of the content at that specific point in time.

****Postconditions:****

- * The version history remains accessible and organized.

****Expected Result:**** All versions are accurately listed in the history, with clear timestamps and content representations for review and comparison.

****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 instructors can compare different versions of content.

****Test Case Description:**** This test case focuses on the functionality allowing comparison of different content versions. Instructors should be able to visualize side-by-side the changes made between versions for informed decision-making.

****Test Suite:**** Content Management

* **Test Priority:** High

* **Preconditions:**

- * Two distinct versions of a content item created (using Test Case 1).

- * A logged-in instructor account.

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

* **Test Steps:**

1. Access the content item's version history.
2. Select the two desired versions for comparison (e.g., Version 1 and Version 2).
3. Utilize the "Compare" or "Diff" functionality to initiate side-by-side comparison.
4. Verify that the changes between the two versions are clearly highlighted.

* **Postconditions:** No specific postconditions besides side-by-side view remaining available.

* **Expected Result:** A clear and visual representation of the differences between the selected content versions is displayed, highlighting added, removed, or modified content.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4****

* **Test Case ID:** TC_004

* **Test Case Title:** Verify that previous versions of content can be restored if needed.

* **Test Case Description:** This test case checks the functionality to revert a content item to a previous version. It ensures that instructors can seamlessly undo unwanted changes and maintain a stable version history.

* **Test Suite:** Content Management

* **Test Priority:** High

****Preconditions:****

- * A content item with multiple versions created (using Test Case 1).
- * A logged-in instructor account.

****Test Data:**** Changes made to the content item in a later version.

****Test Steps:****

1. Make modifications to the latest version of the content item (creating Version 3).
2. Access the version history.
3. Select Version 2 (the desired previous version) and utilize the "Restore" functionality.
4. Verify that the content reverts to the state it was in at the time of Version 2.

****Postconditions:****

- * The most recent version is replaced with the restored version.

****Expected Result:**** The content is successfully restored to the state of the selected previous version (Version 2), discarding the changes made in Version 3.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive, Destructive (restoring a version replaces previous work)

****Test Case 5****

****Test Case ID:**** TC_005

****Test Case Title:**** Verify that notifications are sent for significant content updates.

****Test Case Description:**** This test case verifies the notification system for content updates. Instructors who participate with the content should be notified about substantial changes made to it, ensuring timely awareness and visibility of revisions.

****Test Suite:**** User Notifications

****Test Priority:**** Medium

****Preconditions:****

- * A content item with at least two instructors who have subscribed or are enrolled.

****Test Data:****

- * Content updates that meet the criteria for triggering notifications (e.g., substantial revisions, addition of new materials).

****Test Steps:****

1. Make significant changes to an existing content item.
2. Verify that all subscribing instructors receive a notification regarding the update.

****Postconditions:****

- * The notification should include details about the changes made (e.g., content type, timestamp).

****Expected Result:**** Subscribing instructors receive notifications promptly about the significant content updates.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, UI 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:

Test Cases for Online Learning Platform - API Integration

****Test Case ID:**** TC-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 crucial platform functionalities such as user authentication, course enrollment, content retrieval, and quiz submission.

****Test Suite:**** API Integration

****Test Priority:**** High

****Preconditions:****

- * The API server is running.
- * An API testing tool is available.
- * A valid API key is obtained.

****Test Data:****

- * List of key platform functionalities: user authentication, course enrollment, content retrieval, quiz submission.
- * Example API endpoints for each functionality.

****Test Steps:****

1. Use the API testing tool to access the specified API endpoints for each functionality.
2. Send authentication requests using a valid API key.
3. Validate the response code for each request.
4. Examine the response data to ensure it contains the expected information.
5. Iterate through different functionalities and endpoints.

****Postconditions:****

- * API documentation is reviewed for any discrepancies.
- * Test reports are generated documenting the results.

****Expected Result:**** APIs for all key platform functionalities return expected response codes and data.

****Severity:**** Critical

****Priority:** High**

****Type of Testing:** API Testing, Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** TC-API-002**

****Test Case Title:** Verify that comprehensive API documentation is provided.**

****Test Case Description:** This test case verifies the completeness and clarity of the API documentation, ensuring developers have all the necessary information to integrate with the platform.**

****Test Suite:** API Integration**

****Test Priority:** High**

****Preconditions:****

* Access to the API documentation.

****Test Data:****

* List of key API endpoints and functionalities.

****Test Steps:****

1. Review the API documentation for general information about the platform's API.
2. Locate documentation for specific API endpoints for core functionalities (user authentication, course enrollment, content retrieval, quiz submission).
3. Verify that each endpoint description includes:
 - * HTTP method
 - * Request parameters
 - * Expected response codes

* Response data structure

4. Assess the overall clarity, organization, and comprehensibility of the documentation.

****Postconditions:****

* Feedback is provided on the API documentation for improvement.

****Expected Result:**** Comprehensive and well-structured API documentation is available, covering all essential endpoints and providing clear explanations of how to utilize them.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** API Testing, Exploratory Testing

****Test Case Approach:**** Positive

****Test Case ID:**** TC-API-003

****Test Case Title:**** Verify that secure access protocols are implemented.

****Test Case Description:**** This test case verifies the implementation of secure access protocols to protect API data and prevent unauthorized access.

****Test Suite:**** API Integration

****Test Priority:**** High

****Preconditions:****

* An API testing tool supporting security protocols is available.

****Test Data:****

* Examples of different authentication methods (API key, OAuth2)

****Test Steps:****

1. Attempt to access API endpoints without authentication.
2. Analyze the response code and message for unauthorized access indications.
3. Implement different authentication methods (e.g., API key, OAuth2) as specified in the documentation.
4. Verify successful authentication and access to protected endpoints.
5. Attempt to manipulate or decrypt API requests and responses to identify vulnerabilities.

****Postconditions:****

* Security considerations and findings are documented.

****Expected Result:**** Secure access protocols effectively prevent unauthorized access to API endpoints, and sensitive data is protected.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** API Testing, Security Testing

****Test Case Approach:**** Positive, Negative

****Test Case ID:**** TC-API-004

****Test Case Title:**** Verify that third-party applications can authenticate and access APIs.

****Test Case Description:**** This test case verifies the seamless authentication and access functionality for third-party applications utilizing the platform's API.

****Test Suite:**** API Integration

****Test Priority:**** High

****Preconditions:****

* A sample third-party application is integrated with the API.

****Test Data:****

* Credentials for accessing the third-party application.

****Test Steps:****

1. Set up the sample third-party application to interact with the platform's APIs.
2. Authenticate the third-party application using the API's designated authentication method.
3. Execute API requests relevant to the application's functionality (e.g., retrieving course data, enrolling users).
4. Validate the successful authentication and access to requested data.
5. Monitor API logs for any errors or discrepancies during the process.

****Postconditions:****

* Integration status and performance of the third-party application are documented.

****Expected Result:**** The third-party application successfully authenticates and retrieves data from the platform's API.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** API Testing, Integration Testing

****Test Case Approach:**** Positive

****Test Case ID:** TC-API-005**

****Test Case Title:** Verify that rate limiting and monitoring features are in place.**

****Test Case Description:** This test case verifies the implementation of rate limiting measures to prevent API overload and monitoring features to track API usage.**

****Test Suite:** API Integration**

****Test Priority:** Medium**

****Preconditions:****

* Access to API monitoring tools or logs.

****Test Data:****

* A script to make a high volume of API requests within a short timeframe.

****Test Steps:****

1. Execute the script to send a large number of API requests in a defined timeframe.
2. Monitor the API response for rate limiting indications (e.g., error codes, throttling delays).
3. Analyze API logs to identify any spikes in request volume or potential issues.
4. Verify that the rate limiting mechanisms effectively prevent excessive API usage.

****Postconditions:****

* API monitoring and rate limiting functionalities are documented based on test results.

****Expected Result:** The API implements effective rate limiting to prevent abuse and overloading, and dedicated monitoring tools provide insights into API usage patterns.**

****Severity:** Minor**

****Priority:**** Low

****Type of Testing:**** API Testing, Performance Testing

****Test Case Approach:**** Positive, Negative

Data - 72

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for

further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

Online Learning Platform - Data Analytics and Insights

Test Case Requirements:

- Generate at least 5 detailed test cases.
- Ensure that each test case thoroughly covers one or more of the acceptance criteria.
- Provide clear and concise descriptions for each test case.

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
---	---	---	---	---	---	---	---	---	---	---	---	---	---

TC_DA_001	Verify that data visualization tools are available for analysis.	This test verifies the availability and functionality of the data visualization tools within the platform's analytics section.	Data Analytics & Insights	High	Administrator login is successful.	Course data is available.	User has access to analytics dashboard.	1. Navigate to the analytics dashboard as an administrator. 2. Observe and interact with available data visualization components (e.g., charts, graphs, tables). 3. Verify that data visualization tools are functional and display relevant learner performance data.	None	Data visualization tools are accessible and display learner performance data correctly.	Critical	High	System Testing	Positive
TC_DA_002	Verify that reports can be generated for learner performance metrics.	This test checks the ability to generate reports on learner performance metrics from the platform's analytics section.	Data Analytics & Insights	High	Administrator login is successful.	Course data is available.	User has access to report generation tools.	1. Access the analytics dashboard as an administrator. 2. Select desired learner						

performance metrics for reporting (e.g., quiz scores, course completion rate).
3. Generate a report based on selected metrics.
4. Verify the report displays accurate and relevant data for the chosen metrics. | None | Reports are generated accurately and display relevant learner performance data. | Critical | High | System Testing | Positive |

| TC_DA_003 | Verify that real-time analytics provide insights into learner engagement and outcomes. | This test verifies the functionality of the real-time analytics feature for tracking learner engagement and outcomes. | Data Analytics & Insights | High | Administrators logged in. | Course with ongoing learners and activities. | 1. Access the real-time analytics dashboard.
2. Monitor live data for learner engagement metrics (e.g., active users, forum activity, resource downloads).
3. Observe real-time updates in learner outcome data (e.g., quiz completion, course progress). | None | Real-time analytics display accurate and timely data on learner engagement and outcomes. | Critical | High | System Testing | Positive |

| TC_DA_004 | Verify that data can be exported for further analysis. | This test checks the functionality of data export from the platform's analytics section for external analysis. | Data Analytics & Insights | High | Administrator login is successful. | Course data is available. User has access to export features. | 1. Access the analytics dashboard as an administrator.
2. Select desired data to export (e.g., learner performance, quiz scores).
3. Choose the desired export format (e.g., CSV, Excel).
4. Verify the exported data is accurate and can be opened in the selected format. | None | Data can be successfully exported in the chosen format, preserving its integrity. | Critical | High | System Testing | Positive |

| TC_DA_005 | Verify that notifications are sent for significant data trends or anomalies. | This test ensures the platform generates notifications for critical data trends or unusual learner behaviors. | Data Analytics & Insights | Medium | Administrator login is successful. | Course data with a predefined threshold for anomaly detection. | 1. Set up notifications for specific data thresholds (e.g., sudden drop in quiz scores).
2. Manually trigger a data event that exceeds the defined threshold (e.g., a significant number of low quiz scores).
3. Verify that the administrator receives a notification promptly. | None | Notifications are delivered instantaneously when a predefined data anomaly occurs. | Major | Medium | System Testing | Positive |

****Note:**** This is a starting point and can be expanded based on the specific requirements and complexity of the online learning platform.

****Responsibilities:****

These test cases will be utilized by the QA team to ensure the quality and functionality of the "Data Analytics and Insights" feature within the Online Learning Platform.

Data - 73

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Roles and Permissions

Feature description -

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

Userstory title -

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

Acceptance criteria -

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

Generated Test Cases:

Here are 5 detailed test cases for the "User Roles and Permissions" feature of the Online Learning Platform, following the provided template:

Test Case 1

****Test Case ID:** TC-URP-001**

****Test Case Title:**** Verify that an administrator can create a new user role.

****Test Case Description:**** This test case verifies the administrator's ability to create a new user role with a unique name and associated permissions.

****Test Suite:**** User Roles and Permissions

****Test Priority:**** High

****Preconditions:****

* An administrator account is logged in.

* At least one pre-existing role exists on the platform.

* A form for creating new roles is accessible.

****Test Data:****

* Unique role name: "Course Moderator"

* Desired permissions: View all course content, moderate discussions, assign grades

****Test Steps:****

1. Navigate to the user roles management section.
2. Click on the button to create a new role.
3. Enter the role name "Course Moderator".
4. Select the desired permissions for the role: "View all course content", "Moderate discussions", "Assign grades".
5. Click the "Create Role" button.

****Postconditions:****

* A new role named "Course Moderator" is listed in the user roles management section.

* The newly created role has the assigned permissions.

****Expected Result:**** The role "Course Moderator" is successfully created with the specified name and permissions.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

Test Case 2

****Test Case ID:**** TC-URP-002

****Test Case Title:**** Verify that permissions are customizable for each role.

****Test Case Description:**** This test case ensures that administrators can individually adjust permissions for each user role, granting or removing access to specific features.

****Test Suite:**** User Roles and Permissions

****Test Priority:**** High

****Preconditions:****

* An administrator account is logged in.

* The "Course Moderator" role created in TC-URP-001 exists.

****Test Data:****

* Permission to edit courses: Initially unchecked for "Course Moderator" role

****Test Steps:****

1. Navigate to the "Course Moderator" role in the user roles management section.
2. Locate the "Edit Courses" permission setting.
3. Check the "Edit Courses" permission box.
4. Save the changes to the role's permissions.

****Postconditions:****

* The "Course Moderator" role now has permission to edit courses.

****Expected Result:**** The capability to edit courses is added to the "Course Moderator" role's permissions.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

Test Case 3

****Test Case ID:**** TC-URP-003

****Test Case Title:**** Verify that users are restricted to functionalities based on their roles.

****Test Case Description:**** This test case ensures that users are prevented from accessing features they are not allowed to use according to their assigned roles.

****Test Suite:**** User Roles and Permissions

****Test Priority:**** High

****Preconditions:****

* A "Student" user account is created and assigned the "Student" role.

* A "Course Moderator" user account is created with the "Course Moderator" role.

****Test Data:****

* A course with available editing functionality.

****Test Steps:****

1. Log in as the "Student" user.
2. Attempt to access the course editing section.
3. Log in as the "Course Moderator" user.
4. Attempt to access the course editing section.

****Postconditions:****

* No further actions are needed.

****Expected Result:****

* The "Student" user is prevented from accessing the course editing section. The "Course Moderator" user can access and edit the course.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive and Negative**

Test Case 4

****Test Case ID:** TC-URP-004**

****Test Case Title:** Verify that changes to roles and permissions are logged.**

****Test Case Description:** This test case confirms that the system records all modifications made to user roles and permissions.**

****Test Suite:** User Roles and Permissions**

****Test Priority:** Medium**

****Preconditions:****

* An administrator account is logged in.

* A user role with existing permissions exists.

****Test Data:****

* User role: "Instructor"

* Permission modification: Adding the permission "Assign quizzes"

****Test Steps:****

1. Navigate to the "Instructor" role's settings.
2. Add the permission "Assign quizzes" to the role.
3. Save the changes to the role's permissions.
4. Access the system logs or audit trail.

****Postconditions:****

* The system logs should be accessible.

****Expected Result:****

* A log entry records the addition of the "Assign quizzes" permission to the "Instructor" role.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Security Testing

****Test Case Approach:**** Positive

Test Case 5

****Test Case ID:**** TC-URP-005

****Test Case Title:**** Verify that users are notified of their assigned roles and permissions.

****Test Case Description:**** This test case confirms that users receive information about their assigned roles and the permissions associated with those roles.

****Test Suite:**** User Roles and Permissions

****Test Priority:**** Medium

****Preconditions:****

* A new user account named "NewUser" is created.

* The "NewUser" account is assigned the "Student" role.

****Test Data:****

* User email: newuser@example.com

****Test Steps:****

1. Ensure new user account "NewUser" is properly configured.
2. Simulate a notification email sent to the "NewUser" account.
3. Check the email for information about the assigned "Student" role and its associated permissions.

****Postconditions:**

* The email notification is received.

****Expected Result:****

* The "NewUser" receives an email notification detailing their assigned roles (Student) and the permissions associated with that role.

****Severity:** Minor**

****Priority:** Medium**

****Type of Testing:** Functional Testing, Usability 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:

Online Learning Platform: User Feedback & Rating System Test Cases

****Test Case ID:**** TC_US_01

****Test Case Title:**** Verify that learners can rate courses and instructors on a scale.

****Test Case Description:**** This test case ensures learners can successfully rate courses and instructors using the provided rating scale and that the ratings are properly displayed.

****Test Suite:**** User Feedback and Rating System

****Test Priority:**** High

****Preconditions:****

- * A logged-in learner user.
- * An existing course with an instructor.

