

# Generated Test Cases via gemini-1.5-flash 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 Profiles Feature

**\*\*Project Name:\*\* Social Media Platform**

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

**\*\*Test Case Creation Template:\*\***

<b>**Test Case ID**</b>	<b>**Test Case Title**</b>	<b>**Test Case Description**</b>	<b>**Test Suite**</b>	<b>**Test Priority**</b>	<b>**Preconditions**</b>	<b>**Test Data**</b>	<b>**Test Steps**</b>	<b>**Postconditions**</b>	<b>**Expected Result**</b>	<b>**Severity**</b>	<b>**Priority**</b>	<b>**Type of Testing**</b>	<b>**Test Case Approach**</b>
---	---	---	---	---	---	---	---	---	---	---	---	---	---
TC_UP_01	Verify that users can add and edit their name	This test case checks the functionality of adding and editing a user's name in their profile. It verifies that the system correctly updates the name in the user's profile.	User Profile	High	User is logged in	Name: "John Doe" (initial), "Jane Doe" (updated)	1. Navigate to the "Profile" page. 2. Click on the "Edit Profile" button. 3. Enter the new name "Jane Doe" in the name field. 4. Click on the "Save" button.	The profile is updated with the new name "Jane Doe".	The user's profile displays the correct updated name "Jane Doe".	Minor	High	Functional Testing	Positive
TC_UP_02	Verify that users can add and edit their bio	This test case checks the functionality of adding and editing a user's bio in their profile. It verifies that the system correctly updates the bio in the user's profile.	User Profile	High	User is logged in	Bio: "Software Engineer" (initial), "Creative Writer" (updated)	1. Navigate to the "Profile" page. 2. Click on the "Edit Profile" button. 3. Enter the new bio "Creative Writer" in the bio field. 4. Click on the "Save" button.	The profile is updated with the new bio "Creative Writer".	The user's profile displays the correct updated bio "Creative Writer".	Minor	High	Functional Testing	Positive
TC_UP_03	Verify that the profile picture supports standard image formats	This test case checks the functionality of uploading profile pictures in supported image formats. It verifies that the system accepts the specified image formats and rejects unsupported ones.	User Profile	High	User is logged in	Image formats: JPG, PNG, GIF (supported), BMP, TIF (unsupported)	1. Navigate to the "Profile" page. 2. Click on the "Edit						

Profile" button. 3. Attempt to upload a profile picture in JPG, PNG, GIF, BMP, and TIF formats. | The system should accept JPG, PNG, and GIF formats and reject BMP and TIF formats. | The profile picture is successfully uploaded when using JPG, PNG, and GIF formats. The system displays an error message when attempting to upload BMP and TIF formats. | Minor | High | Functional Testing | Negative |

| TC\_UP\_04 | Verify that users can set privacy settings for each field | This test case checks the functionality of setting privacy settings for individual profile fields. It verifies that the system allows users to control the visibility of their information. | User Profile | High | User is logged in | Privacy settings: Public, Friends, Private | 1. Navigate to the "Profile" page. 2. Click on the "Edit Profile" button. 3. Select different privacy settings for each profile field (Name, Bio, Location, Interests). | The system correctly applies the selected privacy settings for each field. | The selected privacy settings are reflected in the user's profile, restricting visibility as expected. | Minor | High | Functional Testing | Positive |

| TC\_UP\_05 | Verify that users receive a confirmation upon saving their profile | This test case checks that a confirmation message is displayed upon successful saving of the profile. It verifies that the user is informed of the successful update to their profile. | User Profile | High | User is logged in | No test data needed | 1. Navigate to the "Profile" page. 2. Click on the "Edit Profile" button. 3. Make changes to any profile field. 4. Click on the "Save" button. | The system displays a confirmation message indicating that the profile has been saved successfully. | A confirmation message is displayed after saving the profile changes. | Minor | High | Functional Testing | Positive |

-----

## **Data - 2**

### **Project name -**

Social Media Platform

### **Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending

topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

### **Feature name -**

News Feed

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

## Test Cases for News Feed Feature

**\*\*Project Name:\*\* Social Media Platform**

**\*\*Feature Name:\*\* News Feed**

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

**\*\*Acceptance Criteria:\*\***

1. The news feed displays posts from followed accounts.
2. Content is shown in chronological or algorithmic order.

3. Users can interact with posts (like, comment, share).
4. The feed updates in real-time.
5. Users can filter content based on type or topic.

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC\_NF\_01**

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

**\*\*Test Case Description:\*\* This test case checks if the news feed successfully displays posts from accounts the user follows.**

**\*\*Test Suite:\*\* News Feed**

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

**\*\*Preconditions:\*\***

- User account is logged in.
- User has followed at least one account.

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

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

1. Log in to the social media platform using a test user account.
2. Navigate to the news feed section.
3. Observe the displayed posts.
4. Verify that the posts are from accounts the user is following.

**\*\*Postconditions:\*\* Log out of the test user account.**

**\*\*Expected Result:\*\* The news feed should display posts only from accounts that the user has followed.**

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

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

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

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

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TC\_NF\_02**

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

**\*\*Test Case Description:\*\*** This test case verifies if the posts are displayed in the expected order, either chronological or based on an algorithm.

**\*\*Test Suite:\*\*** News Feed

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

**\*\*Preconditions:\*\***

- User account is logged in.
- User has followed at least two accounts.
- Each followed account has made at least one post.

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

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

1. Log in to the social media platform using a test user account.
2. Navigate to the news feed section.
3. Observe the order of the posts.
4. Check if the posts are sorted by the most recent first or by algorithm-based ranking.
5. Verify that the order remains consistent when scrolling down the feed.

**\*\*Postconditions:\*\*** Log out of the test user account.

**\*\*Expected Result:\*\*** The posts should be displayed in either chronological or algorithmic order, with the most recent or highest-ranked posts appearing first.

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

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

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

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

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* TC\_NF\_03**

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

**\*\*Test Case Description:\*\* This test case ensures users can perform common interactions like liking, commenting, and sharing on posts in the news feed.**

**\*\*Test Suite:\*\* News Feed**

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

**\*\*Preconditions:\*\***

- User account is logged in.
- User is viewing a post in the news feed.

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

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

1. Log in to the social media platform using a test user account.
2. Navigate to the news feed and select a post.
3. Perform the following interactions:
  - Click the "Like" button.
  - Add a comment to the post.
  - Click the "Share" button.
4. Verify that the interactions are successfully applied to the selected post.

**\*\*Postconditions:\*\* Log out of the test user account.**

**\*\*Expected Result:\*\* The post should display the user's interactions correctly, showing the like, comment, and share actions.**

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

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

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

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

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* TC\_NF\_04**

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

**\*\*Test Case Description:\*\* This test case verifies that the news feed dynamically updates with new content as it is posted by followed accounts.**

**\*\*Test Suite:\*\* News Feed**

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

**\*\*Preconditions:\*\***

- User account is logged in.
- User is viewing the news feed.
- Another user (following the test user) posts new content.

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

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

1. Log in to the social media platform using a test user account.
2. Navigate to the news feed and observe the current content.
3. Have another user (following the test user) post new content.
4. Observe the news feed.
5. Verify that the new post appears in the news feed without refreshing the page.

**\*\*Postconditions:\*\* Log out of the test user account.**

**\*\*Expected Result:\*\* The news feed should refresh automatically and display the new post without user intervention.**

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

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

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



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

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC\_NF\_05**

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

**\*\*Test Case Description:\*\* This test case ensures that users can filter the displayed content in the news feed based on various criteria, such as post type or topic.**

**\*\*Test Suite:\*\* News Feed**

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

**\*\*Preconditions:\*\***

- User account is logged in.
- User is viewing the news feed.
- The platform supports content filtering by type or topic.

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

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

1. Log in to the social media platform using a test user account.
2. Navigate to the news feed.
3. Locate the content filtering options (e.g., drop-down menu, filter buttons).
4. Select a specific type or topic filter (e.g., photos, videos, technology, travel).
5. Observe the displayed content.
6. Verify that the news feed only shows posts matching the selected filter.

**\*\*Postconditions:\*\* Log out of the test user account.**

**\*\*Expected Result:\*\* The news feed should only display content matching the selected filter criteria (type or topic).**

**\*\*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

\*\*Project:\*\* Social Media Platform

\*\*Feature:\*\* Messaging

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

\*\*Acceptance Criteria:\*\*

- 1. Users can send text messages, images, and videos.
- 2. Messages can be sent in real-time.
- 3. Users can start private conversations or group chats.
- 4. Messages should show read receipts.
- 5. Users should be notified of new messages.