****Test Data:****

- * Rating values (e.g., 1-5 stars)

****Test Steps:****

1. Navigate to the course page.
2. Locate the course rating section and select a rating value.
3. Click on the 'Submit' button.
4. Verify that the selected rating is displayed next to the course name.
5. Repeat steps 2-4 for the instructor rating section.
6. Verify that selected instructor rating is displayed next to the instructor's name.

****Postconditions:****

- * Rating data is saved in the database.

****Expected Result:**** The learner's selected ratings are successfully saved and displayed correctly for both the

course and the instructor.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** TC_US_02**

****Test Case Title:** Verify that feedback forms are available for detailed comments.**

****Test Case Description:** This test case ensures learners can access and utilize feedback forms to provide detailed comments on courses, instructors, and learning materials.**

****Test Suite:** User Feedback and Rating System**

****Test Priority:** Medium**

****Preconditions:****

* A logged-in learner user.

* An existing course with an instructor.

****Test Data:****

* Sample feedback comment

****Test Steps:****

1. Navigate to the course page.
2. Locate the feedback form section.
3. Verify that the form includes fields for providing detailed comments.
4. Enter a sample feedback comment in the provided field.
5. Validate that other relevant fields like rating scales, topic selection, and category selection are accessible.

****Postconditions:****

* Feedback comment and other relevant data are saved in the database.

****Expected Result:**** Learners can access and utilize feedback forms, providing detailed written comments related to their course experience.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:** TC_US_03**

****Test Case Title:** Verify that instructors can view and respond to feedback.**

****Test Case Description:** This test case verifies that instructors can access and respond to learner feedback submitted through the platform's system.**

****Test Suite:** User Feedback and Rating System**

****Test Priority:** High**

****Preconditions:****

- * A logged-in instructor user.
- * Existing learner feedback

****Test Data:****

- * Sample feedback response

****Test Steps:****

1. Navigate to the instructor dashboard.
2. Locate the feedback section.
3. Access a previously submitted learner feedback.
4. Verify that the feedback content is displayed clearly.
5. Compose a sample response to the feedback in the designated field.

6. Submit the feedback response.

****Postconditions:****

* Feedback response is saved in the database and associated with the original feedback.

****Expected Result:**** Instructors can access, view, and respond to learner feedback effectively on the platform.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:**** TC_US_04

****Test Case Title:**** Verify that aggregated ratings are displayed publicly

****Test Case Description:**** This test case ensures that aggregated course and instructor ratings, calculated from multiple learner submissions, are publicly visible.

****Test Suite:**** User Feedback and Rating System

****Test Priority:**** Medium

****Preconditions:****

- * Multiple learners have rated both a course and its instructor.

****Test Data:****

- * N/A

****Test Steps:****

1. Navigate to the course page.
2. Verify that the aggregated course rating (e.g., average star rating) is displayed publicly.
3. Locate the instructor's profile page.
4. Verify that the aggregated instructor rating (e.g., average star rating) is displayed publicly.

****Postconditions:****

- * N/A

****Expected Result:**** Aggregated ratings for courses and instructors are clearly displayed on their respective pages, allowing potential learners to assess their overall quality and popularity.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:** Functional Testing**

****Test Case Approach:** Regression**

****Test Case ID:** TC_US_05**

****Test Case Title:** Verify that anonymous feedback options are available.**

****Test Case Description:** This test case ensures learners can choose to provide feedback anonymously, protecting their identity when expressing opinions.**

****Test Suite:** User Feedback and Rating System**

****Test Priority:** Medium**

****Preconditions:****

* A logged-in learner user.

****Test Data:****

* Sample feedback comment

****Test Steps:****

1. Navigate to the course page.
2. Locate the feedback form section.

3. Ensure an option is available to submit feedback anonymously.
4. Select the 'Anonymous' option.
5. Enter a sample feedback comment.
6. Submit the feedback.

****Postconditions:****

* Feedback is saved in the database with an anonymous identifier.

****Expected Result:**** Learners can choose to provide feedback anonymously, safeguarding their identity while contributing to the platform's improvement.

****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:

Online Learning Platform - Social Media Integration Test Cases

****User Story:**** As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

****Test Case 1:****

* **Test Case ID:** TC_SMI_001

* **Test Case Title:** Verify that learners can connect their social media accounts.

* **Test Case Description:** This test case verifies the functionality of connecting various social media accounts (Facebook, Twitter, LinkedIn) to the learning platform.

* **Test Suite:** Social Media Integration

* **Test Priority:** High

* **Preconditions:**

- * Learner account is created and logged in.

- * Social media accounts (Facebook, Twitter, LinkedIn) are accessible.

* **Test Data:** Sample social media account credentials (username and password)

* **Test Steps:**

1. Navigate to the "Settings" tab in the learner's profile.
2. Click on the "Social Media" section.
3. Select a social media platform (e.g., Facebook) from the available options.
4. Enter the social media account credentials.
5. Authorize the platform to access the social media account.

* **Postconditions:**

- * Social media account should be successfully connected to the learner's profile.

- * A confirmation message should be displayed confirming successful connection.

* **Expected Result:** The user's social media account should be successfully connected to their learner profile.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, User Acceptance Testing

* **Test Case Approach:** Positive

****Test Case 2:****

*** **Test Case ID:** TC_SMI_002**

*** **Test Case Title:** Verify that achievements are shared on social media.**

*** **Test Case Description:** This test case verifies that learners can share their accomplishments (certificate completion, course progress) on their connected social media accounts.**

*** **Test Suite:** Social Media Integration**

*** **Test Priority:** High**

*** **Preconditions:****

- * Learner account is created and logged in with a connected social media account (as tested in TC_SMI_001).

- * Learner has completed a course or achieved a specific milestone.

*** **Test Data:** Sample course completion data**

*** **Test Steps:****

1. Navigate to the "Achievements" section of the learner's profile.
2. Click on the "Share on Social Media" button for a specific achievement.
3. Select the desired social media platform (e.g., Facebook).
4. Review the pre-populated social media post with the achievement details.
5. Publish the post on the selected social media platform.

*** **Postconditions:****

- * The achievement should be successfully published on the learner's chosen social media platform.

- * Visual confirmation (e.g., "Post successful!", green tick) should be displayed on the platform.

*** **Expected Result:** The achievement should be shared on the selected social media platform as expected.**

*** **Severity:** Critical**

* **Priority:** High

* **Type of Testing:** Functional Testing, User Acceptance Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TC_SMI_003

* **Test Case Title:** Verify that privacy settings control the visibility of shared content.

* **Test Case Description:** This test case ensures learners can control the visibility of their social media shares based on privacy settings.

* **Test Suite:** Social Media Integration

* **Test Priority:** High

* **Preconditions:**

- * Learner account is created and logged in with a connected social media account.

- * A test achievement is completed.

* **Test Data:**

- * Different privacy settings (public, private, friends only).

* **Test Steps:**

1. Navigate to the "Settings" tab in the learner's profile>
2. Go to the "Social Media" section.
3. Modify the privacy setting for shared content to "Public".
4. Share the test achievement on social media.
5. Repeat steps 3-4 with privacy settings changed to "Private" and "Friends Only".

* **Postconditions:**

- * The shared content should be visible to appropriate audiences based on selected privacy settings.

* **Expected Result:** The achievement shared with "Public" setting should be visible to everyone.

- * The achievement shared with "Private" setting should only be visible to the learner.

- * The achievement shared with "Friends Only" should be visible only to the learner's social media friends.

- * **Severity:** Medium

- * **Priority:** Medium

- * **Type of Testing:** Functional Testing, User Acceptance Testing

- * **Test Case Approach:** Positive, Negative

****Test Case 4:****

- * **Test Case ID:** TC_SMI_004

- * **Test Case Title:** Verify that instructors can promote courses and updates through social media.

- * **Test Case Description:** This test case checks if instructors can share information about their courses and platform updates on social media.

- * **Test Suite:** Social Media Integration

- * **Test Priority:** Medium

- * **Preconditions:**

- * Instructor account is created and logged in.

- * An instructor has created a course and/or has platform updates to share.

- * **Test Data:**

- * Sample course information (name, description, link).

- * **Test Steps:**

1. Navigate to the instructor's course page.
2. Click on the "Share on Social Media" button for the course.
3. Select the desired social media platform (e.g., Twitter).
4. Review the pre-populated social media post with course information.
5. Publish the post on the selected social media platform.

6. For platform updates, navigate to the relevant update section (e.g., News or Announcements) and repeat steps 2-5.

* **Postconditions:**

* The shared course information or platform updates should be successfully published on the selected social media platform.

* **Expected Result:** The course or platform updates should be shared on the selected social media platform as expected.

* **Severity:** Medium

* **Priority:** Medium

* **Type of Testing:** Functional Testing, User Acceptance Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TC_SMI_005

* **Test Case Title:** Verify that analytics track the reach and engagement of shared content.

* **Test Case Description:** This test case validates the functionality of analytics reporting for social media content shared by learners and instructors.

* **Test Suite:** Social Media Integration

* **Test Priority:** Medium

* **Preconditions:**

* User account is created and logged in (either learner or instructor).

* User has shared content on at least one social media platform.

* **Test Data:**

* Limited sample data (number of shares, likes, comments, etc.) generated through manual interaction.

* **Test Steps:**

1. Navigate to the "Analytics" or "Social Media Reports" section of the platform.
2. Review the available data on shared content, including the number of shares, likes, comments, and reach.
3. Compare the reported data with the manually generated sample data. For example, count the number of likes manually on the shared post and compare it to the reported number.

* **Postconditions:**

* No specific postconditions needed.

* **Expected Result:** The analytics data should accurately reflect the reach and engagement of the shared content.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive, Negative (Check for inaccurate data or missing information).

****Please Note**:** These test cases are a starting point and can be further expanded with more specific scenarios and edge cases based on the platform's specific implementation and functionalities.

Data - 76

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Activity Tracking

Feature description -

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity levels and progress towards fitness goals.

Userstory title -

As a user, I want to track my activities so that I can monitor my daily physical activity levels.

Acceptance criteria -

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']

Generated Test Cases:

Fitness Tracker App: Activity Tracking Test Cases

****Test Case ID:**** ACTIVITY-001

****Test Case Title:**** Verify that the app records distance, duration, pace, and calories burned for running activities.

****Test Case Description:**** This test case verifies that the app accurately captures and displays data for running activities, including distance, duration, pace, and calories burned.

****Test Suite:**** Activity Tracking

****Test Priority:** High**

****Preconditions:****

- * User has created an account.
- * User has allowed the app to access location services.
- * User has a wearable device (optional but recommended) syncing with the app.

****Test Data:**** Simulated running activity data with varying distances, durations, paces, and heart rates.

****Test Steps:****

1. Start a running activity within the app.
2. Run for a predetermined distance and duration, maintaining a consistent pace.
3. Stop the running activity in the app.
4. View the activity summary screen.
5. Verify that the recorded distance, duration, average pace, and calories burned are accurate.

****Postconditions:****

- * Activity is saved in the user's activity log.
- * Notifications (if configured) are sent for milestones reached during the run.

****Expected Result:**** The app accurately records and displays all relevant data for the running activity.

****Severity:** Critical**

****Priority:** High**

****Test Case Approach:** Positive, Valid**

****Test Case ID:** ACTIVITY-002**

****Test Case Title:** Verify that users can view activity summaries for different activities.**

****Test Case Description:**** This test case verifies that users can access and view detailed summaries for various activities tracked by the app, including running, walking, cycling, and swimming.

****Test Suite:**** Activity Tracking

****Test Priority:**** High

****Preconditions:****

- * User has created an account.

- * User has performed activities for different activity types.

****Test Data:**** Recorded activity data for running, walking, cycling, and swimming.

****Test Steps:****

1. Navigate to the "Activity Log" section of the app.
2. Select a specific activity type (e.g., running).
3. View the summary of recorded data for that activity type.
4. Repeat steps 2-3 for different activity types.

****Postconditions:****

- * Activity summaries are displayed for each selected activity type.

- * Filters and sorting options (if available) function as expected.

****Expected Result:**** Users can access and view comprehensive activity summaries for each tracked activity type.

****Severity:**** Major

****Priority:**** High

****Test Case Approach:**** Positive, Valid

****Test Case ID:**** ACTIVITY-003

****Test Case Title:**** Verify that activity data is stored for historical analysis

****Test Case Description:**** This test case verifies the app's ability to store activity data over time, allowing users to review past activities and performance trends.

****Test Suite:**** Activity Tracking

****Test Priority:**** Medium

****Preconditions:****

* User has created an account.

* User has recorded multiple activities over several days.

****Test Data:**** Multiple activity recordings spanning several days.

****Test Steps:****

1. Navigate to the "Activity Log" section and select a specific date range from the past.
2. Verify that activities recorded within the selected date range are displayed.
3. Check that activity data for each day within the range is accurate.

****Postconditions:****

* Historical activity data is accessible and displays properly.

****Expected Result:**** The app effectively stores and retrieves past activity data for historical analysis.

****Severity:**** Major

****Priority:**** Medium

****Test Case Approach:**** Positive, Valid

****Test Case ID:**** ACTIVITY-004

****Test Case Title:**** Verify that activity tracking can be paused and resumed

****Test Case Description:**** This test case verifies the functionality of pausing and resuming an activity track to accommodate breaks or interruptions during exercise.

****Test Suite:**** Activity Tracking

****Test Priority:**** Medium

****Preconditions:****

* User has created an account.

* User is recording an activity.

****Test Data:**** N/A

****Test Steps:****

1. Start a running activity within the app.
2. Allow the activity to record for a short duration.
3. Pause the activity tracking within the app.
4. Wait for a specified period.
5. Resume the activity tracking within the app.
6. Allow the activity to continue for a short duration.

****Postconditions:****

* The app correctly resumes tracking activity data from where it paused.

****Expected Result:**** Activity tracking seamlessly pauses and resumes, capturing accurate data throughout the exercise session.

****Severity:**** Minor

****Priority:** Medium**

****Test Case Approach:** Positive, Edge Case**

****Test Case ID:** ACTIVITY-005**

****Test Case Title:** Verify that users receive notifications for activity milestones.**

****Test Case Description:** This test case verifies the functionality of receiving notifications when users achieve specific milestones during their activities, as configured by the user.**

****Test Suite:** Activity Tracking**

****Test Priority:** Medium**

****Preconditions:****

- * User has created an account.
- * User has configured notification settings for activity milestones (e.g., completing a certain distance, reaching a desired heart rate zone).

****Test Data:****

Simulated activity data to reach predefined milestones.

****Test Steps:****

1. Start an activity (e.g., running) with notifications enabled for a specific milestone.
2. Follow the activity session to reach the configured milestone.
3. Observe the user interface and device notifications.

****Postconditions:****

- * Activities have completed updating in the activity log.
- * No unexpected notifications are triggered.

****Expected Result:**** The user receives timely notifications upon reaching configured activity milestones.

****Severity:**** Minor

****Priority:**** Medium

****Test Case Approach:**** Positive, Functional

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:

Test Cases for Fitness Tracker App - GPS Tracking Feature

****User Story Title:**** As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

****Test Case ID:**** FIT-GPS-001

****Test Case Title:**** Verify that GPS Tracking records accurate route data.

****Test Case Description:**** This test case aims to verify that the GPS tracking feature accurately records the user's route during an outdoor activity.

****Test Suite:**** GPS Tracking

****Test Priority:**** High

****Preconditions:****

- User has enabled location services on their device.
- App has access to the device's GPS functionality.

- The user is in an open area with adequate GPS signal.

****Test Data:**** No test data needed.

****Test Steps:****

1. Launch the Fitness Tracker App.
2. Start a new outdoor workout session.
3. Engage in a predefined walking/running route for at least 15 minutes.
4. Pause the workout session after the predefined route is completed.
5. After pausing the session, review the recorded route on the workout summary screen.

****Postconditions:****

* The recorded route visually matches the actual path taken by the user.

****Expected Result:**** The recorded route accurately reflects the user's movement path during the outdoor activity.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** System Testing, Functional Testing

****Test Case ID:**** FIT-GPS-002

****Test Case Title:**** Verify that users can view detailed maps and elevation profiles.

****Test Case Description:**** This test case verifies the functionality of displaying detailed route maps and elevation profiles to users while visualizing their workout data.

****Test Suite:**** GPS Tracking

****Test Priority:**** High

****Preconditions:****

- The user has completed an outdoor workout session with GPS tracking enabled.

****Test Data:**** No test data needed.

****Test Steps:****

1. Launch the Fitness Tracker App.
2. Access the workout summary screen for the completed outdoor activity.
3. Tap on the button/option to view the route map.
4. Verify that the map displays accurate representation of the recorded route with relevant markers.
5. Check the option for elevation profile visibility and confirm its accuracy.

****Postconditions:****

* The user can zoom and pan the map to explore details of the route.

****Expected Result:**** The app successfully displays a detailed map with clear visualization of the route, including elevation profiles, allowing users to analyze their workout performance.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** System Testing, Functional Testing

****Test Case ID:**** FIT-GPS-003

****Test Case Title:**** Verify that performance data includes location-based metrics.

****Test Case Description:**** This test case validates the inclusion of relevant location-based metrics in the workout performance data, such as distance traveled, average speed, and elevation gain/loss.