\*\*Test Case 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-MSG-01	Verify that users can send text messages, images, and videos.	This test case verifies that users can send different types of messages, including text messages, images, and videos, in private conversations.	Messaging	High	User A and User B have active accounts.	Text message, image file, video file	1. User A opens a private conversation with User B.   2. User A sends a text message.   3. User A sends an image file.   4. User A sends a video file.   5. User A confirms that all sent messages are displayed correctly.	No postconditions.	All messages are sent and displayed correctly.	Major	High	Functional Testing	Positive

| TC-MSG-02 | Verify that messages are sent in real-time. | This test case verifies that messages are delivered and displayed to the recipient in real-time, with minimal delay. | Messaging | High | User A and User B have active accounts. User A and User B are online and connected to the platform. | Text message | 1. User A opens a private conversation with User B. <br> 2. User A sends a text message. <br> 3. User B confirms that the message is received and displayed within 5 seconds. | No postconditions. | The message is displayed to User B within 5 seconds of being sent by User A. | Major | High | Functional Testing | Positive |

| TC-MSG-03 | Verify that users can start private conversations or group chats. | This test case verifies that users can initiate private conversations and create group chats with other users. | Messaging | High | User A, User B, and User C have active accounts. | No test data needed. | 1. User A clicks on the "New Message" button. <br> 2. User A searches for User B and starts a private conversation. <br> 3. User A searches for User B and User C and creates a group chat. | No postconditions. | Private conversation and group chat are successfully created. | Major | High | Functional Testing | Positive |

| TC-MSG-04 | Verify that messages show read receipts. | This test case verifies that the sender receives confirmation when the recipient has read the message. | Messaging | Medium | User A and User B have active accounts. User A and User B are online and connected to the platform. | Text message | 1. User A sends a text message to User B. <br> 2. User B reads the message. <br> 3. User A confirms that a read receipt is displayed indicating that the message has been read. | No postconditions. | Read receipt is displayed when the message is read by the recipient. | Minor | Medium | Functional Testing | Positive |

| TC-MSG-05 | Verify that users are notified of new messages. | This test case verifies that users receive notifications when they receive new messages. | Messaging | High | User A and User B have active accounts. User A sends a message to User B while User B is offline. | Text message | 1. User A sends a text message to User B while User B is offline. <br> 2. User B logs back onto the platform. <br> 3. User B confirms that a notification is displayed indicating a new message from User A. | No postconditions. | Notification is displayed when a new message is received. | Major | High | Functional Testing | Positive |

-----

## **Project name -**

Social Media Platform

## **Project description -**

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

## **Feature name -**

Content Sharing

## **Feature description -**

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

## **Userstory title -**

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

## **Acceptance criteria -**

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

## **Generated Test Cases:**

## Test Cases for Content Sharing Feature

**\*\*Project:\*\*** Social Media Platform

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

**\*\*Acceptance Criteria:\*\***

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

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-CS-01

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

**\*\*Test Case Description:\*\*** This test case verifies the functionality of creating and sharing text posts on the social media platform.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

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

\* Text post content: "This is a test post about content sharing."

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

1. Navigate to the "Create Post" section.
2. Enter the text post content in the provided field.
3. Select the desired privacy setting (followers or public).
4. Click on the "Share" button.

**\*\*Postconditions:\*\***

\* The text post appears on the user's profile and timeline.

**\*\*Expected Result:\*\*** The text post is successfully created and shared according to the selected privacy setting.

**\*\*Severity:\*\*** Blocker

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

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

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

**\*\*Test Case ID:\*\*** TC-CS-02

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

**\*\*Test Case Description:\*\*** This test case verifies the ability of users to share photos and videos on the platform.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* A photo and a video are available for sharing.

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

\* Photo and video files.

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

1. Navigate to the "Create Post" section.
2. Select the "Photo" or "Video" option.
3. Upload the desired photo or video file.
4. Select the desired privacy setting (followers or public).
5. Click on the "Share" button.

**\*\*Postconditions:\*\***

\* The shared photo or video appears on the user's profile and timeline.

**\*\*Expected Result:\*\*** The photo or video is successfully uploaded, shared, and displayed on the user's timeline and profile.

**\*\*Severity:\*\*** Blocker

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

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

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

**\*\*Test Case ID:\*\*** TC-CS-03

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

**\*\*Test Case Description:\*\*** This test case validates the ability of users to modify or remove their previously shared posts.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* A post has been previously created and shared by the user.

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

\* No test data needed

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

1. Navigate to the previously shared post on the user's timeline.
2. Click on the "Edit" option for the post.
3. Modify the post content, privacy settings, or other available options.
4. Click on the "Save" button.
5. Verify the changes made to the post.
6. Click on the "Delete" option for the post.



7. Confirm the deletion of the post.

**\*\*Postconditions:\*\***

\* The post is successfully edited and saved with the modified content.

\* The post is deleted from the user's timeline and profile.

**\*\*Expected Result:\*\*** The user is able to successfully edit and delete their previously shared posts.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-CS-04

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

**\*\*Test Case Description:\*\*** This test case ensures that posts can include hashtags and tagging to categorize and connect content with other users.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

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

\* Text post content: "This is a test post about #contentsharing and tagging @username."

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

1. Navigate to the "Create Post" section.

2. Enter the test post content including hashtags and tagging.

3. Click on the "Share" button.

**\*\*Postconditions:\*\***

\* The shared post displays the hashtags and tags correctly.

**\*\*Expected Result:\*\*** The hashtags and tags are successfully implemented in the post and are clickable, linking to relevant content or user profiles.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-CS-05

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

**\*\*Test Case Description:\*\*** This test case verifies that users receive feedback on their shared posts in the form of likes, comments, and shares.

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

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

**\*\*Preconditions:\*\***

- \* User is logged in to the platform.
- \* A post has been previously created and shared by the user.
- \* Another user is logged in and can interact with the post.

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

- \* No test data needed

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

1. The other user likes, comments on, and shares the post.
2. The user checks their post for notifications.

**\*\*Postconditions:\*\***

- \* The user receives notifications for likes, comments, and shares on their post.

**\*\*Expected Result:\*\*** Users receive real-time notifications for feedback on their shared content, including likes, comments, and shares.

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

**\*\*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 Social Media Platform - Privacy Settings

\*\*Project Context:\*\*

\* \*\*Project Name:\*\* Social Media Platform

\* \*\*Feature Name:\*\* Privacy Settings

\* \*\*User Story Title:\*\* As a user, I want to set privacy settings so that I can control who sees my information.

\*\*Acceptance Criteria:\*\*

- 1. Users can set their profile to public, private, or restricted.
- 2. Privacy settings can be changed at any time.
- 3. Users can control visibility of individual posts.
- 4. The system should respect the privacy settings immediately.
- 5. Users should be notified of any changes in privacy settings.

\*\*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-PS-01	Verify that users can set their profile to public, private, or restricted	This test case verifies the user's ability to choose different privacy settings for their profile.	Privacy Settings	High	User is logged in	No test data needed	1. Access the user's profile settings page. 2. Locate the privacy settings section. 3. Select "Public" privacy setting. 4. Select "Private" privacy setting. 5. Select "Restricted" privacy setting.	The profile					

settings are updated with the selected privacy setting. | The profile is visible according to the selected privacy setting. | Major | High | Functional Testing | Positive |

| TC-PS-02 | Verify that privacy settings can be changed at any time | This test case verifies that users can modify their privacy settings whenever they want. | Privacy Settings | High | User is logged in, profile privacy is set to "Public" | No test data needed | 1. Access the user's profile settings page. 2. Locate the privacy settings section. 3. Select "Private" privacy setting. 4. Access the user's profile page. | The profile is now visible only to approved users. | The profile privacy setting is updated to "Private". | Major | High | Functional Testing | Positive |

| TC-PS-03 | Verify that users can control visibility of individual posts | This test case verifies that users can set different privacy settings for each post they share. | Privacy Settings | High | User is logged in, a post is created | No test data needed | 1. Create a new post. 2. Select "Public" privacy setting for the post. 3. Create another new post. 4. Select "Private" privacy setting for the post. | The posts are visible according to their selected privacy settings. | Both posts are displayed correctly with the respective privacy settings. | Major | High | Functional Testing | Positive |

| TC-PS-04 | Verify that the system immediately respects the privacy settings | This test case verifies that the system applies the selected privacy settings without delay. | Privacy Settings | High | User is logged in, profile privacy is set to "Private" | Another user account | 1. Access the other user's profile page. 2. Attempt to view the user's profile. | The user is denied access to the profile due to the privacy setting. | Access is denied to users who are not approved, confirming immediate implementation of privacy settings. | Critical | High | Functional Testing | Negative |

| TC-PS-05 | Verify that users are notified of any changes in privacy settings | This test case verifies that users receive a notification when they modify their privacy settings. | Privacy Settings | Medium | User is logged in, profile privacy is set to "Public" | No test data needed | 1. Access the user's profile settings page. 2. Change the privacy setting to "Private". | The user receives a notification regarding the change in their privacy setting. | The notification clearly states the change in privacy setting and its implications. | Minor | Medium | Functional Testing | Positive |

---

## Data - 6

### Project name -

Social Media Platform

### Project description -

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

### Feature name -

Notifications

### Feature description -

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

### Userstory title -

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

### Acceptance criteria -

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

### Generated Test Cases:

## Test Cases for Social Media Platform Notifications Feature

\*\*Project Name:\*\* Social Media Platform

**\*\*Feature Name:\*\* Notifications**

**\*\*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 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\_NOT\_01 | Verify that users receive notifications for likes on their posts. | This test case verifies that users are notified when someone likes their posts. The test simulates a user liking a post made by another user and checks if the post owner receives a notification. | Notifications | High | User A has created a post and User B is a friend of User A. | No test data needed | 1. User B likes a post made by User A. 2. Observe User A's notification feed. 3. Verify that the notification includes information about the post liked, User B's name, and a link to the post. | User A should receive a notification about the like. | User A does not receive a notification. | Major | High | Functional Testing | Positive

TC\_NOT\_02 | Verify that users receive notifications for comments on their posts. | This test case validates that users are notified when someone comments on their posts. The test involves a user commenting on a post and verifying if the post owner receives a notification. | Notifications | High | User A has created a post and User B is a friend of User A. | No test data needed | 1. User B comments on a post made by User A. 2. Observe User A's notification feed. 3. Verify that the notification includes information about the post, the comment content, User B's name, and a link to the post. | User A should receive a notification about the comment. | User A does not receive a notification. | Major | High | Functional Testing | Positive

TC\_NOT\_03 | Verify that users receive notifications for shares on their posts. | This test case verifies that users are notified when someone shares their posts. The test simulates a user sharing a post made by another

user and checks if the post owner receives a notification. | Notifications | High | User A has created a post and User B is a friend of User A. | No test data needed | 1. User B shares a post made by User A. 2. Observe User A's notification feed. 3. Verify that the notification includes information about the post, User B's name, and a link to the post. | User A should receive a notification about the share. | User A does not receive a notification. | Major | High | Functional Testing | Positive

TC\_NOT\_04 | Verify that users are notified of new friend requests. | This test case validates that users receive a notification when someone sends them a friend request. The test involves a user sending a friend request to another user and verifying if the recipient receives a notification. | Notifications | High | User A and User B are not friends. | No test data needed | 1. User A sends a friend request to User B. 2. Observe User B's notification feed. 3. Verify that the notification includes information about User A's profile and a link to the friend request. | User B should receive a notification about the friend request. | User B does not receive a notification. | Major | High | Functional Testing | Positive

TC\_NOT\_05 | Verify that users receive notifications for mentions in posts or comments. | This test case validates that users are notified when someone mentions them in a post or comment. The test involves a user mentioning another user in a post and verifying if the mentioned user receives a notification. | Notifications | High | User A and User B are friends. | No test data needed | 1. User A creates a post and mentions User B in the post. 2. Observe User B's notification feed. 3. Verify that the notification includes information about the post, User A's name, and a link to the post. | User B should receive a notification about the mention. | User B does not receive a notification. | Major | High | Functional Testing | 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 Search Functionality on Social Media Platform

**\*\*Project Name:\*\* Social Media Platform**

**\*\*Feature Name:\*\* Search Functionality**

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

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC-SEARCH-01**

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

**\*\*Test Case Description:\*\*** This test case verifies that users can search for content on the platform using keywords or hashtags. The test case involves searching for specific keywords and hashtags and verifying the results displayed.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* The platform has content indexed for search.

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

\* Keyword: "travel"

\* Hashtag: #travel

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

1. Go to the search bar on the platform.

2. Enter the keyword "travel" in the search bar.

3. Press Enter or click the search button.

4. Verify that the search results page displays posts, profiles, and topics related to "travel".

5. Repeat steps 2-4 using the hashtag #travel.

6. Verify that the search results page displays posts, profiles, and topics related to the hashtag #travel.

**\*\*Postconditions:\*\***

\* The user remains on the search results page.

**\*\*Expected Result:\*\*** The search results page displays relevant content related to the entered keyword or hashtag.

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

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

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

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

**\*\*Test Case ID:\*\* TC-SEARCH-02**

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

**\*\*Test Case Description:\*\*** This test case ensures that the search functionality returns results for profiles, posts, and topics related to the search query.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* The platform has content indexed for search.

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

\* Keyword: "photography"

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

1. Go to the search bar on the platform.
2. Enter the keyword "photography" in the search bar.
3. Press Enter or click the search button.
4. Verify that the search results page displays a mix of profiles, posts, and topics related to "photography".

**\*\*Postconditions:\*\***

\* The user remains on the search results page.

**\*\*Expected Result:\*\*** The search results page displays relevant profiles, posts, and topics related to "photography".

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

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

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

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

**\*\*Test Case ID:\*\*** TC-SEARCH-03

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

**\*\*Test Case Description:\*\*** This test case checks whether the search results are displayed in order of relevance, based on factors like popularity, recent activity, and user interactions.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* The platform has content indexed for search.

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

\* Keyword: "music"

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

1. Go to the search bar on the platform.
2. Enter the keyword "music" in the search bar.
3. Press Enter or click the search button.
4. Observe the order of the results displayed.
5. Verify that the most relevant results appear at the top, based on factors like popularity, recent activity, and user interactions.

**\*\*Postconditions:\*\***

\* The user remains on the search results page.

**\*\*Expected Result:\*\*** The search results are displayed in order of relevance, with the most relevant results appearing at the top.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-SEARCH-04

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

**\*\*Test Case Description:\*\*** This test case verifies that users can filter search results based on different content types, such as posts, profiles, or topics.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* The platform has content indexed for search.

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

\* Keyword: "fashion"

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

1. Go to the search bar on the platform.
2. Enter the keyword "fashion" in the search bar.
3. Press Enter or click the search button.
4. Locate the filter options on the search results page.
5. Select the "Posts" filter.
6. Verify that the search results page now only displays posts related to "fashion".
7. Repeat steps 5-6 for the "Profiles" and "Topics" filters.

**\*\*Postconditions:\*\***

\* The user remains on the search results page.

**\*\*Expected Result:\*\*** The search results page displays only content of the selected type.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-SEARCH-05

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

**\*\*Test Case Description:\*\*** This test case ensures that the search functionality provides suggestions as the user types in the search bar, helping them find relevant terms and refine their search.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* The platform has content indexed for search.

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

\* No test data needed.

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

1. Go to the search bar on the platform.
2. Start typing a keyword, such as "food".
3. Observe the suggestions that appear below the search bar as you type.
4. Verify that the suggestions are relevant to the keyword "food" and include related keywords, hashtags, or accounts.

**\*\*Postconditions:\*\***

\* The user remains on the search page.

**\*\*Expected Result:\*\*** Relevant suggestions appear below the search bar as the user types, helping them refine their search.

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

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

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

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

-----

## **Project name -**

Social Media Platform

## **Project description -**

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

## **Feature name -**

Hashtags and Tagging

## **Feature description -**

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

## **Userstory title -**

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

## **Acceptance criteria -**

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

## **Generated Test Cases:**

## Test Cases for Hashtags and Tagging Feature

**\*\*Project Name:\*\* Social Media Platform**

**\*\*Feature Name:\*\* Hashtags and Tagging**

**\*\*User Story 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:\*\***

- 1. Users can add hashtags to their posts.
- 2. Users can tag other profiles, brands, or locations.
- 3. Hashtags and tags are clickable and lead to related content.
- 4. Users can search for content using hashtags.
- 5. The system should suggest popular hashtags as users type.

**\*\*Test Case 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_01	Verify that users can add hashtags to their posts	This test case verifies that users can add hashtags to their posts by creating a new post, entering a valid hashtag, and checking if it is displayed correctly on the post.	Hashtags	High	User is logged in and has the ability to create posts.	#ExampleHashtag	1. Navigate to the post creation page. 2. Enter text in the post editor. 3. Type #ExampleHashtag in the post editor. 4. Click the "Post" button. 5. Verify that #ExampleHashtag is displayed correctly in the post.	The user's post is published.	The hashtag #ExampleHashtag is displayed correctly in the post and is clickable.	Minor	High	Functional Testing	Positive
TC_02	Verify that users can tag other profiles, brands, or locations	This test case verifies that users can tag other profiles, brands, or locations by creating a new post, entering a valid tag, and checking if it is displayed correctly on the post.	Tagging	High	User is logged in and has the ability to create posts.	@ExampleProfile, @ExampleBrand, @ExampleLocation	1. Navigate to the post creation page. 2. Enter text in the post editor. 3. Type @ExampleProfile in the post editor. 4. Select the correct profile from the						



suggestions. 5. Click the "Post" button. 6. Verify that @ExampleProfile is displayed correctly in the post and links to the profile. 7. Repeat steps 2-6 for @ExampleBrand and @ExampleLocation. | The user's post is published. | All tags (@ExampleProfile, @ExampleBrand, @ExampleLocation) are displayed correctly in the post and are clickable, linking to the respective profiles, brands, or locations. | Minor | High | Functional Testing | Positive |

| TC\_03 | Verify that hashtags and tags are clickable and lead to related content | This test case verifies that hashtags and tags are clickable and lead to a page displaying related content by clicking on a hashtag or tag in a post and checking if the page displays relevant content. | Clickability | High | User is logged in and has access to a post with a hashtag or tag. | #ExampleHashtag, @ExampleProfile | 1. Navigate to a post containing #ExampleHashtag or @ExampleProfile. 2. Click on #ExampleHashtag or @ExampleProfile. 3. Verify that the user is redirected to a page displaying content related to the clicked hashtag or tag. | User is redirected to a new page. | The page displays a list of posts, users, or other content relevant to the clicked hashtag or tag. | Minor | High | Functional Testing | Positive |

| TC\_04 | Verify that users can search for content using hashtags | This test case verifies that users can search for content using hashtags by searching for a specific hashtag and checking if the search results display relevant content. | Search Functionality | High | User is logged in and has access to the search bar. | #ExampleHashtag | 1. Navigate to the search bar. 2. Enter #ExampleHashtag in the search bar. 3. Click the "Search" button. 4. Verify that the search results display posts, users, or other content related to #ExampleHashtag. | Search results are displayed. | The search results display content relevant to the entered hashtag #ExampleHashtag. | Minor | High | Functional Testing | Positive |

| TC\_05 | Verify that the system suggests popular hashtags as users type | This test case verifies that the system suggests popular hashtags as users type in the post editor by typing in a partial hashtag and checking if the system suggests relevant popular hashtags. | Hashtag Suggestion | High | User is logged in and has the ability to create posts. | #Example | 1. Navigate to the post creation page. 2. Enter text in the post editor. 3. Start typing #Example in the post editor. 4. Verify that the system suggests popular hashtags starting with #Example. | Popular hashtags are displayed. | The system suggests relevant popular hashtags starting with #Example as the user types. | Minor | High | Functional Testing | Positive |

---

## **Data - 9**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Emojis and Reactions

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

## Test Cases for Emojis and Reactions Feature

\*\*Project Name:\*\* Social Media Platform

**\*\*Feature Name:\*\* Emojis and Reactions**

**\*\*User Story Title:\*\*** As a user, I want to use emojis and reactions so that I can express my feelings about posts and messages.