****Test Suite:**** GPS Tracking

****Test Priority:**** High

****Preconditions:****

- The user has completed a workout session using GPS tracking.

****Test Data:**** No test data needed.

****Test Steps:****

1. Launch the Fitness Tracker App.
2. Navigate to the workout summary screen for the completed GPS-tracked workout.
3. Examine the performance data section and verify the presence of location-based metrics.
4. Compare the recorded distances with a manually measured distance for validation.

****Postconditions:****

* No additional steps required.

****Expected Result:**** The app accurately records and displays location-based metrics as part of the workout performance data.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** System Testing, Functional Testing

****Test Case ID:**** FIT-GPS-004

****Test Case Title:**** Verify that routes can be saved and revisited.

****Test Case Description:**** This test case confirms the ability to save completed workout routes and revisit them later for analysis or replaying.

****Test Suite:**** GPS Tracking

****Test Priority:**** Medium

****Preconditions:****

- The user has completed a notable outdoor workout session.

****Test Data:**** No test data needed.

****Test Steps:****

1. Launch the Fitness Tracker App.
2. Access the completed workout for the session with GPS tracking.
3. Check the option to save the route as a favorite or bookmark.
4. Navigate to the saved routes or history section of the app.
5. Select the saved route and verify its details are displayed correctly.
6. Replay the saved route visualization and confirm its accuracy.

****Postconditions:****

* No additional steps required.

****Expected Result:**** The app successfully saves and allows for easy access and replay of previously recorded routes.

****Severity:**** Major

****Priority:** Medium**

****Type of Testing:** System Testing, Functional Testing**

****Test Case ID:** FIT-GPS-005**

****Test Case Title:** Verify that Notifications are provided for deviations from planned routes.**

****Test Case Description:** This test case examines the notification system's functionality when a user deviates from a planned route during an outdoor activity.**

****Test Suite:** GPS Tracking**

****Test Priority:** Medium**

****Preconditions:****

- The user has set a planned route within the App.
- Location services are enabled on the device.

****Test Data:** No test data needed.**

****Test Steps:****

1. Launch the Fitness Tracker App.
2. Plan a specific route within the app for an outdoor activity.

3. Start the workout session with the planned route selected.
4. Intentionally deviate from the planned route during the workout session.
5. Observe whether any notifications are received regarding the deviation from the planned route.

****Postconditions:****

- Review app notifications history to verify the notification logs.

****Expected Result:**** The app successfully sends timely notifications when the user deviates from the planned route, ensuring awareness and course correction.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** System Testing, Functional Testing, Regression Testing

Data - 78

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Heart Rate Monitoring

Feature description -

The app monitors heart rate in real-time using built-in sensors or compatible wearable devices, providing insights into exercise intensity, recovery, and cardiovascular health. Heart rate data helps users optimize their workouts, maintain target heart rate zones, and track improvements in fitness over time.

Userstory title -

As a user, I want to monitor my heart rate so that I can optimize my workouts and track my cardiovascular health.

Acceptance criteria -

['Heart rate data is recorded in real-time.', 'Users can set and monitor target heart rate zones.', 'Historical heart rate data is stored and viewable.', 'Alerts are provided for abnormal heart rate readings.', 'Data is compatible with various wearable devices.']

Generated Test Cases:

Fitness Tracker App - Heart Rate Monitoring Test Cases

****Test Case ID:**** HR-001

****Test Case Title:**** Verify that heart rate is recorded in real-time during exercise.

****Test Case Description:**** This test case will verify that the app accurately captures and displays real-time heart rate data while the user is engaged in physical activity.

****Test Suite:**** Heart Rate Monitoring

****Test Priority:** High**

****Preconditions:****

* User has a compatible wearable device or the built-in sensor is enabled.

* User has permission to access health data on their device.

****Test Data:** N/A**

****Test Steps:****

1. Start a workout session within the app.
2. Begin exercising at a moderate intensity.
3. Observe the real-time heart rate display in the app.
4. Verify that the heart rate value updates dynamically and accurately reflects the user's current heart rate.
5. Stop the workout session and note the final recorded heart rate.

****Postconditions:****

* Workout session is complete.

****Expected Result:**** Real-time heart rate data is displayed accurately and dynamically throughout the exercise session.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing, System Testing**

****Test Case Approach:** Positive**

****Test Case ID:**** HR-002

****Test Case Title:**** Verify that users can set and monitor target heart rate zones.

****Test Case Description:**** This test case will ensure that users can configure their target heart rate zones and that the app accurately monitors their heart rate within those zones.

****Test Suite:**** Heart Rate Monitoring

****Test Priority:**** High

****Preconditions:****

* User logged into the app.

****Test Data:**** User-defined target heart rate zones (e.g., low, moderate, high).

****Test Steps:****

1. Navigate to the "Settings" section of the app.
2. Access the "Heart Rate Zones" settings.
3. Define user-specific target heart rate zones for different activity intensities.
4. Start an exercise session.
5. Observe the heart rate display and verify that the app indicates which target heart rate zone the user is currently in.

****Postconditions:****

* Target heart rate zones are saved in the user profile.

****Expected Result:**** Users can easily set and monitor their target heart rate zones. The app accurately depicts

their heart rate against their defined zones during exercise.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, System Testing

****Test Case Approach:**** Positive

Let me know if you want me to continue generating these test cases. I can create more that cover:

- * Historical heart rate data storage and retrieval.
- * Alerts for abnormal heart rate readings.
- * Data compatibility with various wearable devices.

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:

Fitness Tracker App - Calorie Tracking Test Cases

Here are 5 detailed test cases for the "Calorie Tracking" feature of the Fitness Tracker App, based on the provided acceptance criteria:

****Test Case ID:** TC_CAL_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 correctly input food items and their**

corresponding calorie amounts into the app's calorie logging interface.

****Test Suite:**** Calorie Tracking Feature Testing

****Test Priority:**** High

****Preconditions:****

- * A user account is created and logged in.
- * The user is on the "Food Log" page within the app.

****Test Data:****

- * A list of common food items with known calorie values (e.g., apple, banana, chicken breast, rice, etc.).

****Test Steps:****

1. Select "Add Food Item" button.
2. Enter a food item name from the provided data list into the search bar.
3. Select the desired food item from the search results.
4. Enter the quantity of the food item consumed (e.g., 1 apple).
5. Verify that the calorie value for the specified quantity is accurately displayed.

****Postconditions:****

- * The logged food item and its calorie value are displayed in the user's daily food log.

****Expected Result:****

The app successfully logs the user's food intake and displays the total calorie consumption for the logged items.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** TC_CAL_02**

****Test Case Title:** Verify that the app calculates and displays total calories consumed and burned.**

****Test Case Description:** This test case verifies the accuracy of the app's calorie calculation by comparing total calories consumed against calories burned from logged activities.**

****Test Suite:** Calorie Tracking Feature Testing**

****Test Priority:** High**

****Preconditions:****

- * A user account is created and logged in.
- * The user has logged both food intake and completed an activity (e.g., running) with recorded calorie expenditure.

****Test Data:****

- * Food items with specific calorie values.
- * Activity data with recorded calorie expenditure.

****Test Steps:****

1. Log various food items with known calorie values using Test Case TC_CAL_01 steps.
2. Log a completed physical activity with recorded calorie burn.
3. Navigate to the "Calorie Tracking" page or section within the app.
4. Verify that the displayed total calories consumed accurately reflects the sum of all logged food items.
5. Next, verify that the total calories burned from logged activities are accurately displayed.

****Postconditions:****

- * The user's total daily calorie consumption and calorie expenditure are displayed.

****Expected Result:****

The app accurately calculates and displays the total calories consumed and burned, providing a summary of the user's daily energy balance.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

Continue creating test cases for the remaining acceptance criteria in a similar format.

Data - 80

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

Fitness Tracker App: Sleep Tracking Test Cases

****Test Case ID:**** FT-STP-001

****Test Case Title:**** Verify that sleep data is recorded correctly, including duration and quality.

****Test Case Description:**** This test case verifies that the app accurately records sleep duration and quality based on sensor data, ensuring data integrity for sleep analysis and insights.

****Test Suite:**** Sleep Tracking

****Test Priority:**** High

****Preconditions:****

* The user has enabled sleep tracking in the app settings.

* The user has set their preferred sleep schedule in the app.

* A wearable device or fitness tracker is connected to the app and recording sleep data.

****Test Data:****

* Simulated sleep data with varying durations and sleep quality scores.

****Test Steps:****

1. Start the sleep tracking functionality at the user-specified bedtime.
2. Simulate sleep data using a wearable device or collected mock data, including periods of light, deep, and REM sleep.
3. End the sleep tracking session at the user-specified wake-up time.
4. Access the sleep tracking summary.
5. Verify that the recorded sleep duration matches the simulated data.
6. Confirm that the reported sleep quality score reflects the simulated sleep patterns.

****Postconditions:****

- * Sleep tracking data is synced with the app database.

****Expected Result:****

The app accurately records the simulated sleep duration and quality, displaying them in the sleep tracking summary.

****Severity:** Blocker**

****Priority:** High**

****Type of Testing:** Integration Testing, Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** FT-STP-002**

****Test Case Title:** Verify that users can view detailed sleep stages and patterns.**

****Test Case Description:**** This test case confirms the app's ability to present detailed sleep stage information, including the duration and percentage of each stage, aiding users in understanding their sleep

patterns.

****Test Suite:**** Sleep Tracking

****Test Priority:**** High

****Preconditions:****

* Sleep tracking is enabled and a successful sleep session recording is available.

* The user has logged at least one full night of sleep.

****Test Data:****

* Recorded sleep data with distinct sleep stage durations (e.g., light, deep, REM).

****Test Steps:****

1. Navigate to the sleep tracking section of the app.
2. Select the desired sleep session from the list.
3. Verify the visual representation of sleep stages, such as a pie chart or segmented bar graph.
4. Confirm that the duration of each sleep stage is displayed accurately (e.g., hours and minutes).
5. Check if the percentage breakdown of each sleep stage is also presented correctly.

****Postconditions:****

* Sleep stage details are visible within the selected sleep session.

****Expected Result:****

The app displays a clear and comprehensive breakdown of sleep stages for the chosen sleep session, including duration and percentage.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Usability Testing

****Test Case Approach:**** Positive

(Add 3 more test cases following the same template considering the remaining acceptance criteria)

Data - 81

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Logging

Feature description -

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

Userstory title -

As a user, I want to log my workouts so that I can track my progress and improve my performance.

Acceptance criteria -

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.',

'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

Generated Test Cases:

Test Cases for Workout Logging Feature

Project: Fitness Tracker App

Feature: Workout Logging

User Story: 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.

Test Case 1:

Test Case ID: TC_WL_01

Test Case Title: Verify that users can log various workout types.

****Test Case Description:**** This test case verifies that the user can successfully log workouts of different types, including cardio, strength training, yoga, etc., into the app.

****Test Suite:**** Workout Logging

****Test Priority:**** High

****Preconditions:****

* User is logged into the app.

* The app is installed and running on a compatible device.

****Test Data:****

* Different workout types (e.g., Running, Weightlifting, Yoga, Swimming).

****Test Steps:****

1. Navigate to the "Workout Log" section in the app.
2. Tap on the "Add Workout" button.
3. Select a specific workout type from the list (e.g., "Running").
4. Enter workout details such as duration, intensity, and notes.
5. Save the workout entry.
6. Verify that the workout is correctly displayed in the Workout Log with the selected type and entered details.

****Postconditions:****

* The added workout should remain visible in the log.

****Expected Result:**** The app successfully logs the chosen workout type with the provided details.

****Severity:**** High

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

******(similarly structure the remaining test cases by considering the remaining acceptance criteria)******

****Test Case 2:****

****Test Case ID:** TC_WL_02**

*** **Test Case Title:** Verify that users can edit and delete workout entries.**

****? (Follow the same template as above, ensuring to cover all acceptance criteria including filtering by date and workout type.)****

****Remember:****

- * Each test case should be independent and focused on a single aspect of functionality.
- * Describe test steps in a clear and concise manner, leaving no room for ambiguity.
- * Use specific and measurable values for expected results (e.g., "duration should be 30 minutes" instead of "duration should be correct").
- * Perform thorough testing across different devices and platforms to ensure cross-platform compatibility and usability.

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:

Test Cases for Fitness Tracker App: Goal Setting Feature

****Test Case ID:** FT-GOAL-001**

****Test Case Title:** Verify that users can set goals for different fitness metrics.**

****Test Case Description:** This test case verifies the ability for users to set individual goals for various fitness metrics provided by the app, such as steps taken, calories burned, distance traveled, and heart rate.**

****Test Suite:** Fitness Tracker App - Goal Setting**

****Test Priority:** High**

****Preconditions:****

- User has successfully logged into the app.
- User is on the goal setting screen.

****Test Data:****

- Valid step count goal (e.g., 10,000)
- Valid calorie burn goal (e.g., 500)
- Valid distance goal (e.g., 5 km)
- Valid heart rate goal (e.g., 120 beats per minute during workout)

****Test Steps:****

1. Navigate to the goal setting section within the app.
2. Select the "Steps" metric and enter a valid goal value (e.g., 10,000).
3. Select the "Calories Burned" metric and enter a valid goal value (e.g., 500).
4. Select the "Distance" metric and enter a valid goal value (e.g., 5 km).
5. Select the "Heart Rate" metric and enter a valid goal value (e.g., 120 bpm during workout).
6. Save the changes.

****Postconditions:****

- All four chosen goals should be saved and visible in the user's profile.
- The app displays a success message or confirmation upon saving the goals.

****Expected Result:****

The system successfully saves user-defined goals for all chosen fitness metrics.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

****Test Case ID:**** FT-GOAL-002

****Test Case Title:**** Verify that goal progress tracking is available for each goal.

****Test Case Description:**** This test case verifies that the app accurately tracks progress towards each set goal

and displays it to the user.

****Test Suite:**** Fitness Tracker App - Goal Setting

****Test Priority:**** High

****Preconditions:****

- User has successfully logged into the app.
- User has set at least one goal (described in FT-GOAL-001).
- The user has performed activities corresponding to the set goals.

****Test Data:****

- Data recording from user's activity over a specific time period.

****Test Steps:****

1. Log in to the app and access the goal tracking section.
2. View the progress indicator for the "Steps" goal and verify it reflects the total steps taken based on recorded data.
3. View the progress indicator for the "Calories Burned" goal and verify it reflects the total calories burned based on recorded data.
4. View the progress indicator for the "Distance" goal and verify it reflects the total distance covered based on recorded data.
5. View the progress indicator for the "Heart Rate" goal and verify it reflects the average heart rate during workouts based on recorded data.

****Postconditions:****

- App should display accurate progress information for each goal.

****Expected Result:**** The system accurately tracks and displays the progress made towards all set goals.

****Severity:**** Major

****Priority:** High**

****Type of Testing:** Functional Testing, User Validation Testing**

****Test Case ID:** FT-GOAL-003**

****Test Case Title:** Verify that users receive motivational feedback and reminders.**

****Test Case Description:** This test case evaluates the functionality of the app's motivational feedback and reminder system for set goals.**

****Test Suite:** Fitness Tracker App - Goal Setting**

****Test Priority:** High**

****Preconditions:****

- User has successfully logged into the app.
- User has set at least one goal.
- Targeted date or time for reminder is set.

****Test Data:****

- Goals with specific deadlines or target dates.

****Test Steps:****

1. Set a goal with a target date or time in the future (e.g., "Run 5km by next Saturday").
2. Ensure the app has access to user's location settings for location-based reminders.
3. The user may need to trigger a manual refresh or restart the app on the designated date/time.
4. Observe if the app sends a motivational message or reminder about the goal.

****Postconditions:****

- The app confirms the sending of the reminder or feedback message.

****Expected Result:****

The user receives motivational feedback or reminders according to the set reminders schedule, promoting engagement and goal adherence.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Validation Testing

****Test Case ID:**** FT-GOAL-004

****Test Case Title:**** Verify that goals can be adjusted or updated as needed.

****Test Case Description:**** This test case verifies the user's ability to modify existing goals either by changing the target value or deadline.

****Test Suite:**** Fitness Tracker App - Goal Setting

****Test Priority:**** Medium

****Preconditions:****

- User has successfully logged into the app.
- User has set at least one goal (described in FT-GOAL-001).

****Test Data:****

- Modified goal values (e.g., increase step count goal, reduce calorie burn goal).
- New target dates for existing goals.

****Test Steps:****

1. Access the list of set goals within the app.
2. Select the goal you want to modify (e.g., Steps).
3. Change the target value for the selected goal (e.g., increase from 10,000 to 12,000 steps).

4. Save the changes.
5. Access the targeted goal page and verify the updated value.

****Postconditions:****

- The app confirms the successful update of the goal.
- If the target date is changed, the progress tracking should reflect the new deadline.

****Expected Result:**** The system successfully allows users to adjust existing goal values and deadlines.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Usability Testing

****Test Case ID:**** FT-GOAL-005

****Test Case Title:**** Verify that the app provides a summary of goal attainment.

****Test Case Description:**** This test case confirms the ability of the app to display a detailed summary of the user's progress against set goals.

****Test Suite:**** Fitness Tracker App - Goal Setting

****Test Priority:**** Medium

****Preconditions:****

- User has successfully logged into the app.
- User has set at least one goal and achieved a specific amount of progress towards it.

****Test Data:****

- Recorded activity data reflecting progress toward at least one goal.

****Test Steps:****

1. Access the goal summary section within the app.

2. Review the summary for each set goal.

3. Verify the summary includes:

- Current progress towards the goal (e.g., percentage complete)
- Number of days or weeks of progress achieved.
- Historical data on goal attainment (optional, but beneficial)

****Postconditions:****

- No errors are displayed when accessing the goal summary

****Expected Result:****

The app provides a clear and comprehensive summary of the user's progress against each set goal.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, User Validation Testing

Data - 83

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Plans

Feature description -

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

Userstory title -

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

Acceptance criteria -

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

Generated Test Cases:

Fitness Tracker App - Workout Plans Test Cases

****User Story:**** As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

****Test Case 1****

* **Test Case ID:** FTR-WP-001

* **Test Case Title:** Verify that users can select workout plans based on fitness levels.

* **Test Case Description:** This test case verifies if the app allows user selection of workout plans categorized by different fitness levels (beginner, intermediate, advanced)

* **Test Suite:** Functionality - Workout Plans

* **Test Priority:** High

* **Preconditions:**

- * User has a registered account.

- * User is logged into the app.

* **Test Data:** N/A

* **Test Steps:**

1. Navigate to the "Workout Plans" section of the app.
2. Observe the available plan categories (e.g., Beginner, Intermediate, Advanced).
3. Select a plan from each category.
4. Verify that the selected plans display relevant information based on the chosen fitness level.

* **Postconditions:**

- * User remains on the "Workout Plans" section.

* **Expected Result:** The app should display workout plans categorized by fitness levels and allow users to select plans accordingly.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** FTR-WP-002

* **Test Case Title:** Verify that plans include detailed exercise routines and schedules.

* **Test Case Description:** This test case verifies if the selected workout plan includes detailed information about exercises, proper form, sets, reps, rest periods, and daily/weekly schedules.

* **Test Suite:** Functionality - Workout Plans

* **Test Priority:** High

* **Preconditions:**

- * User has a registered account.

- * User is logged into the app.

* **Test Data:** A selected workout plan (Beginner, Intermediate, Advanced).

* **Test Steps:**

1. Navigate to the "My Plans" section of the app.

2. Select the chosen workout plan.

3. Review the plan details, including exercise descriptions, sets, reps, rest periods, and the weekly/daily schedule.

* **Postconditions:**

- * User remains on the "Workout Plan Details" screen.

* **Expected Result:** The plan should provide comprehensive details about each exercise, including images or videos demonstrating proper form. Schedules should be clearly defined with daily or weekly workouts.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive, Exploratory

Test Case 3

* **Test Case ID:** FTR-WP-003

* **Test Case Title:** Verify that users can track adherence and progress within the plan.

* **Test Case Description:** This test case verifies the app's ability to track user adherence to the workout plan and display progress metrics.

* **Test Suite:** Functionality - Workout Plans

* **Test Priority:** High

* **Preconditions:**

- * User has a registered account.

- * User is logged into the app.

* **Test Data:** Selected workout plan (Beginner, Intermediate, Advanced).

* **Test Steps:**

1. Navigate to the "My Plans" section of the app.

2. Start following the chosen workout plan.

3. Complete multiple workouts within the plan period.

4. Check the "Journal" section to verify workout completion records.

5. Review the "Progress" section to observe tracked metrics (e.g., workouts completed, weight lifted, calories burned).

* **Postconditions:**

- * User remains on the "My Plans" or "Progress" section.

* **Expected Result:** The app should accurately record workout completion and display relevant progress metrics within the chosen plan.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4****

*** **Test Case ID:** FTR-WP-004**

*** **Test Case Title:** Verify that the app provides notifications and reminders for workouts.**

*** **Test Case Description:** This test case confirms the functionality of workout reminders and notifications.**

*** **Test Suite:** Functionality - Workout Plans**

*** **Test Priority:** Medium**

*** **Preconditions:****

- * User has a registered account.

- * User is logged into the app.

- * Workout reminders are enabled in app settings.

*** **Test Data:** A chosen workout plan with upcoming workouts scheduled.**

*** **Test Steps:****

1. Navigate to the "Workout Plans" section and select a workout plan.
2. Enable notification settings for workout reminders within the plan.
3. Wait for the scheduled workout time.
4. Confirm that a notification is received, reminding the user about the upcoming workout.

*** **Postconditions:****

- * User can dismiss the notification.

*** **Expected Result:** The app should deliver timely notifications and reminders for scheduled workouts.**

*** **Severity:** Major**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing**

*** **Test Case Approach:** Positive**

**** Test Case 5****

* **Test Case ID:** FTR-WP-005

* **Test Case Title:** Verify that users can adjust plans based on progress and preferences.

* **Test Case Description:** This test case verifies the user's ability to modify workout plan settings based on their progress and evolving fitness goals.

* **Test Suite:** Functionality - Workout Plans

* **Test Priority:** Medium

* **Preconditions:**

- * User has a registered account.

- * User is logged into the app.

- * User has been following a workout plan.

* **Test Data:** A chosen workout plan with some progress recorded.

* **Test Steps:**

1. Navigate to the "My Plans" section of the app.

2. Select the ongoing workout plan.

3. Identify plan options for customization (e.g., adjusting workout frequency, intensity level, or swapping exercises).

4. Make adjustments to the plan settings based on user preferences and progress.

5. Verify that the updated plan reflects the chosen modifications.

* **Postconditions:**

- * User remains on the "My Plans" section with the adjusted plan details.

* **Expected Result:** The app should allow users to customize and manage their workout plans according to their needs and progress.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 84

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Community Support

Feature description -

Users can connect with a community of like-minded individuals, share progress updates, participate in challenges, and offer support and encouragement to fellow users. Community support fosters accountability, motivation, and a sense of camaraderie among users striving towards similar fitness goals.

Userstory title -

As a user, I want to connect with a community so that I can share my progress and stay motivated.

Acceptance criteria -

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.', 'The app facilitates participation in community challenges.', 'Users can offer and receive support and encouragement.', 'Community activity and engagement are tracked.']

Generated Test Cases:

Fitness Tracker App: Community Support Test Cases

Here are 5 detailed test cases for the "Community Support" feature of the Fitness Tracker App:

Test Case 1:

* **Test Case ID:** TC_CS_001

* **Test Case Title:** Verify that users can join and participate in community groups.

* **Test Case Description:** This test case verifies the functionality of joining and navigating within community groups.

* **Test Suite:** Community Support

* **Test Priority:** High

* **Preconditions:**

- * A user has created an account and logged into the app.
- * There are pre-existing community groups within the app.

* **Test Data:** List of community group names and descriptions

* **Test Steps:**

1. Navigate to the "Community" section of the app.
2. Explore the list of available community groups.
3. Select a community group and view its details (description, members, recent activity).
4. Tap the "Join Group" button for the selected group.
5. Confirm the join request within the app.
6. Verify that the user is now a member of the selected group and able to see relevant content.

* **Postconditions:**

- * The user remains a member of the selected group.
- * The user can access group-specific content and features.

* **Expected Result:** The user successfully joins the chosen community group and can engage with its content and members.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive Testing

Test Case 2:

* **Test Case ID:** TC_CS_002

* **Test Case Title:** Verify that users can share progress updates and achievements.

* **Test Case Description:** This test case ensures the functionality for users to post updates and achievements within community groups.

* **Test Suite:** Community Support

* **Test Priority:** High

* **Preconditions:**

- * A user is logged into the app and a member of at least one community group.

* **Test Data:** Sample progress updates and achievements (e.g., completed workout, personal record achieved)

* **Test Steps:**

1. Navigate to the chosen community group.
2. Tap on the "Post Update" or "Share" button within the group.
3. Compose a progress update or achievement post using the in-app editor.
4. Upload a relevant image or video (Optional).
5. Select the community group to share the post with.

6. Tap the "Post" or "Share" button to publish the update.

* **Postconditions:**

* The user's post appears in the community group feed.

* **Expected Result:** The user's progress update or achievement is successfully published in the specified community group, visible to other members.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive Testing

Test Case 3:

* **Test Case ID:** TC_CS_003

* **Test Case Title:** Verify that the app facilitates participation in community challenges.

* **Test Case Description:** This test case verifies the ability to join, track progress, and participate in community challenges.

* **Test Suite:** Community Support

* **Test Priority:** High

* **Preconditions:**

* A user is logged into the app and a member of a community group.

* A community challenge is active within the chosen group.

* **Test Data:** Details of an active community challenge (e.g., duration, goals, rewards)

* **Test Steps:**

1. Navigate to the chosen community group.
2. Locate the active community challenge.
3. Tap the "Join Challenge" button.
4. Confirm the challenge registration.
5. Track progress within the challenge using the app's activity tracking features.
6. Verify that the progress updates are reflected within the challenge leaderboard (if applicable).

* **Postconditions:**

- * The user's participation in the challenge is accurately tracked.

* **Expected Result:** The user successfully joins the challenge, tracks progress, and sees their standing on the leaderboard.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive Testing

Test Case 4:

* **Test Case ID:** TC_CS_004

* **Test Case Title:** Verify that users can offer and receive support and encouragement.

* **Test Case Description:** This test case confirms the functionality for users to comment on posts and engage in supportive discussions within community groups.

* **Test Suite:** Community Support

* **Test Priority:** Medium

* **Preconditions:**

- * A user is logged into the app.

* A post related to fitness progress or challenge participation is available in a community group.

* **Test Data:** Sample supportive comments and messages.

* **Test Steps:**

1. Navigate to a community group and locate a relevant post.
2. Compose a supportive comment or message for the post.
3. Include encouraging words, motivational quotes, or helpful advice related to the post content.
4. Post the comment and verify its appearance below the original post.
5. Leave a like or reaction to another user's comment to demonstrate engagement.

* **Postconditions:**

* The user's comment appears under the original post in the community group feed.

* **Expected Result:** The user's supportive comment is successfully posted, and they can engage with other users' comments by leaving likes and reactions.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive Testing

Test Case 5:

* **Test Case ID:** TC_CS_005

* **Test Case Title:** Verify Community Activity Tracking.

* **Test Case Description:** This test case validates the app's ability to track and display recent community activity.

* **Test Suite:** Community Support

* **Test Priority:** Medium

* **Preconditions:**

- * A user is logged into the app.

- * There are active users within selected community group(s).

* **Test Data:** User engagement in a chosen community group (e.g., posting updates, leaving comments, joining challenges).

* **Test Steps:**

1. Access the "Community" section of the app and navigate to a chosen group.
2. Observe the displayed community feed.
3. Verify that recent activity posts (updates, comments, challenge participation) are chronologically ordered and accurately represent the latest engagements.
4. Refresh the feed to ensure that new activity updates are promptly reflected.

* **Postconditions:**

- * The "Community Activity Tracking" section reflects the recent activity within the group accurately.

* **Expected Result:** The app correctly displays recent community activity, providing users with a dynamic and up-to-date view of group engagements.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, Non-Functional Testing

* **Test Case Approach:** Positive Testing

Data - 85

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Wearable Devices

Feature description -

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

Userstory title -

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

Acceptance criteria -

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

Generated Test Cases:

Fitness Tracker App: Wearable Integration Test Cases

****Test Case ID:**** FIT-WI-001

****Test Case Title:**** Verify that the app supports integration with multiple wearable devices.

****Test Case Description:**** This test case verifies the app's ability to connect and authenticate with various popular wearable devices (e.g., Fitbit, Apple Watch, Garmin).

****Test Suite:**** Fitness Tracker App - User Authentication

****Test Priority:**** High

****Preconditions:****

- * A user account exists in the Fitness Tracker app.
- * The device has Bluetooth enabled
- * A compatible wearable device (e.g., Fitbit, Apple Watch, Garmin) is nearby and powered on.

****Test Data:**** List of supported wearable device names and credentials

****Test Steps:****

1. Open the Fitness Tracker app.
2. Navigate to the "Settings" or "Account" section.
3. Select the "Connect Wearable" option.
4. Select the desired wearable device from the list of supported devices.
5. Follow the on-screen prompts to authenticate with the wearable device.
6. Verify that the app successfully connects to the wearable device and displays its status as "Connected".

****Postconditions:****

* The wearable device is logged into the app account.

****Expected Result:**** The app should successfully connect and authenticate with multiple popular wearable devices without errors.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:**** FIT-WI-002

****Test Case Title:**** Verify that users can sync activity data from their wearables.

****Test Case Description:**** This test case verifies the app's ability to receive and synchronize activity data (steps, heart rate, calories burned) from a connected wearable device.

****Test Suite:**** Fitness Tracker App - Data Synchronization

****Test Priority:**** High

****Preconditions:****

* A user account with a connected wearable device as per FIT-WI-001

* The wearable device has recorded activity data.

****Test Data:**** Sample activity data generated by the wearable device.

****Test Steps:****

1. Open the Fitness Tracker app.

2. Ensure the wearable device is connected and active.
3. Navigate to the activity tracking screen (dashboard or history).
4. Manually trigger a data synchronization process.
5. Observe the activity data on the app's interface.

****Postconditions:****

* Activity data from the wearable device should be accurately reflected in the app.

****Expected Result:**** The app should successfully sync and display activity data from the connected wearable device in real time.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive - real-time data synchronisation

****Test Case ID:**** FIT-WI-003

****Test Case Title:**** Verify the app provides real-time data updates from wearables while exercising.

****Test Case Description:**** This test case verifies the app's ability to display real-time data updates from a connected wearable device while the user is actively engaged in an exercise.

****Test Suite:**** Fitness Tracker App - Workout Activity

****Test Priority:**** High

****Preconditions:****

- * A user account with a connected wearable device as per FIT-WI-001
- * The wearable device is tracking data while the user exercises (e.g., heart rate, steps).
- * The app is open and the workout tracking screen is active.

****Test Data:**** No test data needed

****Test Steps:****

1. Start a workout session within the Fitness Tracker app.
2. Simultaneously, engage in a physical activity that triggers data collection on the connected wearable device (e.g., walking, running, cycling).
3. Observe the real-time data updates (steps, heart rate, calories burned) displayed on the app's workout screen.
4. Compare the data displayed on the app to the data being recorded by the wearable device.

****Postconditions:****

- * The workout session is completed.

****Expected Result:**** The app should display real-time data updates from the wearable device during the workout session, accurately reflecting the user's activity level.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:** FIT-WI-004**

****Test Case Title:**** Verify that wearable data includes steps, heart rate, and calories burned.

****Test Case Description:**** This test case confirms that the app receives and interprets specific data points (steps, heart rate, calories burned) from a connected wearable device.

****Test Suite:**** Fitness Tracker App - Data Accuracy

****Test Priority:**** Medium

****Preconditions:****

- * A user account with a connected wearable device
- * The wearable device has recorded data including steps, heart rate, and calories burned.

****Test Data:****

- * Steps

****Test Steps:****

1. Open the Fitness Tracker app and ensure the wearable device is connected.
2. Review the user profile or tracked activity screen to identify the fields for steps, heart rate, and calories burned.
3. Compare the reported data under these fields to the data logged by the connected wearable device using its native app or interface.

****Postconditions:****

- * The data comparison is complete.

****Expected Result:**** The app should accurately display the step count, heart rate, and calories burned data received from the wearable device.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive - data comparison

****Test Case ID:**** FIT-WI-005

****Test Case Title:**** Verify that the app allows troubleshooting connectivity issues with wearables.

****Test Case Description:**** This test case verifies the app's functionality to guide users through troubleshooting common connectivity issues with their wearable devices.

****Test Suite:**** Fitness Tracker App - Error Handling

****Test Priority:**** Medium

****Preconditions:****

- * A user account with a connected wearable device

- * A simulated connectivity issue (e.g., Bluetooth disconnected, device out of range).

****Test Data:**** Simulated connectivity issues

****Test Steps:****

1. Initiate a simulated connectivity issue with the wearable device.
2. Observe the app's response to the lost connection.

3. Follow the on-screen prompts provided by the app for troubleshooting.
4. Verify if the app guides the user through appropriate troubleshooting steps (e.g., re-pair device, check Bluetooth settings, restart device).

****Postconditions:****

- * The Connectivity issue is addressed.