**\*\*Acceptance Criteria:\*\***

1. Users can react to posts with a variety of emojis.
2. Emojis can be used in comments and messages.
3. Users can see a count of reactions on posts.
4. The platform should support a wide range of emojis.
5. Users can remove or change their reactions.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC\_Emojis\_01

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

**\*\*Test Case Description:\*\*** This test case verifies that users can react to posts with a variety of emojis. It checks if different emoji options are available and if users can select and apply them to posts.

**\*\*Test Suite:\*\*** Functionality Testing

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* A post exists on the platform.

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

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

1. Navigate to a post on the platform.
2. Locate the reaction options below the post.

3. Click on the reaction options and observe the available emojis.

4. Select a few different emojis and apply them to the post.

**\*\*Postconditions:\*\***

\* The selected emoji should be visible next to the post.

\* The emoji count should be updated after each reaction.

**\*\*Expected Result:\*\*** The user should be able to see and select a variety of emojis to react to the post.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Emojis\_02

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

**\*\*Test Case Description:\*\*** This test case verifies that users can use emojis in comments and messages. It checks if users can type or insert emojis within the comment and message fields.

**\*\*Test Suite:\*\*** Functionality Testing

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* A post exists on the platform.

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

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

1. Navigate to a post on the platform.

2. Click on the "Comment" or "Message" field.

3. Type or insert a few different emojis within the comment or message.

4. Submit the comment or message.

**\*\*Postconditions:\*\***

\* The emojis should be displayed correctly in the submitted comment or message.

**\*\*Expected Result:\*\*** The user should be able to use emojis in comments and messages.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Emojis\_03

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

**\*\*Test Case Description:\*\*** This test case verifies that users can see a count of reactions on posts. It checks if the platform displays the total number of reactions for each post, including different emojis.

**\*\*Test Suite:\*\*** Functionality Testing

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* A post exists with multiple reactions from different users.

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

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

1. Navigate to a post with multiple reactions.
2. Observe the reaction count displayed below the post.
3. Verify that the count includes all different types of emojis used.

**\*\*Postconditions:\*\***

\* The reaction count should be accurate and reflect the total number of reactions.

**\*\*Expected Result:\*\*** The user should be able to see a count of all reactions, including different emoji reactions, on each post.

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

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

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

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

**\*\*Test Case ID:\*\* TC\_Emojis\_04**

**\*\*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 wide range of emojis. It checks if the platform includes a comprehensive set of emojis covering various categories and expressions.**

**\*\*Test Suite:\*\* Functionality Testing**

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* A post exists on the platform.

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

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

1. Navigate to a post on the platform.

2. Click on the reaction options below the post.

3. Scroll through all available emojis and observe different categories (like smileys, animals, food, etc.).

4. Select emojis from different categories and apply them to the post.

**\*\*Postconditions:\*\***

\* The emojis should be displayed correctly and match the selected ones.

**\*\*Expected Result:\*\* The platform should support a wide range of emojis across various categories, ensuring diverse expression options for users.**

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

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

**\*\*Type of Testing:\*\*** Functionality Testing

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

**\*\*Test Case ID:\*\*** TC\_Emojis\_05

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

**\*\*Test Case Description:\*\*** This test case verifies that users can remove or change their reactions on posts. It checks if users can easily remove their existing reactions and apply a different one.

**\*\*Test Suite:\*\*** Functionality Testing

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* A post exists with the user's reaction.

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

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

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

2. Locate the user's reaction emoji.

3. Click on the user's reaction to remove it.

4. Select a different emoji and apply it to the post.

**\*\*Postconditions:\*\***

\* The user's previous reaction should be removed.

\* The newly selected emoji should be displayed.

\* The reaction count should be updated accordingly.

**\*\*Expected Result:\*\*** The user should be able to remove their previous reaction and change it to a different emoji.

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

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

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

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

-----

## **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

**\*\*Project:\*\*** Social Media Platform

**\*\*Feature:\*\*** Live Streaming

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

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-LS-01

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

**\*\*Test Case Description:\*\*** This test case verifies the functionality of initiating and terminating a live stream.

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* User has a working camera and microphone.

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

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

1. Navigate to the Live Stream creation page.
2. Click on the "Start Live Stream" button.
3. Verify that the live stream starts successfully.
4. Click on the "End Live Stream" button.
5. Verify that the live stream ends successfully.

**\*\*Postconditions:\*\*** The live stream should be terminated.

**\*\*Expected Result:\*\*** The user should be able to successfully start and stop a live stream.

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

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

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

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

**\*\*Test Case ID:\*\* TC-LS-02**

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

**\*\*Test Case Description:\*\* This test case verifies the ability to control the visibility of a live stream.**

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

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

**\*\*Preconditions:\*\***

\* User is logged in to the platform.

\* User has a working camera and microphone.

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

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

1. Navigate to the Live Stream creation page.
2. Select the "Public" visibility option for the live stream.
3. Verify that the live stream is visible to all users.
4. Start a new live stream.
5. Select the "Followers Only" visibility option.
6. Verify that the live stream is only visible to the user's followers.

**\*\*Postconditions:\*\* Live stream visibility should be updated as per the selected option.**

**\*\*Expected Result:\*\* The user should be able to successfully set the visibility of a live stream to either "Public" or "Followers Only."**

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

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

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

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

**\*\*Test Case ID:\*\* TC-LS-03**

**\*\*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 during a live stream.**

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

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

**\*\*Preconditions:\*\***

\* User A is logged in and broadcasting a live stream.

\* User B is logged in and viewing the live stream.

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

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

1. User B types a comment in the live stream comment section.
2. Verify that User A sees the comment appear in the live stream.
3. User B clicks on a reaction button (e.g., like, heart).
4. Verify that User A sees the reaction displayed on the live stream.

**\*\*Postconditions:\*\* The comment and reaction should be displayed on the live stream.**

**\*\*Expected Result:\*\* Viewers should be able to comment and react to the live stream, and the streamer should be able to see these interactions in real-time.**

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

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

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

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

**\*\*Test Case ID:\*\* TC-LS-04**

**\*\*Test Case Title:\*\* Verify that the stream should display viewer count and live comments.**

**\*\*Test Case Description:\*\* This test case verifies that the live stream interface displays relevant information for viewers.**

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

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

**\*\*Preconditions:\*\***

\* User A is logged in and broadcasting a live stream.

\* User B and User C are logged in and viewing the live stream.

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

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

1. Verify that the live stream interface displays the current number of viewers (should show 2).

2. User B types a comment in the comment section.

3. Verify that the comment appears in the live stream comment section.

**\*\*Postconditions:\*\*** The viewer count and comments should be displayed accurately.

**\*\*Expected Result:\*\*** The live stream interface should display the number of current viewers and the live comments in real-time.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-LS-05

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

**\*\*Test Case Description:\*\*** This test case verifies the functionality of saving a live stream for later viewing.

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

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

**\*\*Preconditions:\*\***

\* User is logged in and has completed a live stream.

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

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

1. Access the completed live stream from the user's profile or the live stream history.
2. Click on the "Save" or "Download" button (depending on the implementation).
3. Verify that the live stream is saved or downloaded successfully.

**\*\*Postconditions:\*\*** The live stream should be saved or downloaded as a file.

**\*\*Expected Result:\*\*** The user should be able to save or download the completed live stream for later viewing.

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

**\*\*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:

## Test Cases for Stories Feature

\*\*Project:\*\* Social Media Platform

\*\*Feature:\*\* Stories

\*\*User Story:\*\* As a user, I want to share stories so that I can share temporary moments with my followers.

\*\*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-ST-01	Verify that users can create stories with photos, videos, and text overlays.	This test case verifies that the user can successfully create a story using different media types, including photos, videos, and text overlays.	Stories	High	The user is logged in to the platform.	A photo, a video, and some text content.	1. Navigate to the "Stories" section of the platform.   2. Select the "Create Story" option.   3. Upload a photo from the user's device.   4. Upload a video from the user's device.   5. Add text overlays to the story.   6. Publish the story.	The story is successfully created and published, containing the uploaded photo, video, and text overlay.	Minor	High	Functional Testing	Positive	

| TC-ST-02 | Verify that stories disappear after 24 hours. | This test case verifies that stories are automatically removed from the user's profile after a 24-hour period. | Stories | High | The user is logged in to the platform. | A story created at least 24 hours prior to the test. | 1. Create a story with any content. <br> 2. Wait for 24 hours. <br> 3. Check the user's profile to see if the story is still visible. | The story should no longer be visible on the user's profile after 24 hours. | Major | High | Functional Testing | Positive |

| TC-ST-03 | Verify that users can see who viewed their stories. | This test case verifies that the user can view the list of users who have viewed their story. | Stories | High | The user is logged in to the platform. | A story with at least one viewer. | 1. Create a story and share it with at least one other user. <br> 2. Navigate to the story and check the "Viewers" section. | The "Viewers" section should display a list of users who have viewed the story. | Minor | High | Functional Testing | Positive |

| TC-ST-04 | Verify that stories can be shared publicly or with selected followers. | This test case verifies that users have the option to share their stories publicly or with a select group of followers. | Stories | High | The user is logged in to the platform. | A user with multiple followers. | 1. Create a story. <br> 2. Navigate to the "Share" options for the story. <br> 3. Select the "Public" option to share the story with all users. <br> 4. Create a new story. <br> 5. Select the "Selected Followers" option and choose a specific group of followers. | The story should be shared according to the selected sharing option: publicly or with selected followers. | Major | High | Functional Testing | Positive |

| TC-ST-05 | Verify that users can add stickers and filters to their stories. | This test case verifies that users can enhance their stories by adding stickers and filters. | Stories | High | The user is logged in to the platform. | A photo for the story. | 1. Create a story with a photo. <br> 2. Navigate to the "Stickers" section and add a sticker to the story. <br> 3. Navigate to the "Filters" section and apply a filter to the story. | The stickers and filter are successfully added to the story. | Minor | High | Functional Testing | Positive |

-----

**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

**\*\*Project Name:\*\* Social Media Platform**

**\*\*Feature Name:\*\* Analytics and Insights**

**\*\*User Story Title:\*\* As a user, I want to access analytics and insights so that I can understand my audience**



and optimize my content.

**\*\*Acceptance Criteria:\*\***

- 1. Users can view data on follower demographics.
- 2. Post reach and engagement metrics are available.
- 3. Users can track audience growth trends.
- 4. Analytics are displayed in a user-friendly format.
- 5. Data can be filtered by time range and content type.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

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

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

**\*\*Preconditions:\*\***

- User is logged in to the platform.
- User has at least one post with engagement.

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

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

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Priority:\*\***

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

**\*\*Test Case Approach:\*\***

**\*\*Test Cases:\*\***

**\*\*Test Case ID: TC\_01\*\***

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

**\*\*Test Case Description:\*\*** This test case verifies that the platform displays follower demographics data, including age, gender, location, and other relevant information.

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

1. Navigate to the "Analytics" section of the user profile.
2. Select "Follower Demographics" from the available options.
3. Verify that data on follower demographics is displayed, including age range, gender distribution, location distribution, etc.

**\*\*Expected Result:\*\*** The platform should display detailed follower demographics data in a clear and understandable format.

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

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

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

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

**\*\*Test Case ID: TC\_02\*\***

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

**\*\*Test Case Description:\*\*** This test case checks that the platform displays metrics related to post reach and engagement, including impressions, reach, likes, comments, shares, and other relevant metrics.

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

1. Navigate to the "Analytics" section of the user profile.
2. Select "Post Performance" from the available options.

3. Choose a specific post from the list.

4. Verify that the post details include metrics such as impressions, reach, likes, comments, shares, and other engagement data.

**\*\*Expected Result:\*\*** The platform should display detailed post reach and engagement metrics for each selected post.

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

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

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

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

**\*\*Test Case ID: TC\_03\*\***

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

**\*\*Test Case Description:\*\*** This test case confirms that the platform provides data on follower growth over time, enabling users to analyze trends and identify potential growth opportunities.

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

1. Navigate to the "Analytics" section of the user profile.

2. Select "Audience Growth" from the available options.

3. Observe the displayed graph or chart representing follower growth over a specific period.

4. Verify that the chart accurately shows the trend of follower growth over time.

**\*\*Expected Result:\*\*** The platform should display a visual representation of follower growth trends, allowing users to track their audience growth over time.

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

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

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

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

**\*\*Test Case ID: TC\_04\*\***

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

**\*\*Test Case Description:\*\*** This test case ensures that analytics are presented in an intuitive and easy-to-understand manner, using clear labels, charts, graphs, and other visual aids.

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

1. Navigate to the "Analytics" section of the user profile.
2. Explore different sections of the analytics dashboard, such as follower demographics, post performance, and audience growth.
3. Verify that data is presented in a clear, concise, and visually appealing format using charts, graphs, tables, and relevant labels.

**\*\*Expected Result:\*\*** Analytics data should be presented in a user-friendly format with clear labels, charts, and graphs that are easy to understand and interpret.

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

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

**\*\*Type of Testing:\*\*** Usability Testing

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

**\*\*Test Case ID: TC\_05\*\***

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

**\*\*Test Case Description:\*\*** This test case checks that users can filter analytics data by specific time periods and content types to gain more focused insights.

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

1. Navigate to the "Analytics" section of the user profile.
2. Choose any section of the analytics dashboard, such as post performance or audience growth.
3. Select a specific time range filter (e.g., last week, last month, custom range).
4. Choose a content type filter (e.g., posts, photos, videos) if applicable.
5. Verify that the data displayed is filtered according to the selected time range and content type.

**\*\*Expected Result:\*\*** Analytics data should be successfully filtered by the selected time range and content

type, allowing users to customize their analysis and gain specific insights.

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

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

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

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

-----

## **Data - 13**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Events and Groups

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

and see the list of attendees.']

**Generated Test Cases:**

## Test Cases for Social Media Platform - Events and Groups

**Project Name:** Social Media Platform

**Feature Name:** Events and Groups

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

**Acceptance Criteria:**

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

**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-EG-01 | Verify that users can create events | This test case verifies that a user can create an event on the platform by filling in the required details such as name, date, time, location, and description. | Events and Groups | High | User is logged in to the platform | Event name, date, time, location, description | 1. Navigate to the Events section. 2. Click on "Create Event". 3. Fill in the required details for the event. 4. Click on "Create Event". | User is redirected to the created event page | The event should be successfully created and

displayed on the platform. | Major | High | Functional Testing | Positive |

| TC-EG-02 | Verify that users can join events | This test case verifies that a user can successfully join an existing event on the platform. | Events and Groups | High | User is logged in to the platform, an event is already created | Event ID | 1. Navigate to the Events section. 2. Select an existing event. 3. Click on "Join Event". | User is added to the event's list of attendees | The user should be added to the event's attendee list and receive a notification about joining the event. | Major | High | Functional Testing | Positive |

| TC-EG-03 | Verify that users can join interest-based groups | This test case verifies that a user can successfully join an existing interest-based group on the platform. | Events and Groups | High | User is logged in to the platform, a group is already created | Group ID | 1. Navigate to the Groups section. 2. Select an existing group based on their interests. 3. Click on "Join Group". | User is added to the group's member list | The user should be added to the group's member list and receive a notification about joining the group. | Major | High | Functional Testing | Positive |

| TC-EG-04 | Verify that group discussions can be organized by topic | This test case verifies that group discussions are organized by different topics, allowing users to filter and participate in relevant conversations. | Events and Groups | Medium | User is logged in to the platform, a group is already created | Group ID, Discussion topic | 1. Navigate to a group's discussion section. 2. Verify that discussions are categorized by different topics. 3. Search for a specific topic within the discussions. | Users can easily find and participate in discussions related to their specific interests. | Minor | Medium | Functional Testing | Positive |

| TC-EG-05 | Verify that users can RSVP to events and see the list of attendees | This test case verifies that users can RSVP to events and view the list of attendees who have confirmed their participation. | Events and Groups | High | User is logged in to the platform, an event is already created | Event ID | 1. Navigate to an event page. 2. Click on "RSVP". 3. Select "Attending", "Maybe", or "Not Attending". 4. View the list of attendees for the event. | User's RSVP status is updated, and they can view the list of attendees | The user's RSVP status should be updated accordingly, and they should be able to view the list of attendees. | Major | High | Functional Testing | 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

**\*\*Project Name:\*\* Social Media Platform**



**\*\*User Story Title:\*\*** As a content creator, I want to access monetization options so that I can earn revenue from my content.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-MON-01

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

**\*\*Test Case Description:\*\*** This test case verifies that users can initiate the process of applying for monetization options on the platform.

**\*\*Test Suite:\*\*** Monetization

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

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* The user has uploaded content on the platform.

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

\* No test data needed.

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

1. Navigate to the user profile page.
2. Locate the "Monetization" or "Earnings" section.
3. Click on the "Apply for Monetization" or "Enable Monetization" button.
4. Review the eligibility requirements and terms and conditions.
5. Submit the application form.

**\*\*Postconditions:\*\***

\* The application is submitted successfully.

\* The user receives a notification or confirmation about the application submission.

**\*\*Expected Result:\*\*** The user successfully submits the monetization application.

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

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

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

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

**\*\*Test Case ID:\*\* TC-MON-02**

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

**\*\*Test Case Description:\*\* This test case confirms that the platform offers advertising revenue and sponsored content as monetization options.**

**\*\*Test Suite:\*\* Monetization**

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

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* The user has activated monetization options.

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

\* No test data needed.

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

1. Navigate to the user profile page.

2. Access the "Monetization" or "Earnings" section.

3. Check the available monetization options.

4. Verify the presence of "Advertising Revenue" and "Sponsored Content" options.

**\*\*Postconditions:\*\***

\* The user can view available monetization options.

**\*\*Expected Result:\*\* Both "Advertising Revenue" and "Sponsored Content" options are present and accessible to the user.**

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

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

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

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

**\*\*Test Case ID:\*\* TC-MON-03**

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

**\*\*Test Case Description:\*\* This test case ensures that the platform provides a mechanism for users to receive payments for their monetized content.**

**\*\*Test Suite:\*\* Monetization**

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

**\*\*Preconditions:\*\***

- \* A user account is created and logged in.
- \* The user has activated monetization options.
- \* The user has earned revenue through monetization.

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

- \* User account with sufficient earned revenue.