****Expected Result:**** The app should clearly indicate the connection issue and provide users with actionable troubleshooting steps to resolve it.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Usability Testing

****Test Case Approach:**** Negative - simulating connectivity issues

Data - 86

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Water Intake Tracking

Feature description -

Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

Userstory title -

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

Acceptance criteria -

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

Generated Test Cases:

Fitness Tracker App: Water Intake Tracking Test Cases

Here are 5 detailed test cases for the "Water Intake Tracking" feature of the Fitness Tracker app, following the provided template:

****Test Case ID:**** FIT-WI001

****Test Case Title:**** Verify that users can log their daily water intake.

****Test Case Description:**** This test case verifies the functionality of logging water intake manually. It will ensure that users can input the amount of water consumed and the app correctly records this information.

****Test Suite:**** Water Intake Tracking

****Test Priority:** High**

****Preconditions:****

- User is logged into the Fitness Tracker app.
- User has access to the water intake tracking feature.

****Test Data:****

- Various amounts of water (e.g., 250ml, 500ml, 1 liter)

****Test Steps:****

1. Navigate to the water intake tracking section of the app.
2. Input a specific amount of water consumed (e.g., 500ml).
3. Tap the "Log" or "Add" button to record the intake.
4. Verify that the logged water intake is displayed in the appropriate section (e.g., daily log, consumption graph).
5. Repeat steps 2-4 with different water amounts.

****Postconditions:****

- Logged water intake remains visible in the app.

****Expected Result:**** The app successfully logs each water intake entered by the user.

****Severity:** Critical**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case ID:** FIT-WI002**

****Test Case Title:** Verify that the app calculates total water consumption for the day.**

****Test Case Description:**** This test case verifies the functionality of the app to accurately calculate the total water consumed by the user throughout the day.

****Test Suite:** Water Intake Tracking**

****Test Priority:** High**

****Preconditions:****

- User is logged into the Fitness Tracker app.
- User has logged multiple water intakes for the current day.

****Test Data:****

- Multiple water intake values (total exceeding the user's goal).

****Test Steps:****

1. Log several water intake values throughout the day using various amounts.
2. Navigate to the water intake tracking section of the app.
3. Verify that the "Total Water Consumed" value accurately reflects the sum of all logged intakes.
4. Repeat steps 1-3 with different intake amounts and frequencies.

****Postconditions:****

- The calculated total water consumption remains accurate.

****Expected Result:**** The app calculates and displays the total water consumed for the day accurately.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case ID:**** FIT-WI003

****Test Case Title:**** Verify that users can set daily hydration goals.

****Test Case Description:**** This test case verifies the functionality of setting a personalized daily water intake goal.

****Test Suite:**** Water Intake Tracking

****Test Priority:**** High

****Preconditions:****

- User is logged into the Fitness Tracker app.
- User has access to the water intake tracking feature.

****Test Data:****

- Valid integer values representing water intake goals (e.g., 2000ml, 3000ml)

****Test Steps:****

1. Navigate to the water intake tracking section of the app.
2. Locate the option to set a daily hydration goal.
3. Input a specific water intake goal value (e.g., 2500ml).
4. Tap the "Save" or "Set Goal" button to confirm the goal.
5. Verify that the set goal is displayed prominently within the water intake tracking section.

****Postconditions:****

- The set hydration goal is saved and persists after closing and reopening the app.

****Expected Result:**** The app allows users to set a desired daily water intake goal and displays the goal clearly.

****Test Case ID:**** FIT-WI004

****Test Case Title:**** Verify that reminders are provided to encourage regular water intake.

****Test Case Description:**** This test case verifies the app's functionality to remind users to drink water throughout the day.

****Test Suite:**** Water Intake Tracking

****Test Priority:**** High

****Preconditions:****

- User has set a daily hydration goal in the app.
- User has enabled reminder notifications in app settings.

****Test Data:****

- N/A

****Test Steps:****

1. Set a daily hydration goal within the app.

2. Enable in-app notifications for hydration reminders.
3. Allow sufficient time for scheduled reminders (e.g., hourly, every 2 hours).
4. Observe if and when reminders are displayed on the user's device.
5. Record the frequency and timing of the reminders.

****Postconditions:****

- Reminder notifications are received according to the chosen schedule.

****Expected Result:**** The app sends timely reminders to encourage users to drink adequate water throughout the day.

****Test Case ID:**** FIT-WI005

****Test Case Title:**** Verify that users can view hydration progress and trends.

****Test Case Description:**** This test case verifies the app's ability to track hydration progress and present it to the user in a meaningful way.

****Test Suite:**** Water Intake Tracking

****Test Priority:**** Medium

****Preconditions:****

- User has logged multiple water intakes over several days.

****Test Data:****

- N/A

****Test Steps:****

1. Navigate to the water intake tracking section of the app.
2. Review the daily log of water intake and confirm all entries are accurate.
3. Observe the display of hydration progress:
 - Percentage consumed toward the daily goal

- Graphical representation of water consumption over the logged period
 - Historical data visualization (e.g., weekly, monthly trends)
4. Adjust the time period view (e.g., daily, weekly, monthly) and verify that the displayed progress reflects the selected timeframe.

****Postconditions:****

- Hydration progress metrics are calculated and updated accurately.

****Expected Result:**** The app presents clear and insightful data on the user's hydration progress and trends over time.

Data - 87

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Nutrition Tracking

Feature description -

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

Userstory title -

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

Acceptance criteria -

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

Generated Test Cases:

Here are 5 detailed test cases for the "Nutrition Tracking" feature of the Fitness Tracker App, following the provided template:

Test Case 1

****Test Case ID:**** TC_NUTRITION_001

****Test Case Title:**** Verify that users can log their daily meals and snacks.

****Test Case Description:**** This test case will verify the functionality of logging meals and snacks within the app's food diary. It will assess the ability to input food names, quantities, and meal types.

****Test Suite:**** Nutrition Tracking

****Test Priority:**** High

****Preconditions:****

- * User is logged into the app.
- * The app is connected to internet.

****Test Data:****

- * List of common foods (e.g., apple, banana, chicken breast) with varying quantities (e.g., 1, 2, 0.5)
- * Different meal types (breakfast, lunch, dinner, snack)

****Test Steps:****

1. Navigate to the "Nutrition" section of the app.
2. Select the "Log Meal" option.
3. Enter the name of a food item in the search bar or select it from the food database.
4. Input the quantity of the food item consumed.
5. Choose the meal type (e.g., Breakfast, Lunch, Dinner, Snack).
6. Save the logged meal.

****Postconditions:****

- * The logged meal appears in the food diary.

****Expected Result:**** The user successfully logs various meals and snacks with accurate information displayed in the food diary.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Regression Testing

****Test Case Approach:**** Positive

Test Case 2

****Test Case ID:**** TC_NUTRITION_002

****Test Case Title:**** Verify that the app calculates macronutrient intake (carbs, fats, proteins).

****Test Case Description:**** This test case will verify the accuracy of macronutrient calculations for logged meals. It will test the calculation of carbohydrates, fats, and proteins based on user-entered food information.

****Test Suite:**** Nutrition Tracking

****Test Priority:**** High

****Preconditions:****

* User is logged into the app.

* The app is connected to internet.

****Test Data:****

* List of foods with known macronutrient values (e.g., apple, banana, chicken breast)

* Different quantities of the foods.

****Test Steps:****

1. Log multiple meals with varying quantities of foods containing known macronutrients.
2. Review the "Nutrition Summary" section of the app.
3. Verify the displayed values for total carbohydrates, fats, and proteins match the expected values based on the logged foods and quantities.

****Postconditions:****

* None.

****Expected Result:**** The app accurately calculates and displays the total macronutrient intake for logged meals.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Regression Testing

****Test Case Approach:**** Positive

Test Case 3

****Test Case ID:**** TC_NUTRITION_003

****Test Case Title:**** Verify that users can set nutritional goals and track progress.

****Test Case Description:****

This test case will verify the functionality of setting personalized nutritional goals (e.g., daily calorie intake, protein target) and tracking progress towards those goals.

****Test Suite:**** Nutrition Tracking

****Test Priority:**** High

****Preconditions:****

* User is logged into the app.

* The app is connected to internet.

****Test Data:****

* User-defined calorie goal

* User-defined macronutrient (protein) target

****Test Steps:****

1. Navigate to the "Goals" section within the Nutrition Tracking module.
2. Set a new daily calorie goal.
3. Set a target for daily protein intake.
4. Log meals for several days.
5. Review the "Progress" section to verify the app calculates and displays progress toward the set goals.
6. Compare the logged macronutrient intake with the set target.

****Postconditions:****

****Expected Result:**** The app accurately calculates and displays progress toward defined nutritional goals, based on logged meal data.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, Regression Testing

****Test Case Approach:**** Positive

Test Case 4

****Test Case ID:**** TC_NUTRITION_004

****Test Case Title:**** Verify that the app provides nutritional information and suggestions.

****Test Case Description:**** This test case will verify the app's ability to display nutritional information for logged foods and provide helpful suggestions based on dietary habits.

****Test Suite:**** Nutrition Tracking

****Test Priority:**** Medium

****Preconditions:****

* User is logged into the app.

* The app is connected to internet.

****Test Data:****

* Several meals with foods from different nutritional categories (fruits, vegetables, protein sources).

****Test Steps:****

1. Log various meals with different food choices.
2. View the detailed information panel for a logged food item.
3. Verify the app displays information like calories, macronutrients, vitamins, and minerals.
4. Check if the app provides suggestions for healthier food swaps or balanced meals based on user's logged intake.

****Postconditions:****

****Expected Result:**** The app provides accurate nutritional information for logged foods and offers relevant suggestions for dietary improvement.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Regression Testing

****Test Case Approach:**** Positive

Test Case 5

****Test Case ID:** TC_NUTRITION_005**

****Test Case Title:** Verify that users can view their nutritional history and trends.**

****Test Case Description:** This test case will verify the functionality of accessing and visualizing past nutrition data, including trends in food choices and macronutrient intake.**

****Test Suite:** Nutrition Tracking**

****Test Priority:** Medium**

****Preconditions:****

- * User is logged into the app.

- * The app has been used to log a significant amount of historical meal data.

****Test Data:****

- * Data from several weeks of logged meals.

****Test Steps:****

1. Navigate to the "History" section within Nutrition Tracking.
2. Review the daily logs of meals and snacks over different periods (e.g., last week, last month).
3. Explore the graphical representations (charts or graphs) showing trends in caloric intake, macronutrient breakdown, and food groups consumed.

4. Utilize filtering options to focus on specific foods or meal types.

****Postconditions:****

****Expected Result:**** The app allows users to easily access, view, and analyze their past nutrition history.

*** **Severity:** Major**

*** **Priority:** Medium**

*** **Type of Testing:** Functional Testing, Regression 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:

Test Case Generation for Fitness Tracker App: Progress Reports Feature

****Test Case ID:**** FIT-PR-001

****Test Case Title:**** Verify that users receive weekly progress reports.

****Test Case Description:**** This test case verifies the functionality of the app to generate and deliver weekly progress reports to users as scheduled. This includes checking the report content, format, and delivery method.

****Test Suite:**** Progress Reports

****Test Priority:** High**

****Preconditions:****

- * User has an active account in the Fitness Tracker App.
- * User has logged at least one week's worth of activity data.
- * Notification settings are enabled for the user.

****Test Data:**** User activity data for at least one week.

****Test Steps:****

1. User logs into the Fitness Tracker App.
2. User navigates to the report section or dashboard.
3. User verifies that a progress report for the past week is available.
4. User opens the weekly progress report.
5. User checks the report for details on activity levels, workout performance, and goal attainment.
6. User confirms receipt of a notification related to the weekly progress report.

****Postconditions:****

- * Weekly progress report is successfully displayed and viewed by the user.

****Expected Result:****

A detailed weekly progress report is generated and delivered to the user through the app and a notification. The report includes relevant data on activity levels, workouts, and goal attainment.

****Severity:****

Critical

****Priority:****

High

****Type of Testing:****

Functional Testing, Acceptance Testing

****Test Case Approach:****

Positive

****Test Case ID:**** FIT-PR-002

****Test Case Title:**** Verify that progress reports include activity levels, workout performance, and goal attainment.

****Test Case Description:**** This test case verifies that the app includes all specified data points within the progress reports, ensuring comprehensive user feedback.

****Test Suite:**** Progress Reports

****Test Priority:**** High

****Preconditions:****

* User has an active account in the Fitness Tracker App.

* User has logged at least one week's worth of activity data, including workouts and specific goals.

****Test Data:**** User activity data for at least one week with detailed workout information and goals.

****Test Steps:****

1. User logs into the Fitness Tracker App.
2. User navigates to the report section or dashboard.
3. User opens the most recent weekly progress report.
4. User verifies the presence of data on activity levels (steps, calories burned, distance travelled).

5. User checks for details on workout performance (duration, types of workouts, intensity).

6. User confirms that the report reflects their progress towards set goals.

****Postconditions:****

* All expected data points are clearly displayed and accurately reflect user activity.

****Expected Result:****

The progress reports display comprehensive data on activity levels, workout performance, and goal attainment, providing a detailed overview of the user's progress.

****Severity:****

Critical

****Priority:****

High

****Type of Testing:****

Functional Testing, Acceptance Testing

****Test Case Approach:****

Positive

Data - 89

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

In-App Challenges

Feature description -

The app offers in-app challenges and competitions, encouraging users to set personal records, compete with friends, and earn rewards for achieving fitness milestones. In-app challenges add an element of fun, motivation, and friendly competition to the fitness tracking experience.

Userstory title -

As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals.

Acceptance criteria -

['Users can join various in-app challenges.', 'Challenges have clear objectives and rules.', 'Users can track their progress within the challenge.', 'Rewards are provided for achieving challenge milestones.', 'Users can compete with friends and view leaderboards.']

Generated Test Cases:

Here are 5 test cases for the "In-App Challenges" feature of the Fitness Tracker App, following your template:

****Test Case 1****

* **Test Case ID:** TC_FIT_CHALL_001

* **Test Case Title:** Verify that users can join various in-app challenges

* **Test Case Description:** This test case verifies that users can access and join different types of pre-defined challenges within the app.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

- * The user is logged in to the app.

- * There are at least three different in-app challenges available.

* **Test Data:** N/A

* **Test Steps:**

1. Navigate to the "Challenges" section of the app.
2. Observe and verify that three or more distinct challenge cards are displayed.
3. Select one of the challenge cards.
4. Verify that the challenge details, rules, duration, and rewards are clearly presented.
5. Tap the "Join Challenge" button.
6. Confirm that the user is successfully enrolled in the chosen challenge.

* **Postconditions:**

- * The user is now part of the joined challenge.

- * The challenge should be reflected in the user's profile or activity feed.

* **Expected Result:** The user can successfully join the chosen challenge.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

****Test Case 2****

* **Test Case ID:** TC_FIT_CHALL_002

* **Test Case Title:** Verify that challenges have clear objectives and rules

* **Test Case Description:** This test case ensures that the rules and objectives of an in-app challenge are clearly defined and understandable to the user.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

- * The user is logged in to the app.

- * A specific challenge with defined rules is available.

* **Test Data:** N/A

* **Test Steps:**

1. Navigate to the "Challenges" section of the app.
2. Select a challenge card with a description.
3. Read the challenge's objectives, rules, and any specific requirements.
4. Verify that the language is clear, concise, and easy to comprehend.
5. Ensure that the rules are comprehensive and cover all aspects of the challenge.

* **Postconditions:** N/A

* **Expected Result:** The challenge's objectives and rules are clearly stated and easy to understand.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

****Test Case 3****

* **Test Case ID:** TC_FIT_CHALL_003

* **Test Case Title:** Verify that users can track their progress within the challenge

* **Test Case Description:** This test case verifies that the app accurately tracks the user's progress towards the challenge goals and displays it effectively.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

- * The user has joined a challenge.

- * The user has performed some actions related to the challenge.

* **Test Data:** The user's initial data for the challenge (e.g., steps, calories burned).

* **Test Steps:**

1. Navigate to the "Challenges" section.

2. Select the joined challenge.

3. Observe the user's current progress towards the challenge goals (e.g., steps completed, calories burned, distance covered).

4. Perform additional actions related to the challenge (e.g., walk for 15 minutes, run on the treadmill).

5. Refresh or navigate back to the challenge screen.

6. Verify that the progress tracker reflects the newly completed actions and the updated progress status.

* **Postconditions:** N/A

* **Expected Result:** The challenge progress tracker accurately reflects the user's activities and progress towards the challenge goals.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

****Test Case 4****

* **Test Case ID:** TC_FIT_CHALL_004

* **Test Case Title:** Verify that rewards are provided for achieving challenge milestones

* **Test Case Description:** This test case ensures that the app offers rewards as promised when users reach specific milestones during a challenge.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

- * The user has joined a challenge with defined reward tiers.

- * The user has an account with sufficient in-app currency or a points system.

* **Test Data:** N/A

* **Test Steps:**

1. Participate in the chosen challenge and aim to achieve a specific milestone.

2. Once the milestone is reached, observe if the app awards the promised reward (e.g., in-app currency, badges, virtual trophies).