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

1. Navigate to the user profile page.
2. Access the "Monetization" or "Earnings" section.
3. Verify the availability of a payment option (e.g., bank transfer, PayPal).
4. Enter the required payment information (e.g., bank account details).
5. Request payment withdrawal.

**\*\*Postconditions:\*\***

- \* The payment request is submitted successfully.
- \* The user receives a confirmation about the payment request.

**\*\*Expected Result:\*\* The user successfully initiates a payment withdrawal request.**

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

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

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

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

**\*\*Test Case ID:\*\* TC-MON-04**

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

**\*\*Test Case Description:\*\* This test case confirms that the platform provides detailed reports on user earnings from monetization options.**

**\*\*Test Suite:\*\* Monetization**

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

**\*\*Preconditions:\*\***

- \* A user account is created and logged in.
- \* The user has activated monetization options.
- \* The user has earned revenue through monetization.

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

- \* No test data needed.

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

1. Navigate to the user profile page.
2. Access the "Monetization" or "Earnings" section.
3. Locate the "Earnings Report" or "Transaction History" section.
4. Verify the availability of detailed earnings data including date, source of earnings, and amount earned.

**\*\*Postconditions:\*\***

- \* The user can view detailed earnings reports.

**\*\*Expected Result:\*\* The user can view detailed reports of their earnings from different monetization sources.**

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

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

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

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

**\*\*Test Case ID:\*\* TC-MON-05**

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

**\*\*Test Case Description:\*\* This test case ensures that the platform clearly displays the eligibility requirements for monetizing content.**

**\*\*Test Suite:\*\* Monetization**

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

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* The user has not yet activated monetization options.

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

\* No test data needed.

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

1. Navigate to the user profile page.
2. Access the "Monetization" or "Earnings" section.
3. Verify the presence of a "Eligibility Requirements" or "Terms & Conditions" section.
4. Review the eligibility criteria listed.
5. Check if the requirements are clearly stated and easily understandable.

**\*\*Postconditions:\*\***

\* The user has reviewed the eligibility requirements.

**\*\*Expected Result:\*\* The platform clearly displays the eligibility requirements for monetizing content, including minimum follower count, content quality standards, and other criteria.**

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

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

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

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

---

## **Data - 15**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Explore or Discover Tab

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

## Test Cases for Explore/Discover Tab

**\*\*Project:\*\* Social Media Platform**

**\*\*Feature:\*\* Explore/Discover Tab**

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

**\*\*Acceptance Criteria:\*\***

1. The explore tab shows trending and popular content.
2. Content recommendations are personalized based on user interests.
3. Users can filter content by category or topic.
4. The tab updates regularly with new content.
5. Users can save or follow content from the explore tab.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC-EXP-01**

**\*\*Test Case Title:\*\*** Verify that the Explore tab displays trending and popular content.

**\*\*Test Case Description:\*\*** This test case verifies that the Explore tab displays a feed of trending and popular content from across the platform, showcasing posts, accounts, and topics that are currently gaining significant attention.

**\*\*Test Suite:\*\*** Explore Tab Functionality

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

**\*\*Preconditions:\*\***

\* User is logged into the platform.

\* Explore/Discover tab is accessible.

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

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

1. Navigate to the Explore/Discover tab.
2. Observe the displayed content, including posts, accounts, and topics.
3. Verify that the content displayed is labeled as trending or popular.
4. Check if the content displayed is diverse and reflects various categories and topics.

**\*\*Postconditions:\*\*** User remains on the Explore tab.

**\*\*Expected Result:\*\*** The Explore tab displays a diverse selection of trending and popular content, with clear labels indicating their relevance.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-EXP-02

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

**\*\*Test Case Description:\*\*** This test case validates whether content recommendations displayed in the Explore tab are customized based on the user's past interactions, including posts they've liked, accounts they follow, and topics they engage with.

**\*\*Test Suite:\*\*** Explore Tab Functionality

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

**\*\*Preconditions:\*\***

\* User is logged into the platform with browsing history and engagement data.

\* Explore/Discover tab is accessible.

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

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

1. Navigate to the Explore/Discover tab.



2. Observe the displayed content, including posts, accounts, and topics.
3. Identify the content related to user's interests based on their past interactions.
4. Check if the content aligns with the user's past likes, comments, and follows.

**\*\*Postconditions:\*\*** User remains on the Explore tab.

**\*\*Expected Result:\*\*** The Explore tab displays content recommendations that align with the user's past engagement history, showcasing personalized suggestions based on their interests.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-EXP-03

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

**\*\*Test Case Description:\*\*** This test case confirms that users can filter the content displayed in the Explore tab by selecting specific categories or topics of interest, enabling them to narrow down the displayed content to relevant areas.

**\*\*Test Suite:\*\*** Explore Tab Functionality

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

**\*\*Preconditions:\*\***

\* User is logged into the platform.

\* Explore/Discover tab is accessible.

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

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

1. Navigate to the Explore/Discover tab.
2. Locate the filtering options, typically presented as a dropdown menu or clickable tags.
3. Select a specific category or topic from the filtering options.
4. Verify that the content displayed updates to reflect the selected filter.

**\*\*Postconditions:\*\*** User remains on the Explore tab, viewing filtered content.

**\*\*Expected Result:\*\*** The Explore tab content dynamically adjusts based on the applied filter, displaying only content related to the selected category or topic.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-EXP-04

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

**\*\*Test Case Description:\*\*** This test case verifies that the Explore tab refreshes its displayed content periodically, ensuring users see fresh and up-to-date trending and popular content without manually refreshing the page.

**\*\*Test Suite:\*\*** Explore Tab Functionality

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

**\*\*Preconditions:\*\***

\* User is logged into the platform.

\* Explore/Discover tab is accessible.

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

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

1. Navigate to the Explore/Discover tab.
2. Observe the initial displayed content.
3. Wait for a predetermined period (e.g., 5 minutes).
4. Verify that new content has appeared in the feed, reflecting recent updates and changes in trending topics.

**\*\*Postconditions:\*\*** User remains on the Explore tab.

**\*\*Expected Result:\*\*** The Explore tab displays refreshed content, including new posts, accounts, and topics that have gained popularity since the last view.

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

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

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

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

**\*\*Test Case ID:\*\* TC-EXP-05**

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

**\*\*Test Case Description:\*\* This test case confirms that users can bookmark or follow specific content they encounter in the Explore tab, allowing them to revisit it later or stay updated on the chosen accounts or topics.**

**\*\*Test Suite:\*\* Explore Tab Functionality**

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

**\*\*Preconditions:\*\***

\* User is logged into the platform.

\* Explore/Discover tab is accessible.

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

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

1. Navigate to the Explore/Discover tab.

2. Identify a post, account, or topic of interest.

3. Locate the "Save" or "Follow" button/icon associated with the chosen content.

4. Click on the "Save" or "Follow" button/icon.

5. Verify that the content is saved or followed successfully, typically indicated by a visual change on the button/icon or a confirmation message.

**\*\*Postconditions:\*\* User remains on the Explore tab.**

**\*\*Expected Result:\*\* The selected content is saved or followed successfully, enabling users to access or stay updated on it later.**

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

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

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

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

-----

**Data - 16**

**Project name -**

Social Media Platform

**Project description -**

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

**Feature name -**

Accessibility Features

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

## ## Test Cases for Accessibility Features

**\*\*Project:\*\*** Social Media Platform

**\*\*User Story:\*\*** As a user with disabilities, I want accessible features so that I can use the platform comfortably.

**\*\*Acceptance Criteria:\*\***

1. The platform provides alt text for images.
2. Screen reader compatibility is available.
3. Keyboard shortcuts can be used for navigation.
4. Color contrast and font size options are provided.
5. Accessibility features are clearly documented.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\*** (TC-Accessibility-XX)

**\*\*Test Case Title:\*\*** (Verify that...)

**\*\*Test Case Description:\*\*** (Brief description of the test case)

**\*\*Test Suite:\*\*** (Accessibility Features)

**\*\*Test Priority:\*\*** (High)

**\*\*Preconditions:\*\***

- \* User account is created and logged in.
- \* The platform is accessible in a web browser.

**\*\*Test Data:\*\*** (Specify any data required for the test case)

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

- \* (List the steps for executing the test case)

**\*\*Postconditions:\*\***

- \* (List any follow-up actions after test execution)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Major)

**\*\*Priority:\*\*** (High)

**\*\*Type of Testing:\*\*** (Accessibility Testing)

**\*\*Test Case Approach:\*\*** (Positive, Negative)

**\*\*Test Cases:\*\***

**\*\*TC-Accessibility-01\*\***

**\*\*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 alternative text descriptions that are accurate and meaningful.

**\*\*Test Data:\*\*** A variety of images from different sections of the platform (e.g., profile pictures, post images, videos).

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

1. Navigate to different sections of the platform containing images.
2. Select an image and use a screen reader to read the alt text provided.
3. Repeat steps 1 & 2 for a range of images.

**\*\*Expected Result:\*\*** The screen reader should read a concise and accurate description of the image, providing context and meaning to visually impaired users.

**\*\*TC-Accessibility-02\*\***

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

**\*\*Test Case Description:\*\*** This test case ensures that a popular screen reader (e.g., JAWS, NVDA) can

navigate the platform's main features and content, including navigation menus, user profiles, posts, and comments.

**\*\*Test Data:\*\*** None.

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

1. Use a screen reader to access the platform.
2. Navigate through the main sections of the platform (e.g., home page, profile page, posts, comments).
3. Use the screen reader to read page titles, headers, and content.
4. Verify that the screen reader can accurately read and announce elements on the page.

**\*\*Expected Result:\*\*** The screen reader should be able to navigate the platform seamlessly and read all elements correctly, providing a clear and accessible user experience.

**\*\*TC-Accessibility-03\*\***

**\*\*Test Case Title:\*\*** Verify that keyboard shortcuts are available for navigation and common actions.

**\*\*Test Case Description:\*\*** This test case ensures that keyboard shortcuts are available for common actions like navigating menus, opening and closing elements, and interacting with content.

**\*\*Test Data:\*\*** None.

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

1. Navigate to different sections of the platform (e.g., home page, profile page, posts, comments).



2. Use keyboard shortcuts to navigate menus, access different sections, and interact with content (e.g., like posts, comment, share).

**\*\*Expected Result:\*\*** The keyboard shortcuts should function correctly and allow users to navigate the platform and interact with its features without needing a mouse.

**\*\*TC-Accessibility-04\*\***

**\*\*Test Case Title:\*\*** Verify that color contrast and font size options are available for users to customize.

**\*\*Test Case Description:\*\*** This test case ensures that users can adjust the platform's color contrast and font size settings to improve readability and accessibility.

**\*\*Test Data:\*\*** None.

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

1. Access the platform's settings or accessibility options.
2. Adjust the color contrast settings (e.g., high contrast mode).
3. Observe the visual changes to the platform's interface.
4. Adjust the font size settings (e.g., increase or decrease font size).
5. Observe the changes in font size and readability.

**\*\*Expected Result:\*\*** The platform should provide options to adjust color contrast and font size settings, allowing users to personalize the interface for improved accessibility.

**\*\*TC-Accessibility-05\*\***

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

**\*\*Test Case Description:\*\*** This test case checks whether the platform provides clear documentation on its accessibility features, including information on keyboard shortcuts, color contrast options, and screen reader compatibility.

**\*\*Test Data:\*\*** None.

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

1. Locate the platform's help center or documentation.
2. Search for information on accessibility features (e.g., "accessibility", "keyboard shortcuts", "screen reader compatibility").
3. Verify that the documentation provides clear and comprehensive information about the platform's accessibility features.

**\*\*Expected Result:\*\*** The platform should offer clear and accessible documentation that outlines all accessibility features, providing users with the information they need to use the platform effectively.

-----

**Data - 17**

**Project name -**

Social Media Platform

**Project description -**

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

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

#### **Feature name -**

Safety and Moderation Tools

#### **Feature description -**

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

#### **Userstory title -**

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

#### **Acceptance criteria -**

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

#### **Generated Test Cases:**

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

**\*\*Project:\*\*** 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 Template:\*\***

**\*\*Test Case ID:\*\*** (Unique identifier)

**\*\*Test Case Title:\*\*** (Starts with "Verify that...")

**\*\*Test Case Description:\*\*** (Brief description of the test case)

**\*\*Test Suite:\*\*** Safety & Moderation Tools

**\*\*Test Priority:\*\*** (High, Medium, Low)

**\*\*Preconditions:\*\*** (Prerequisites for test execution)

**\*\*Test Data:\*\*** (Required data, if any)

**\*\*Test Steps:\*\*** (Steps for executing the test case)

**\*\*Postconditions:\*\*** (Follow-up actions after test execution)

**\*\*Expected Result:\*\*** (Expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Priority:\*\*** (High, Medium, Low)

**\*\*Type of Testing:\*\*** (Functional Testing)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

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

**\*\*Test Case ID:\*\*** TC-SM-01

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

**\*\*Test Case Description:\*\*** This test case verifies that the platform's content moderation algorithms effectively identify and filter inappropriate content, such as hate speech, violence, and nudity.

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

**\*\*Preconditions:\*\***

- \* A user account is created.
- \* The user is logged in.
- \* The user has permission to post content.

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

\* A sample post containing inappropriate content.

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

1. Log in as the user.
2. Create a post containing inappropriate content (e.g., hate speech, violence).
3. Submit the post.
4. Observe the content moderation algorithm's response.

**\*\*Postconditions:\*\***

- \* The inappropriate content should be flagged and removed from the platform.
- \* The user should receive a notification about the flagged content.

**\*\*Expected Result:\*\*** The content moderation algorithm should identify and filter the inappropriate content, preventing it from being displayed on the platform.

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

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

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

**\*\*Test Case Approach:\*\*** Negative

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

**\*\*Test Case ID:\*\*** TC-SM-02

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

**\*\*Test Case Description:\*\*** This test case ensures that users have a mechanism to report harmful or inappropriate content, enabling the platform to take appropriate action.

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

**\*\*Preconditions:\*\***

- \* A user account is created.
- \* The user is logged in.
- \* A post containing harmful or inappropriate content is available on the platform.

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

\* No test data needed.

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

1. Log in as the user.
2. Locate the harmful or inappropriate content.
3. Click on the "Report" or "Flag" button.
4. Select the appropriate reason for reporting (e.g., hate speech, harassment).
5. Submit the report.

**\*\*Postconditions:\*\***

- \* The user should receive a confirmation message that the report has been submitted.
- \* The platform should review the reported content and take appropriate action.

**\*\*Expected Result:\*\*** The user should be able to successfully report the content, and the platform should receive and process the report.

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

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

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

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

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

**\*\*Test Case ID:\*\*** TC-SM-03

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

**\*\*Test Case Description:\*\*** This test case ensures that users have the ability to block other users from interacting with them, promoting a safer online experience.

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

**\*\*Preconditions:\*\***

- \* Two user accounts are created.

\* Both users are logged in.

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

\* No test data needed.

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

1. Log in as the user who wants to block another user.
2. Visit the profile of the user to be blocked.
3. Locate the "Block" or "Report" button.
4. Click on the "Block" button.
5. Confirm the block action.

**\*\*Postconditions:\*\***

- \* The blocking user should no longer see content from the blocked user.
- \* The blocked user should not be able to interact with the blocking user.

**\*\*Expected Result:\*\*** The blocking functionality should work as intended, preventing further interaction between the blocked and blocking user.

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

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

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

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

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

**\*\*Test Case ID:\*\*** TC-SM-04

**\*\*Test Case Title:\*\*** Verify that comment filters prevent offensive language.

**\*\*Test Case Description:\*\*** This test case verifies that the platform's comment filter effectively identifies and prevents offensive language from being displayed in comments, maintaining a respectful and appropriate online environment.

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

**\*\*Preconditions:\*\***

- \* A user account is created.
- \* The user is logged in.
- \* The user has permission to comment on posts.

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

- \* A sample comment containing offensive language.

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

1. Log in as the user.
2. Locate a post and attempt to leave a comment containing offensive language.
3. Submit the comment.
4. Observe the comment filter's response.

**\*\*Postconditions:\*\***

- \* The offensive language should be detected and filtered from the comment.
- \* The user should receive an error message or notification about the filtered content.

**\*\*Expected Result:\*\*** The comment filter should prevent offensive language from being displayed in comments, promoting a positive and respectful online environment.

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

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

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

**\*\*Test Case Approach:\*\*** Negative

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

**\*\*Test Case ID:\*\*** TC-SM-05

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

**\*\*Test Case Description:\*\*** This test case ensures that the platform's community guidelines are consistently applied across all user content, ensuring a fair and consistent user experience.



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

**\*\*Preconditions:\*\***

- \* A user account is created.
- \* The user is logged in.
- \* The user has permission to post content.
- \* The community guidelines are clearly defined and accessible on the platform.

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

- \* Multiple user posts that violate different aspects of the community guidelines.

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

1. Log in as the user.
2. Create multiple posts that violate different community guidelines (e.g., spam, impersonation, harassment).
3. Submit the posts.
4. Observe the platform's response to the violations.

**\*\*Postconditions:\*\***

- \* The violating content should be flagged, removed, or moderated accordingly.
- \* The user should receive appropriate consequences (e.g., warnings, account suspension).

**\*\*Expected Result:\*\*** The platform should consistently enforce community guidelines across all user content, ensuring a safe and respectful online environment for all users.

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

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

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

**\*\*Test Case Approach:\*\* Negative**

These test cases provide a comprehensive approach to testing the safety and moderation features of the social media platform, ensuring that each acceptance criterion is covered and potential defects are identified early in the development cycle.

---

## **Data - 18**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Multi-Language Support

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

## Test Cases for Multi-Language Support Feature

**\*\*Project Name:\*\* Social Media Platform**

**\*\*Feature Name:\*\* Multi-Language Support**

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

**\*\*Acceptance Criteria:\*\***

- 1. Users can select their preferred interface language.
- 2. The platform offers content translation features.
- 3. Posts and messages can be translated into different languages.
- 4. Language preferences are saved and applied consistently.
- 5. The platform supports a wide range of languages.