3. Verify if the reward is correctly reflected in the user's profile or inventory.

* **Postconditions:** N/A

* **Expected Result:** The user receives the designated reward upon achieving the defined challenge milestone.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC_FIT_CHALL_005

* **Test Case Title:** Verify that users can compete with friends and view leaderboards

* **Test Case Description:** This test case ensures that the app facilitates friendly competition by allowing users to view leaderboards and compare their progress with friends who participate in the same challenge.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

- * The user is logged in.

- * The user has connected to at least one friend on the platform.

* **Test Data:** N/A

* **Test Steps:**

1. Participate in a challenge with leaderboard functionality.

2. Observe the leaderboard for that specific challenge.

3. Verify that your friend's username and their current rank are displayed on the leaderboard.

4. If your friend is ahead of you, verify that their progress is higher than yours.

* **Postconditions:** N/A

* **Expected Result:** The user can see the leaderboard displaying their friends' progress and their own ranking within the challenge.

* **Severity:** High

* **Priority:** High

* **Type of Testing:** Functional Testing, Acceptance Testing

* **Test Case Approach:** Positive

Data - 90

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Customizable Dashboards

Feature description -

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

Userstory title -

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Acceptance criteria -

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

Generated Test Cases:

Fitness Tracker App - Customizable Dashboards Test Cases

Test Case 1

* **Test Case ID:** TC_CD_001

* **Test Case Title:** Verify that users can select metrics to display on their dashboard

* **Test Case Description:** This test case verifies that users can choose from a list of available fitness metrics and add them to their personalized dashboard display.

* **Test Suite:** Dashboard Customization

* **Test Priority:** High

* **Preconditions:**

- * User is logged into the fitness tracker app.

- * User has at least one fitness metric recorded in the app (e.g., steps, calories burned).

* **Test Data:** List of available metrics (steps, calories burned, distance, heart rate, etc.)

* **Test Steps:**

1. Navigate to the dashboard customization settings.
2. Select at least three different metrics from the list.
3. Add the selected metrics to the dashboard display.
4. Verify that the selected metrics are displayed on the dashboard in real-time.
5. Remove one of the added metrics and confirm its absence from the dashboard display.

* **Postconditions:**

- * The dashboard displays the chosen metrics.

* **Expected Result:** The selected metrics are successfully displayed on the dashboard, updating in real-time.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TC_CD_002

* **Test Case Title:** Verify that users can customize the dashboard layout

* **Test Case Description:** This test case ensures that users can rearrange the position of displayed metrics on the dashboard and adjust the size of the individual widgets.

* **Test Suite:** Dashboard Customization

* **Test Priority:** High

* **Preconditions:**

- * User is logged into the fitness tracker app

- * User has at least three metrics displayed on the dashboard

* **Test Data:** Available dashboard layout options (columns, rows, grids)

* **Test Steps:**

1. Access the dashboard customization settings.
2. Drag and drop a metric widget to a different position on the dashboard.
3. Verify that the metric widget is moved to the new position.
4. Adjust the size of a metric widget using the available sizing options.
5. Confirm that the widget size is updated accordingly.

* **Postconditions:**

- * Dashboard layout and widget sizes are modified as per user's actions.

* **Expected Result:** Users can seamlessly customize the dashboard layout and widget sizes without any errors or unexpected behavior.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TC_CD_003

* **Test Case Title:** Verify that users can change the dashboard theme

* **Test Case Description:** This test case ensures that users can apply different themes to their dashboard for a personalized visual experience.

* **Test Suite:** Dashboard Customization

* **Test Priority:** Medium

* **Preconditions:**

- * User is logged into the fitness tracker app.

* **Test Data:** List of available dashboard themes (e.g., light, dark, sports, minimalist)

* **Test Steps:**

1. Navigate to the dashboard customization settings.
2. Choose a theme from the available options.
3. Observe the changes in the dashboard appearance (background color, text color, widget style)

* **Postconditions:**

- * Dashboard displays the chosen theme.

* **Expected Result:** The dashboard theme is successfully changed according to user selection.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4

* **Test Case ID:** TC_CD_004

* **Test Case Title:** Verify that users can save and update their dashboard settings

* **Test Case Description:** This test case validates that users can save their changes to the dashboard

customization and reload the updated dashboard layout.

* **Test Suite:** Dashboard Customization

* **Test Priority:** Medium

* **Preconditions:**

- * User is logged into the fitness tracker app.

- * User has made changes to the dashboard layout, metrics, and/or theme.

* **Test Data:** Saved dashboard settings (e.g., layout, metrics, theme)

* **Test Steps:**

1. Apply desired changes to the dashboard (e.g., add/remove metrics, adjust layout, choose a theme).

2. Save the dashboard settings.

3. Log out of the app and log back in.

4. Verify that the saved dashboard settings are loaded and displayed correctly.

* **Postconditions:** Dashboards with customized settings are loaded correctly after logging in.

* **Expected Result:** The updated dashboard settings are successfully saved and are restored when the user logs back in.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5

* **Test Case ID:** TC_CD_005

* **Test Case Title:** Verify that users can reset the dashboard to default settings

* **Test Case Description:** This test case ensures that users can revert their dashboard back to the original factory settings, losing any applied customizations.

* **Test Suite:** Dashboard Customization

* **Test Priority:** Medium

* **Preconditions:**

- * User is logged into the fitness tracker app.

- * User has made adjustments to their dashboard settings.

* **Test Data:** Default dashboard settings (layout, metrics, theme)

* **Test Steps:**

1. Navigate to the dashboard customization settings.
2. Locate and select the "Reset to Default" option.
3. Confirm the reset action.
4. Verify that the dashboard returns to its original, pre-customized state.

* **Postconditions:** Dashboard settings have been reset to the default configuration.

* **Expected Result:** The dashboard successfully resets to its default configuration.

* **Severity:** Minor

* **Priority:** Medium

* **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:

Fitness Tracker App: Social Sharing Test Cases

User Story Title: As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

Test Suite: Social Sharing Feature

Test Case 1

Test Case ID: TC_001

****Test Case Title:**** Verify that users can share workout summaries on social media.

****Test Case Description:**** This test case verifies the functionality of sharing a completed workout summary to the user's selected social media platform.

****Test Priority:**** High

****Preconditions:****

- * User has successfully logged into the app.
- * User has completed a workout and saved it.
- * User has connected their preferred social media account to the app.

****Test Data:**** Sample workout data with details like duration, type, calories burned.

****Test Steps:****

1. Navigate to the user's workout history.
2. Select the completed workout to be shared.
3. Choose the desired social media platform from the sharing options.
4. Review and modify the pre-populated workout summary content as desired.
5. Initiate the sharing process.
6. Verify that the workout summary is successfully posted on the selected social media platform.

****Postconditions:****

- * Workout summary is visible on the selected social media platform.

****Expected Result:**** The workout summary is shared on the chosen social media platform with accurate details and the ability to customize the content.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Acceptance Testing

****Test Case Approach:**** Positive

Test Case 2

****Test Case ID:**** TC_002

****Test Case Title:**** Verify that the app supports sharing on multiple social media platforms.

****Test Case Description:**** This test case confirms that users can share their accomplishments on various popular social media platforms supported by the app.

****Test Priority:**** Medium

****Preconditions:****

- * User has successfully logged into the app.
- * User has completed a workout and saved it.
- * User has connected a social media account from each supported platform.

****Test Data:**** Sample workout data

****Test Steps:****

1. Navigate to the user's workout history.
2. Select the completed workout to be shared.
3. Review the list of supported social media platforms for sharing.
4. Choose each supported platform one by one and initiate the sharing process.
5. Verify that the workout summary is successfully posted on each selected platform.

****Postconditions:****

* Workout summary is visible on each selected social media platform.

****Expected Result:**** The app should effectively share the workout summary to all configured social media platforms without encountering any errors.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Acceptance Testing

****Test Case Approach:**** Positive

Test Case 3

****Test Case ID:**** TC_003

****Test Case Title:**** Verify that users can customize the content and format of shared posts

****Test Case Description:**** This test case ensures that users have control over the content and appearance of their shared workout achievements.

****Test Priority:**** High

****Preconditions:****

* User has successfully logged into the app.

* User has completed a workout and saved it.

* User has connected a social media account.

****Test Data:****

* Varying text lengths for workout summaries.

- * Various emoji options to be incorporated.
- * Different image selections illustrating the workout.

****Test Steps:****

1. Navigate to the user's workout history.
2. Select the completed workout to be shared.
3. Review and modify the pre-populated workout summary content, experimenting with different lengths and emoji usage.
4. Upload different images, if applicable
5. Initiate the sharing process.
6. Verify that the customized content and chosen images are accurately reflected in the shared post.

****Postconditions:****

- * User has successfully customized the shared post content and formatting.

****Expected Result:**** Users should be able to adjust the text, add emojis, and select images to personalize their shared workout posts.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional Testing, Acceptance Testing

****Test Case Approach:**** Positive

Test Case 4

****Test Case ID:**** TC_004

****Test Case Title:**** Verify that sharing features are integrated seamlessly with social media.

****Test Case Description:**** This test case assesses the smoothness and user-friendliness of the integration between the app and the chosen social media platforms.

****Test Priority:**** High

****Preconditions:****

- * User has successfully logged into the app.
- * User has customized their workout summary content and pre-selected images.
- * User has connected their social media account and granted necessary permissions.

****Test Data:**** Example workout summary with customized content and images.

****Test Steps:****

1. Navigate to the user's workout history.
2. Select the completed workout to be shared.
3. Review the sharing prompt and permissions requested by the app.
4. Initiate the sharing process.
5. Verify that the user is seamlessly redirected to the connected social media platform's posting interface.
6. Confirm the successful sharing of the workout summary with the customized content and images.

****Postconditions:****

- * Workout summary is published on the social media platform without issues.

****Expected Result:**** The sharing transition should be smooth, requiring minimal user interaction, and the content should be correctly uploaded.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Integration Testing, Acceptance Testing

****Test Case Approach:** Positive**

Test Case 5

****Test Case ID:** TC_005**

****Test Case Title:** Verify that users can track the engagement of their shared posts.**

****Test Case Description:** This test case confirms the functionality of tracking likes, comments, and shares on shared posts.**

****Test Priority:** Medium**

****Preconditions:****

- * User has successfully logged into the app and connected their social media account.
- * User has shared a workout summary on a supported platform.

****Test Data:****

- * Expected number of likes, comments, and shares for the shared post.

****Test Steps:****

1. Navigate to the social media platform where the workout summary was shared.
2. Locate the shared post.
3. Observe the number of likes, comments, and shares on the post.
4. Return to the Fitness Tracker app.
5. Open the shared workout's details and verify that the app displays the accurate engagement metrics (like, comments, shares) from the social media platform.

****Postconditions:****

- * Engagement data from social media is reflected in the app.
- * User can assess the performance of their shared post.

****Expected Result:**** The app accurately displays the engagement metrics (likes, comments, shares) for the shared post from the connected social media platform.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Acceptance 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:

Fitness Tracker App - Rest and Recovery Tracking Test Cases

****Test Case ID:** FTT_RC_01**

****Test Case Title:** Verify that users can log rest days**

****Test Case Description:**** This test case verifies that users can successfully log rest days within the app, ensuring proper functionality for tracking recovery.

****Test Suite:** Rest and Recovery**

****Test Priority:** High**

****Preconditions:****

- * User has a registered account in the Fitness Tracker app.
- * User has logged at least one workout in the past week.

****Test Data:**** Date range for rest days, optional notes for rest day

****Test Steps:****

1. Log in to the Fitness Tracker app.
2. Navigate to the "Recovery" section within the app.
3. Select the "Log Rest Day" option.
4. Choose a specific date for the rest day from the calendar.
5. (Optional) Add notes regarding the rest day, such as "Active Recovery: Light Yoga".
6. Save the logged rest day.

****Postconditions:****

- * A new rest day entry should be visible in the "Recovery" section for the chosen date.
- * The date should be marked as a rest day on the user's daily activity log.

****Expected Result:**** The rest day is successfully logged with the provided date and optional notes.

****Severity:**** Moderate

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:**** FTT_RC_02

****Test Case Title:**** Verify that the app tracks resting heart rate

****Test Case Description:**** This test case verifies that the app accurately tracks the user's resting heart rate when manually entered or automatically captured from a connected device.

****Test Suite:**** Rest and Recovery

****Test Priority:**** High

****Preconditions:****

- * User has a registered account in the Fitness Tracker app.
- * User has connected a compatible wearable device or fitness equipment.

****Test Data:**** Accurate resting heart rate reading; optional target heart rate range

****Test Steps:****

1. Ensure device is connected and tracking heart rate.
2. Open the "Recovery" section of the app.
3. Manually enter a resting heart rate reading that falls within a reasonable range.
4. Check recorded resting heart rate

****Postconditions:****

- * The entered resting heart rate should be reflected in the app's "Recovery" section.

****Expected Result:**** The app successfully tracks the user's resting heart rate with accuracy. If a wearable device is connected, compare app data with device readings.

****Severity:**** High

****Priority:**** High

****Type of Testing:**** Functional, Integration Testing

****Test Case Approach:**** Positive

****Test Case ID:**** FTT_RC_03

****Test Case Title:**** Verify that users receive recovery recommendations

****Test Case Description:**** This test case verifies that the app provides personalized recovery recommendations based on user activity and data.

****Test Suite:** Rest and Recovery**

****Test Priority:** Medium**

****Preconditions:****

- * User has logged at least 3 workout sessions in the past week.
- * User has also logged at least one rest day within the same period.
- * User's profile includes basic information like age and fitness level.

****Test Data:** Simulated workout data (intensity, duration, type)**

****Test Steps:****

1. Log into Fitness Tracker app.
2. Navigate to the "Recovery" section.
3. View the "Recommendations" tab.

****Postconditions:****

- * The app displays personalized recovery recommendations.

****Expected Result:** The app provides relevant and actionable recovery recommendations based on the user's data.**

****Severity:** Medium**

****Priority:** Medium**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** FTT_RC_04**

****Test Case Title:**** Verify that the app alerts for insufficient recovery time

****Test Case Description:**** This test case verifies the app's functionality in alerting the user when they haven't allowed sufficient recovery time before engaging in strenuous activity.

****Test Suite:**** Rest and Recovery

****Test Priority:**** High

****Preconditions:****

- * User has logged a high-intensity workout recently.
- * User has not logged any rest days within the past 48 hours.
- * User settings allow for recovery time alerts.

****Test Data:**** Simulated workout data of high intensity.

****Test Steps:****

1. Log into Fitness Tracker app.
2. Log a high-intensity workout session.
3. Attempt to log another high-intensity workout within the next 24 hours.

****Postconditions:****

- * The app displays an alert message.

****Expected Result:**** The app alerts the user about insufficient recovery time and discourages them from logging another high-intensity workout immediately.

****Severity:**** Moderate

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive, Negative

****Test Case ID:** FTT_RC_05**

****Test Case Title:** Verify that users can view trends and history of their recovery data**

****Test Case Description:** This test case verifies the app's ability to display historical and trend data on rest days logged, resting heart rate, and other relevant recovery metrics.**

****Test Suite:** Rest and Recovery**

****Test Priority:** Medium**

****Preconditions:****

* User has logged at least 7 rest days over a period of 2 weeks.

* User has logged at least 5 heart rate readings within those 2 weeks.

****Test Data:** User's logged recovery data**

****Test Steps:****

1. Log into the Fitness Tracker app.
2. Navigate to the "Recovery" section.
3. Access the "History" or "Trends" tab.

****Postconditions:****

- * The user can view a visual representation of their recovery data trends (e.g., line graph for resting heart rate, bar graph for rest day frequency).
- * Data filter options (by date range, activity type etc.) are available

****Expected Result:** The app displays clear and informative visualizations of the user's recovery data trends and history.**

****Severity:**** Moderate

****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:

Fitness Tracker App: Audio Coaching Test Cases

Here are 5 detailed test cases for the "Audio Coaching" feature of the Fitness Tracker App:

Test Case 1

* **Test Case ID:** TC-ACT-001

* **Test Case Title:** Verify that users can access a variety of audio-guided workouts.

* **Test Case Description:** This test case verifies the availability and diversity of audio coaching options for different workout types within the app.

* **Test Suite:** Audio Coaching

* **Test Priority:** High

* **Preconditions:**

- * The user is logged into the app.

- * The app has an active internet connection.

* **Test Data:** Exercise types (e.g., running, strength training, yoga, HIIT)

* **Test Steps:**

1. Navigate to the "Workouts" section of the app.
2. Select the "Audio Coaching" filter.
3. Verify that a list of audio workouts is displayed for each selected exercise type.
4. Check if the workouts offer varying durations and intensity levels.

* **Postconditions:**

* Close the "Workouts" section.

* **Expected Result:**

A diverse range of audio-guided workouts for different exercise types, durations, and intensities should be accessible.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2

* **Test Case ID:** TC-ACT-002

* **Test Case Title:** Verify that audio coaching includes motivational messages and coaching cues.

* **Test Case Description:** This test case ensures the audio coaching integrates motivational messages and real-time cues to guide the user during the workout.

* **Test Suite:** Audio Coaching

* **Test Priority:** High

* **Preconditions:**

* The user is logged into the app.

* An audio workout is selected from the app.

* **Test Data:** Actively use an audio guided workout.

* **Test Steps:**

1. Start an audio workout session.

2. Observe and listen attentively to the audio guide for the duration of the workout.

3. Make note of the type and frequency of motivational messages and coaching cues provided.

* **Postconditions:**

* Stop the audio workout session.

* **Expected Result:**

The audio guide must consistently provide motivational messages and relevant coaching cues throughout the workout, aligning with the exercise instructions.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3

* **Test Case ID:** TC-ACT-003

* **Test Case Title:** Verify that users can choose coaching based on workout type and intensity.

* **Test Case Description:** This test case confirms the user's ability to filter and select audio coaching based on their desired workout type and intensity level.

* **Test Suite:** Audio Coaching

* **Test Priority:** High

* **Preconditions:**

- * The user is logged into the app.

- * The app has an active internet connection.

* **Test Data:** Exercise types (e.g., running, strength training, yoga), Intensity levels (Beginner, Intermediate, Advanced)

* **Test Steps:**

1. Navigate to the "Workouts" section of the app.
2. Select the "Audio Coaching" filter.
3. Choose a specific exercise type (e.g., running).
4. Further filter the results by intensity level (e.g., Beginner).
5. Verify that only audio workouts matching the chosen type and intensity are displayed.

* **Postconditions:**

* Close the "Workouts" section.

* **Expected Result:**

The app should accurately display audio coaching options that align with the selected workout type and intensity level.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

****Test Case 4****

* **Test Case ID:** TC-ACT-004

* **Test Case Title:** Verify seamless integration of audio coaching with other app features.

* **Test Case Description:** This test case evaluates the functionality and fluidity of audio coaching when used alongside other app features, ensuring a smooth user experience.

* **Test Suite:** Audio Coaching

* **Test Priority:** Medium

* **Preconditions:**

* The user is logged into the app.

* An audio workout is selected and playing.

* **Test Data:** Other app features, e.g., heart rate tracking, progress tracking

* **Test Steps:**

1. Start an audio workout session.
2. Simultaneously, monitor the user's heart rate tracking data displayed on the app.
3. Pause and resume the audio workout while observing the app's workflow and data updates.
4. Track the user's progress within the workout and verify its consistent display in the app.

* **Postconditions:**

- * Stop the audio workout session and close the app completely.

* **Expected Result:**

Audio coaching should seamlessly integrate with heart rate tracking, progress tracking, and pausing/resuming features without causing any interruptions or discrepancies in data display.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing, Integration Testing

* **Test Case Approach:** Positive, Negative

****Test Case 5****

* **Test Case ID:** TC-ACT-005

* **Test Case Title:** Verify the user's ability to control audio settings and preferences.

* **Test Case Description:** This test case ensures users can personalize their audio coaching experience by adjusting volume, playback speed, and other relevant settings.

* **Test Suite:** Audio Coaching

* **Test Priority:** Medium

* **Preconditions:**

- * The user is logged into the app.

- * An audio workout is selected and playing.

* **Test Data:** Volume settings, playback speed options (e.g., faster, slower)

* **Test Steps:**

1. Access the app's audio settings while the audio workout is running.
2. Adjust the volume level up and down, verifying the corresponding changes in audio output.
3. Change the playback speed, listening to the audio at faster and slower rates.
4. Verify that other settings, such as language selection or cue repetition, function as intended.

* **Postconditions:**

* Return the audio settings to their default values.

* Stop the audio workout session.

* **Expected Result:**

The user should be able to effortlessly control audio settings and preferences, allowing for a personalized and enjoyable audio coaching experience.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing, Usability 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:

Fitness Tracker App - Test Cases for Health App Integration

****Test Case 1:****

****Test Case ID:**** HTC-001

****Test Case Title:**** Verify that the app integrates with popular health and wellness apps.

****Test Case Description:**** This test case verifies that the Fitness Tracker App successfully connects and synchronizes data with a selection of popular health and wellness apps.

****Test Suite:**** Health App Integration

****Test Priority:**** High

****Preconditions:****

- * A user account is created in the Fitness Tracker App.
- * The app has access to the device's necessary permissions for third-party app integration.
- * Internet connectivity is available.

* A test account exists for a popular health app (e.g., MyFitnessPal, Headspace).

****Test Data:**** Credentials for a test health app account.

****Test Steps:****

1. Navigate to the 'Connect Apps' section within the Fitness Tracker App.
2. Select the desired test health app from the list of available integrations.
3. Enter the required credentials for the test health app account.
4. Grant the necessary permissions requested by the app.
5. Verify that the app successfully connects to the chosen test health app.

****Postconditions:****

- * The Fitness Tracker App confirms a successful connection with the test health app.
- * Data from the connected test health app is displayed within the Fitness Tracker App.

****Expected Result:**** The app successfully connects to the chosen popular health app, and data from the app is appropriately displayed in the Fitness Tracker App.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Integration Testing, Functional Testing

****Test Case Approach:**** Positive

****Test Case 2:****

****Test Case ID:**** HTC-002

****Test Case Title:**** Verify that users can sync data from connected health apps.

****Test Case Description:**** This test case evaluates the functionality of data synchronization between the Fitness Tracker App and a connected health app.

****Test Suite:**** Health App Integration

****Test Priority:**** High

****Preconditions:****

- * A user account is created in the Fitness Tracker App.

- * The app has access to the device's necessary permissions for third-party app integration.

- * Internet connectivity is available.

- * A test account exists for a popular health app (e.g., MyFitnessPal, Headspace), and is successfully connected to the Fitness Tracker app (following Test Case HTC-001).

****Test Data:**** Sample workout data logged in the connected test health app.

****Test Steps:****

1. Execute a workout or log relevant data within the connected test health app.
2. Verify that the synced data from the test health app is reflected in the Fitness Tracker App within a predetermined time frame (e.g., 15 minutes).

****Postconditions:****

- * The Fitness Tracker App accurately displays data from the synced workout or logged information.

****Expected Result:**** The Fitness Tracker App successfully syncs data from the connected test health app, reflecting the latest information accurately.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Integration Testing, Functional Testing

****Test Case Approach:** Positive**

****Test Case 3:****

****Test Case ID:** HTC-003**

****Test Case Title:** Verify that the app provides a consolidated view of health metrics.**

****Test Case Description:** This test case explores the app's ability to merge data from different integrated health apps to present a comprehensive overview of the user's health metrics.**

****Test Suite:** Health App Integration**

****Test Priority:** High**

****Preconditions:****

- * A user account is created in the Fitness Tracker App.
- * At least two different health apps are successfully integrated with the Fitness Tracker App (e.g., a calorie counter, a meditation app).

****Test Data:** Sample data from both integrated health apps (e.g., calorie intake, meditation sessions duration).**

****Test Steps:****

1. Log into the Fitness Tracker App and access the 'Dashboard' or 'Overview' section.
2. Verify that the combined health metrics from the connected apps are clearly displayed on the dashboard (e.g., total calories burned from both exercise and activities tracked by the fitness tracker, combined with calories consumed from the calorie counter, along with meditation session duration).

****Postconditions:****

- * The consolidated view of health metrics is visually appealing and easy to understand.

****Expected Result:**** The Fitness Tracker App presents a consolidated and organized view of health metrics from all connected health apps.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Integration Testing, Functional Testing

****Test Case Approach:**** Positive

****Continue creating tests in a similar format, covering the remaining acceptance criteria: "Users receive insights based on combined data from multiple apps" and "Integration settings are easy to manage and configure".****

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:

Test Cases: Body Composition Tracking Feature

****Test Case ID:**** TC_BC_01

****Test Case Title:**** Verify that users can log their body composition metrics.

****Test Case Description:**** This test case verifies the user's ability to input weight, body fat percentage, and muscle mass into the app.

****Test Suite:**** User Story - Body Composition Tracking

****Test Priority:**** High

****Preconditions:****

- * User is logged into the app.
- * Body composition tracking feature is enabled.
- * No existing body composition data is present.

****Test Data:****

- * Valid weight values (e.g., 70, 85, 150 kg)
- * Valid body fat percentage values (e.g., 15, 25, 35%)
- * Valid muscle mass values (e.g., 40, 50, 60 kg)

****Test Steps:****

1. Navigate to the body composition tracking section of the app.
2. Enter a valid weight value.
3. Enter a valid body fat percentage value.
4. Enter a valid muscle mass value.
5. Click on "Save" or "Add" to log the data.

****Postconditions:****

- * The logged data is successfully saved and displayed in the app.
- * The user receives a confirmation message indicating successful data saving.

****Expected Result:**** All entered body composition data is accurately saved and displayed in the app.

****Severity:**** Critical

****Priority:** High**

****Type of Testing:** Functional Testing, User Acceptance Testing**

****Test Case Approach:** Positive**

****Test Case ID:** TC_BC_02**

****Test Case Title:** Verify that the app calculates BMI based on logged data.**

****Test Case Description:** This test case verifies if the app correctly calculates the user's BMI based on the entered weight and height values.**

****Test Suite:** User Story - Body Composition Tracking**

****Test Priority:** High**

****Preconditions:****

* User is logged into the app.

****Test Data:****

* Valid weight values and height values (metric or imperial units).

****Test Steps:****

1. Navigate to the body composition tracking section of the app.
2. Enter a valid weight value.
3. Enter a valid height value.
4. Verify that the app automatically calculates the BMI based on the entered data and displays it accurately.

****Postconditions:****

- * The calculated BMI value is displayed in the app.

****Expected Result:**** The calculated BMI value is accurate and consistent with the entered weight and height data.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing

****Test Case ID:**** TC_BC_03

****Test Case Title:**** Verify that users can view trends and history of body composition changes.

****Test Case Description:**** This test case verifies the app's ability to display historical body composition data and highlight trends over time.

****Test Suite:**** User Story - Body Composition Tracking

****Test Priority:**** High

****Preconditions:****

- * User is logged into the app.

- * At least 2 body composition data points have been logged with varying values for each metric (weight, body fat percentage, muscle mass).

****Test Data:****

- * Existing body composition data logged at different time points.

****Test Steps:****

1. Navigate to the body composition tracking section of the app.
2. Review the graph or chart displaying body composition history.
3. Observe the trends in weight, body fat percentage, and muscle mass over time.
4. Verify that the display format allows for easy understanding of the trends.

****Postconditions:****

* Historical data is displayed in a clear and easy-to-understand format.

****Expected Result:**** The app displays accurate trends in body composition changes over time through a visually appealing graph or chart.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

****Test Case ID:**** TC_BC_04

****Test Case Title:**** Verify that the app provides insights and recommendations based on body composition data.

****Test Case Description:**** This test case verifies the app's ability to generate personalized insights and recommendations based on a user's body composition data.

****Test Suite:**** User Story - Body Composition Tracking

****Test Priority:**** High

****Preconditions:****

* User is logged into the app.

* Sufficient body composition data has been logged (at least 3-4 data points).

****Test Data:****

* Individualized body composition data (weight, body fat percentage, muscle mass, BMI).

* Comparison data from a healthy range (e.g., average BMI for age and gender).

****Test Steps:****

1. Navigate to the body composition tracking section of the app.

2. Analyze the insights and recommendations provided by the app based on the logged data.

3. Verify that the recommendations are specific, actionable, and relevant to the user's body composition data.

****Postconditions:****

* Insights and recommendations are displayed in the app.

****Expected Result:**** The app provides personalized insights and actionable recommendations based on the user's body composition data, promoting healthier fitness and lifestyle choices.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, User Acceptance Testing

****Test Case ID:**** TC_BC_05

****Test Case Title:**** Verify that users can set goals for body composition metrics.

****Test Case Description:**** This test case verifies the functionality of setting and tracking goals for weight, body fat percentage, muscle mass, and BMI.

****Test Suite:**** User Story - Body Composition Tracking

****Test Priority:**** High

****Preconditions:****

* User is logged into the app.

****Test Data:****

****Test Steps:****

1. Navigate to the body composition tracking section of the app.
2. Navigate to the "Goals" or "Settings" section.
3. Select a body composition metric (weight, body fat percentage, muscle mass, or BMI).
4. Set a specific and achievable goal for the selected metric (e.g., lose 5 kg of weight, reduce body fat percentage by 2%).
5. Set a target date or timeframe for achieving the goal.
6. Save the goal.

****Postconditions:****

* The goal is stored and displayed in the app.

****Expected Result:**** Users can successfully set personalized goals for their desired body composition metrics, along with a target date for achievement. The app should facilitate progress tracking and visualization

towards these goals.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional Testing, User Acceptance Testing

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:

Test Cases for Workout Analysis Feature

Project Name: Fitness Tracker App

Test Suite: Workout Analysis Feature

User Story: As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

Test Case ID	Test Case Title	Test Case Description	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
---	---	---	---	---	---	---	---	---	---	---	---	---

TC_WA_001	Verify that users can view detailed performance metrics for each workout.	This test case will verify that the app displays detailed performance metrics like distance, duration, heart rate, and calories burned for each individual workout.	High	User has completed at least one workout.	Workout data with varying distance, duration, heart rate, and calorie burn values.	1. Navigate to the Workouts tab. 2. Select a completed workout. 3. Verify that the detailed performance metrics are displayed accurately.	No specific
-----------	---	---	------	--	--	---	-------------

postconditions. | Detailed workout metrics (distance, duration, heart rate, calories burned) are accurately displayed for the selected workout. | Critical | High | Functional Testing | Positive |

| TC_WA_002 | Verify that the app provides trends and insights into workout intensity and duration. | This test case aims to ensure the app presents trends and insights regarding workout intensity and duration over time. | High | User has completed multiple workouts with varying intensities and durations. | Workout data with varying intensities and durations across multiple days. | 1. Navigate to the Workout Analysis tab. 2. Select 'Intensity Trend' or 'Duration Trend'. 3. Observe the trend visualization and data points. | No specific postconditions. | The app accurately visualizes trends in workout intensity and duration, providing insightful data points. | Major | High | Functional Testing, Regression Testing | Positive |

| TC_WA_003 | Verify that users can track calorie burn and other health metrics. | This test case verifies the functionality of tracking calorie burn and other relevant health metrics associated with workouts. | High | User has completed a workout with calorie burn data. | Workout data with recorded calorie burn. | 1. Navigate to the Workouts tab. 2. Select a completed workout. 3. Verify that the calorie burn and other tracked health metrics are displayed correctly. | No specific postconditions. | Calorie burn values alongside other tracked health metrics (e.g., steps, distance) are accurately displayed for the selected workout. | Critical | High | Functional Testing | Positive |

| TC_WA_004 | Verify that the app offers recommendations based on workout analysis. | This test case evaluates if the app provides personalized recommendations based on the collected workout data and analysis. | Medium | User has completed multiple workouts with exercise type and intensity data. | Workout data with various exercise types and intensities over several days. | 1. Navigate to the Workout Analysis tab. 2. Review the recommendations provided. 3. Check if the recommendations are relevant to the provided workout data and align with user goals. | No specific postconditions. | The app generates personalized recommendations for workout improvements, adjusted to user's activity and goals. | Major | Medium | Functional Testing | Positive |

| TC_WA_005 | Verify that users can compare current workouts with past performance. | This test case confirms the ability to compare the current workout data with previous performance to highlight progress and identify areas for improvement. | Medium | User has completed multiple workouts. | Workout data for past and current workouts. | 1. Navigate to the Workout Analysis tab. 2. Select the current workout and a previous

workout for comparison. 3. Review the comparative data points (e.g., distance, duration, calorie burn) and visual representation. | No specific postconditions. | The app accurately compares current workout data with past performance, highlighting significant differences and trends. | Minor | Low | Functional Testing, Regression Testing | 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:

Fitness Tracker App - Weather Integration Test Cases

Test Case 1:

* **Test Case ID:** TC_WTH_001

* **Test Case Title:** Verify that the app displays current weather conditions accurately.

* **Test Case Description:** This test case aims to verify that the app correctly retrieves and displays current weather information such as temperature, humidity, and wind speed for the user's location.

* **Test Suite:** Weather Integration

* **Test Priority:** High

* **Preconditions:**

- * User has granted location permission to the app.

- * User is connected to the internet.

* **Test Data:** Valid user location

* **Test Steps:**

1. Launch the Fitness Tracker app.
2. Navigate to the weather section.
3. Verify that the app displays current weather conditions including temperature, humidity, wind speed, and weather description.
4. Compare the displayed data with a trusted weather source for accuracy.

* **Postconditions:** None

* **Expected Result:** The app accurately displays current weather conditions for the user's location.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** System Testing, Functional Testing

Test Case 2:

* **Test Case ID:** TC_WTH_002

* **Test Case Title:** Verify that the app allows users to view weather forecasts for different locations.

* **Test Case Description:** This test case verifies the functionality of selecting different locations to view weather forecasts within the app.

* **Test Suite:** Weather Integration

* **Test Priority:** Medium

* **Preconditions:**

* User is connected to the internet.

* **Test Data:** Valid city names from different countries

* **Test Steps:**

1. Launch the Fitness Tracker app.
2. Navigate to the weather section.
3. Search for a city name from a different country.
4. Verify that the app displays a weather forecast for the selected location.

* **Postconditions:** None

* **Expected Result:** The app successfully displays forecasts for various user-selected locations globally.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** System Testing, Functional Testing

Test Case 3:

* **Test Case ID:** TC_WTH_003

* **Test Case Title:** Verify that weather data is updated in real-time.

* **Test Case Description:** This test case ensures that weather information within the app is dynamically updated to reflect current conditions.

* **Test Suite:** Weather Integration

* **Test Priority:** High

* **Preconditions:**

- * User is connected to the internet.

- * Weather conditions are changing (e.g., temperature, wind speed).

* **Test Data:** No test data needed

* **Test Steps:**

1. Launch the Fitness Tracker app.

2. Observe the weather details in the app for a period of time (e.g., 15 minutes), noting any significant changes in weather conditions.

3. Compare the changes in displayed weather data with real-time weather updates from a reliable source (e.g., website, other weather app).

* **Postconditions:** None

* **Expected Result:** The app's weather data updates in real-time to accurately reflect changing weather conditions.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** System Testing, Functional Testing, Performance Testing

****Test Case 4:****

*** **Test Case ID:** TC_WTH_004**

*** **Test Case Title:** Verify that the app offers weather-based workout recommendations.**

*** **Test Case Description:** This test case evaluates if the app provides personalized workout suggestions based on the current weather conditions.**

*** **Test Suite:** Weather Integration**

*** **Test Priority:** Medium**

*** **Preconditions:****

*** User has created a workout profile within the app.**

*** User is connected to the internet.**

*** **Test Data:** Various weather conditions (sunny, cloudy, rainy, etc.)**

*** **Test Steps:****

1. Launch the Fitness Tracker app.

2. Access the workout recommendations section. Note the suggestions provided.

3. Manually change the weather conditions in the app to different scenarios (sunny, cloudy, rainy, etc.).

4. Observe any changes in the recommended workout types or intensities based on the simulated weather conditions.

*** **Postconditions:** None**

*** **Expected Result:** The app offers diverse workout recommendations that are contextually relevant to the prevailing weather conditions.**

*** **Severity:** Major**

*** **Priority:** Medium**

*** **Type of Testing:** System Testing, Functional Testing, Regression Testing**

****Test Case 5:****

*** **Test Case ID:** TC_WTH_005**

*** **Test Case Title:** Verify that the app sends weather alerts for extreme conditions.**

*** **Test Case Description:** This test case verifies if the app notifies users about potentially dangerous weather conditions like thunderstorms, heat waves, or severe cold.**

*** **Test Suite:** Weather Integration**

*** **Test Priority:** High**

*** **Preconditions:****

*** User has enabled weather alerts within the app settings.**

*** User is connected to the internet.**

*** Simulated extreme weather conditions are introduced into the app's data.**

*** **Test Data:** Simulated extreme weather conditions (e.g., tornado warning, blizzard alert)**

*** **Test Steps:****

1. Enable weather alerts for specific weather conditions in the app settings (e.g., thunderstorms, extreme heat).

2. Simulate specific extreme weather conditions in the app's data.

3. Verify that the app sends a corresponding weather alert notification to the user.

*** **Postconditions:** None**

*** **Expected Result:** The app accurately detects and notifies users about extreme weather conditions based on their configured alert preferences.**

*** **Severity:** Critical**

*** **Priority:** High**

*** **Type of Testing:** System Testing, Functional Testing**

Data - 98

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Mood Tracking

Feature description -

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

Userstory title -

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

Acceptance criteria -

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and energy levels.']

Generated Test Cases:

Fitness Tracker App: Mood Tracking Test Cases

****Test Case ID:** TC_Mood_01**

****Test Case Title:** Verify that users can log their mood before a workout.**

****Test Case Description:** This test case verifies the functionality of logging mood before a workout session.**

It ensures the user can select from a predefined list of mood options and accurately record their emotional state prior to exercise.

****Test Suite:** Mood Tracking**

****Test Priority:** High**

****Preconditions:****

- * User is logged into the Fitness Tracker app.
- * User has created at least one workout routine.

****Test Data:****

- * Sample mood options: Happy, Sad, Anxious, Calm, Energetic, Stressed

****Test Steps:****

1. Navigate to the workout logging screen.
2. Select a workout routine from the list.
3. Tap on the "Mood" field.
4. Select a mood from the predefined list.
5. Tap on "Save" or "Log Workout" to complete the workout log entry.

****Postconditions:****

- * The selected mood is accurately displayed next to the workout entry.
- * No error messages are displayed.

****Expected Result:**** The user's selected mood is successfully logged for the chosen workout routine.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, Smoke Testing

****Test Case Approach:**** Positive Testing

****Test Case ID:**** TC_Mood_02

****Test Case Title:**** Verify that users can log their energy levels before a workout.

****Test Case Description:**** This test case verifies the functionality of logging energy levels before a workout session. It ensures the user can select from a predefined scale for energy levels and accurately record their current energy state prior to exercise.

****Test Suite:**** Mood Tracking

****Test Priority:**** High

****Preconditions:****

- * User is logged into the Fitness Tracker app.
- * User has created at least one workout routine.

****Test Data:****

- * Energy Level Scale options: (0-10 with increments of 1)

****Test Steps:****

1. Navigate to the workout logging screen.
2. Select a workout routine from the list.
3. Tap on the "Energy Levels" field.
4. Choose an energy level on the predefined scale.
5. Tap on "Save" or "Log Workout" to complete the workout log entry.

****Postconditions:****

- * The selected energy level is accurately displayed next to the workout entry.
- * No error messages are displayed.

****Expected Result:**** The user's selected energy level is successfully logged for the chosen workout routine.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing, Smoke Testing

****Test Case Approach:** Positive Testing**

****Test Case ID:** TC_Mood_03**

****Test Case Title:** Verify that users receive insights into mood trends over time.**

****Test Case Description:** This test case verifies the app's ability to provide insights into user mood trends based on logged data. It checks if the app displays historical mood data visually (e.g., graphs) and highlights potential patterns.**

****Test Suite:** Mood Tracking**

****Test Priority:** Medium**

****Preconditions:****

- * User has logged mood data for at least 5 workouts.
- * The user has enabled data analytics or insights within the app settings.

****Test Data:****

- * User-defined mood data spanning several workouts (e.g., varying moods)

****Test Steps:****

1. Navigate to the "Mood Insights" section.
2. Examine the displayed visualizations (e.g., graphs, charts) of logged mood data.
3. Observe if the app highlights any trends or patterns in the user's mood over time.
4. Try filtering the insights by date range or workout type (if available).

****Postconditions:****

* The app displays personalized mood insights based on logged data.

* No error messages are displayed.

****Expected Result:**** The app accurately displays historical mood data visually, highlighting potential trends and patterns in the user's mood over time.

****Severity:**** Major

****Priority:**** Medium

****Type of Testing:**** Functional Testing, Performance Testing

****Test Case Approach:**** Positive Testing

Let me know if you'd like me to continue generating more test cases! I'm happy to help you craft a comprehensive suite for your Fitness Tracker app.

Data - 99

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary

inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Creation

Feature description -

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

Userstory title -

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

Acceptance criteria -

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', 'Recipes are saved in the user's profile.', 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

Generated Test Cases:

Test Cases for Recipe Creation Feature

****Test Case ID:**** RC-001

****Test Case Title:**** Verify that users can input and save ingredients for their recipes.

****Test Case Description:**** This test case verifies the functionality of inputting ingredients for a recipe and saving them successfully.

****Test Suite:**** Recipe Feature

****Test Priority:**** High

****Preconditions:****

* User is logged in to the platform.

* A new recipe is initiated for creation.

****Test Data:****

* List of ingredients with varying names and quantities.

****Test Steps:****

1. Navigate to the recipe creation page.
2. Enter various ingredients into the designated ingredient field, including names and quantities.
3. Verify that the entered ingredients are displayed correctly in the recipe details.
4. Proceed to save the recipe.
5. Verify that the saved recipe includes the entered ingredients.

****Postconditions:****

* Recipe is saved in the user's profile.

****Expected Result:**** All entered ingredients are accurately saved and displayed within the created recipe.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional, Integration

****Test Case Approach:**** Positive

****Test Case ID:**** RC-002

****Test Case Title:**** Verify that users can input and save cooking instructions for their recipes.

****Test Case Description:**** This test case checks if users can enter cooking instructions and successfully save

them with the recipe.

****Test Suite:**** Recipe Feature

****Test Priority:**** High

****Preconditions:****

* User is logged in to the platform.

* A new recipe is initiated for creation.

****Test Data:****

* Detailed step-by-step cooking instructions for a recipe.

****Test Steps:****

1. Navigate to the recipe creation page.
2. Input the cooking instructions for the recipe in the designated text area.
3. Verify that the entered instructions are displayed correctly in the recipe details.
4. Proceed to save the recipe.
5. Verify that the saved recipe includes the entered cooking instructions.

****Postconditions:****

* Recipe is saved in the user's profile.

****Expected Result:**** All entered cooking instructions are accurately saved and displayed within the created recipe.

****Severity:**** Critical

****Priority:**** High

****Type of Testing:**** Functional, Integration

****Test Case Approach:** Positive**

****Test Case ID:** RC-003**

****Test Case Title:** Verify that users can upload photos for their recipes.**

****Test Case Description:** This test case validates the ability of users to upload photos associated with their recipes.**

****Test Suite:** Recipe Feature**

****Test Priority:** High**

****Preconditions:****

- * User is logged in to the platform.
- * A new recipe is initiated for creation.
- * A valid image file is available on the user's device.

****Test Data:****

- * A valid image file (e.g., JPG, PNG)

****Test Steps:****

1. Navigate to the recipe creation page.
2. Locate the designated photo upload field.
3. Select and upload the valid image file from the user's device.
4. Verify that the uploaded photo is displayed correctly in the recipe details.
5. Proceed to save the recipe.
6. Verify that the saved recipe includes the uploaded photo.

****Postconditions:****

* Recipe is saved in the user's profile.

****Expected Result:**** The uploaded photo is successfully associated with the created recipe and displayed accordingly.

****Severity:**** High

****Priority:**** High

****Type of Testing:**** Functional, Integration

****Test Case Approach:**** Positive

****Test Case ID:**** RC-004

****Test Case Title:**** Verify that recipes are accessible to other users on the platform.

****Test Case Description:**** This test case checks if publicly shared recipes from different users are accessible on the platform.

****Test Suite:**** Recipe Feature

****Test Priority:**** Medium

****Preconditions:****

* Users A and B are logged in to the platform.

* User A creates a publicly shareable recipe.

****Test Data:****

* One or more recipes created by user A

****Test Steps:****

1. User B navigates to the recipe listing or search page.
2. User B searches for or browses recipes (or the specific recipe from user A).
3. User B successfully loads and views recipes created by user A.

****Postconditions:****

* User B can access and view user A's recipes

****Expected Result:**** User B can access and view all publicly available recipes created by User A.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional, System

****Test Case Approach:**** Positive

****Test Case ID:**** RC-005

****Test Case Title:**** Verify that users receive a confirmation upon successful recipe submission.

****Test Case Description:**** This test case validates that users are notified upon successfully saving a new recipe.

****Test Suite:**** Recipe Feature

****Test Priority:**** Medium

****Preconditions:****

* User is logged in to the platform.

* A new recipe is initiated for creation.

****Test Data:****

* Sample recipe data

****Test Steps:****

1. User enters recipe data and clicks the "save" or "submit" button.
2. Observe the response from the platform after clicking the "save" button.

****Postconditions:****

* The recipe is saved successfully.

****Expected Result:**** The user receives an appropriate confirmation message confirming that the recipe was successfully submitted.

****Severity:**** Minor

****Priority:**** Medium

****Type of Testing:**** Functional

****Test Case Approach:**** Positive

Data - 100

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Discovery

Feature description -

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

Userstory title -

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance criteria -

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

Generated Test Cases:

Recipe Sharing Platform - Test Cases for Recipe Discovery

****User Story:**** As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

****Test Case ID:**** RC-001

****Test Case Title:**** Verify that users can search for recipes using keywords

****Test Case Description:**** This test case aims to verify that the recipe search functionality effectively retrieves recipes based on provided keywords.

****Test Suite:**** Recipe Discovery

****Test Priority:**** High

****Preconditions:****

* User is logged in or logged out

* The recipe database contains a sufficient number of recipes.

****Test Data:**** "Pasta", "Chicken", "Vegetarian"

****Test Steps:****

1. Navigate to the recipe search page.
2. Enter the keyword "Pasta" in the search field and click "Search".
3. Verify that the search results display recipes containing the keyword "Pasta".
4. Repeat steps 2 & 3 with the keywords "Chicken" and "Vegetarian".
5. Verify that the search results accurately match the provided keywords.

****Postconditions:****

* No changes required to the application state after testing

****Expected Result:**** Search results will return relevant recipes containing the input keywords.

****Severity:**** Major

****Priority:**** High

****Type of Testing:**** Functional Testing

****Test Case Approach:**** Positive

****Test Case ID:** RC-002**

****Test Case Title:** Verify that users can save search results for future reference**

****Test Case Description:** This test case verifies the functionality of saving search results for re-accessing later.**

****Test Suite:** Recipe Discovery**

****Test Priority:** Medium**

****Preconditions:****

* User is logged in.

* A search has been conducted with specific keywords.

****Test Data:** "Vegan Cookies"**

****Test Steps:****

1. Navigate to the recipe search page.
2. Enter the keyword "Vegan Cookies" in the search field and click "Search".
3. Click the "Save Search" button (or equivalent).
4. Verify that the saved search is accessible from the user's profile or a designated saved searches section.
5. Retrieve the saved search and verify that it displays the previously searched "Vegan Cookies" recipes.

****Postconditions:** Saved search is accessible.**

****Expected Result:** Saved search will be accessible and retrieve the original search results**

****Severity:** Major**

****Priority:** Medium**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** RC-003**

****Test Case Title:** Verify that users can filter search results by cuisine type**

****Test Case Description:** This test case confirms the functionality of filtering search results based on cuisine type.**

****Test Suite:** Recipe Discovery**

****Test Priority:** High**

****Preconditions:****

* User is logged in or logged out

* The recipe database contains recipes categorized by cuisine types.

****Test Data:** "Italian", "Indian", "American"**

****Test Steps:****

1. Navigate to the recipe search page.
2. Enter a generic search term, such as "Stir-fry".
3. Select "Italian" from the cuisine filter dropdown menu.
4. Verify that the search results display only Italian stir-fry recipes.
5. Repeat steps 3 & 4 with the cuisine types "Indian" and "American".

****Postconditions:** No changes to the application state besides filtering.**

****Expected Result:** Filters will effectively narrow down search results based on selected cuisine types.**

****Severity:** Major**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** RC-004**

****Test Case Title:** Verify that users can filter search results by dietary preferences**

****Test Case Description:** This test case aims to ensure accurate filtering of recipes based on user-selected dietary preferences.**

****Test Suite:** Recipe Discovery**

****Test Priority:** High**

****Preconditions:****

- * User is logged in or logged out
- * The recipe database contains recipes categorized by dietary preferences (e.g., vegetarian, vegan, gluten-free, etc.).

****Test Data:** "Vegetarian", "Vegan", "Gluten-Free"**

****Test Steps:****

1. Navigate to the recipe search page.
2. Enter a generic search term, such as "Soup".
3. Select "Vegetarian" from the dietary preference filter dropdown menu.
4. Verify that the search results display only vegetarian soup recipes.
5. Repeat steps 3 & 4 with the dietary preferences "Vegan" and "Gluten-Free".

****Postconditions:** No changes to the application state besides dietary preference filtering.**

****Expected Result:** Filters will accurately display recipes matching the selected dietary preferences.**

****Severity:** Major**

****Priority:** High**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Positive**

****Test Case ID:** RC-005**

****Test Case Title:** Verify that users encounter appropriate error messages for invalid search inputs**

****Test Case Description:** This test case ensures the system provides clear error handling for invalid search inputs.**

****Test Suite:** Recipe Discovery**

****Test Priority:** Medium**

****Preconditions:****

* User is logged in or logged out

****Test Data:** "1234", "", " ",**

****Test Steps:****

1. Navigate to the recipe search page.
2. Enter the invalid search inputs "1234", "", " " respectively in the search field and click "Search".
3. Verify that an appropriate error message is displayed, indicating the invalid input.

****Postconditions:** Error message is displayed, no unintended actions occur.**

****Expected Result:** The system should display informative error messages for invalid search inputs,**

guiding the user to enter valid data.

****Severity:** Minor**

****Priority:** Medium**

****Type of Testing:** Functional Testing**

****Test Case Approach:** Negative**