**\*\*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_MLS_01	Verify that users can select their preferred interface language	This test case checks if the user interface can be switched to different languages as per user preference.	Multi-Language Support	High	User is logged into the platform.	Interface language options: English, Spanish, French, German, Japanese	1. Navigate to the user profile settings. 2. Select the desired interface language from the dropdown menu. 3. Save the language preference.	User is logged into the platform with the selected interface language.	The platform interface should be displayed in the chosen language.	Major	High	Functional Testing	Positive
TC_MLS_02	Verify that the platform offers content translation features	This test case verifies the availability of content translation functionality for posts and messages.	Multi-Language Support	High	User is logged into the platform.	A post written in English.	1. Create a new post in English. 2. View the post and						

check for the "translate" option. 3. Select the desired target language (e.g., Spanish) | Post content is translated into the selected target language. | Major | High | Functional Testing | Positive |

| TC\_MLS\_03 | Verify that posts and messages can be translated into different languages | This test case checks if the content translation feature works correctly for both posts and messages. | Multi-Language Support | High | User is logged into the platform. | A post written in English and a message sent in French. | 1. Create a new post in English and send a message in French. 2. Translate the post into Spanish and the message into German. | The post and message are translated into the selected languages. | Major | High | Functional Testing | Positive |

| TC\_MLS\_04 | Verify that language preferences are saved and applied consistently | This test case verifies that the user's preferred language is saved and used across the platform. | Multi-Language Support | High | User is logged into the platform with a preferred language selected. | No test data needed | 1. Log out of the platform and log back in. 2. Navigate to different sections of the platform (newsfeed, messages, settings). | The user interface and displayed content should be consistent with the saved language preference. | Major | High | Functional Testing | Positive |

| TC\_MLS\_05 | Verify that the platform supports a wide range of languages | This test case checks if the platform supports a variety of popular languages, including major and less common ones. | Multi-Language Support | Medium | User is logged into the platform. | Interface language options: English, Spanish, French, German, Japanese, Chinese, Arabic, Russian, Portuguese, Korean, Hindi | 1. Access the language selection options. 2. Select various languages from the list and verify if they are supported. | The platform interface should display correctly in all the selected languages. | Minor | Medium | Functional Testing | Positive |

-----

**Data - 19**

**Project name -**

Social Media Platform

**Project description -**

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

**Feature name -**

Cross-Platform Integration

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

## Test Cases for Cross-Platform Integration

**\*\*Project Name:\*\* Social Media Platform**

**\*\*Feature Name:\*\* Cross-Platform Integration**

**\*\*User Story Title:\*\* As a user, I want cross-platform integration so that I can connect and share content across multiple platforms.**

**\*\*Acceptance Criteria:\*\***

1. Users can link their accounts with other social media platforms.
2. Content can be cross-posted to other platforms.
3. Users can sign in using single sign-on (SSO).
4. Integration with external media sources is available.
5. Content sharing across platforms is seamless.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-CP-01

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

**\*\*Test Case Description:\*\*** This test case verifies the functionality of linking accounts with other social media platforms. Users should be able to successfully link their accounts with supported platforms and have the option to disconnect them if needed.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

**\*\*Preconditions:\*\***

- \* A user account exists on the social media platform.
- \* The user has an account on a supported external platform.

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

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

1. Log in to the social media platform with the user account.
2. Navigate to the settings or profile section.
3. Locate the option to link accounts with other platforms.
4. Select the desired external platform.
5. Authorize the platform to connect with the social media account.
6. Verify that the account is successfully linked.

7. Attempt to disconnect the linked account.
8. Verify that the account is disconnected successfully.

**\*\*Postconditions:\*\***

- \* The user account remains active on the social media platform.
- \* The linked account status is updated accordingly.

**\*\*Expected Result:\*\*** The user account should be successfully linked and disconnected with the external platform.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-CP-02

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

**\*\*Test Case Description:\*\*** This test case verifies the functionality of cross-posting content to other platforms. Users should be able to choose which platforms to share content on, and the content should be posted correctly with appropriate formatting.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

**\*\*Preconditions:\*\***

- \* A user account exists on the social media platform.
- \* A user account is linked to an external platform.
- \* Content is available for cross-posting.

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

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

1. Log in to the social media platform with the user account.
2. Create a new post with text, images, or video content.

3. Select the desired external platforms for cross-posting.
4. Publish the post on the social media platform.
5. Verify that the content is successfully cross-posted to the selected platforms.
6. Verify that the content formatting is preserved on each platform.

**\*\*Postconditions:\*\***

- \* The content remains on the social media platform.
- \* The content is successfully shared on the external platforms.

**\*\*Expected Result:\*\*** The content should be cross-posted to the selected platforms with correct formatting.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-CP-03

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

**\*\*Test Case Description:\*\*** This test case verifies the functionality of using SSO to sign in. Users should be able to seamlessly sign in to the social media platform using their credentials from a linked platform.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

**\*\*Preconditions:\*\***

- \* A user account exists on the social media platform.
- \* The user account is linked to a supported platform with SSO capabilities.

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

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

1. Navigate to the social media platform's login page.
2. Select the SSO option for the linked platform.
3. Log in to the linked platform using the user's credentials.



4. Verify that the user is automatically redirected to the social media platform's homepage.

5. Verify that the user is successfully logged in to the social media platform.

**\*\*Postconditions:\*\***

\* The user is successfully logged in to the social media platform.

\* The user remains logged in on the external platform.

**\*\*Expected Result:\*\*** The user should be able to log in to the social media platform using SSO with the linked platform.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-CP-04

**\*\*Test Case Title:\*\*** Verify that integration with external media sources is available.

**\*\*Test Case Description:\*\*** This test case verifies that users can embed content from external media sources within the social media platform. The integration should allow users to embed content from popular sources like YouTube, Vimeo, or SoundCloud.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

**\*\*Preconditions:\*\***

\* A user account exists on the social media platform.

\* Content from supported external media sources is available for embedding.

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

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

1. Log in to the social media platform with the user account.

2. Create a new post or comment.

3. Paste the URL of content from a supported external media source.

4. Verify that the content is correctly embedded with playback functionality.
5. Attempt to embed content from an unsupported external media source.
6. Verify that the embed functionality is not available.

**\*\*Postconditions:\*\***

\* The content remains on the social media platform.

**\*\*Expected Result:\*\*** Supported external media content should be embedded correctly with playback functionality. Unsupported content should not be embeddable.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-CP-05

**\*\*Test Case Title:\*\*** Verify that content sharing across platforms is seamless.

**\*\*Test Case Description:\*\*** This test case verifies the seamless experience of content sharing across platforms. Users should be able to easily share content from the social media platform to linked external platforms with no unnecessary steps or errors.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

**\*\*Preconditions:\*\***

\* A user account exists on the social media platform.

\* The user account is linked to an external platform.

\* Content is available for sharing on the social media platform.

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

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

1. Log in to the social media platform with the user account.
2. Select a post or comment on the platform.

3. Identify the option to share the content on external platforms.
4. Select the desired external platform.
5. Verify that the content is shared to the platform without any errors or prompts.
6. Verify that the content is successfully shared on the external platform.

**\*\*Postconditions:\*\***

- \* The content remains on the social media platform.
- \* The content is successfully shared on the external platform.

**\*\*Expected Result:\*\*** The content should be seamlessly shared on the selected external platform without any errors.

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

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

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

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

-----

**Data - 20**

**Project name -**

Social Media Platform

**Project description -**

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

**Feature name -**

Gamification and Rewards

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

## Test Cases for Gamification and Rewards Feature

**\*\*Project Name:\*\* Social Media Platform**

**\*\*Feature Name:\*\* Gamification and Rewards**

**\*\*User Story Title:\*\* As a user, I want gamification and rewards so that I can earn recognition and incentives for my activities.**

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC\_Gamification\_01**

**\*\*Test Case Title:\*\* Verify that users can earn points and badges for their activities.**

**\*\*Test Case Description:\*\* This test case verifies that users earn points and badges for various activities on the platform, such as posting content, liking posts, commenting on posts, following other users, and sharing content.**

**\*\*Test Suite:\*\*** Gamification and Rewards

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

**\*\*Preconditions:\*\***

\* User account is created and logged in.

\* Activities are available for users to perform (posting, liking, commenting, following, sharing).

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

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

1. Log in as a registered user.
2. Perform various activities on the platform (post, like, comment, follow, share).
3. Observe the points and badge count in the user profile.
4. Verify that points and badges are awarded for each activity.
5. Check if the points and badge count updates accordingly with each activity performed.

**\*\*Postconditions:\*\***

\* Points and badges awarded for the performed activities should reflect in the user profile.

**\*\*Expected Result:\*\*** The user should earn points and badges for every activity performed on the platform, and the total count should reflect in the user's profile.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Gamification\_02

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

**\*\*Test Case Description:\*\*** This test case checks if the leaderboards accurately display the top users based on their earned points or achievements.

**\*\*Test Suite:\*\*** Gamification and Rewards

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

**\*\*Preconditions:\*\***

\* Users have earned points and badges through activities.

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

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

1. Access the leaderboard section of the platform.
2. Verify that the leaderboard displays users ranked based on their points or achievements.
3. Check if the ranking order is correct based on the earned points or badges.
4. Verify that the displayed users have corresponding points or achievements in their profiles.

**\*\*Postconditions:\*\***

\* Leaderboards should accurately reflect the top users based on their points or achievements.

**\*\*Expected Result:\*\*** The leaderboards should display the top users correctly based on their earned points or achievements, with the highest points or achievements ranked at the top.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Gamification\_03

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

**\*\*Test Case Description:\*\*** This test case checks if various challenges are available for users to complete and earn corresponding rewards.

**\*\*Test Suite:\*\*** Gamification and Rewards

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

**\*\*Preconditions:\*\***

\* User account is created and logged in.

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

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

1. Access the challenges section of the platform.
2. Verify that multiple challenges with different requirements and rewards are available.
3. Check if the challenge descriptions and reward details are clearly defined.
4. Attempt to complete a challenge by fulfilling the required conditions.
5. Verify that the reward is granted upon successful completion of the challenge.

**\*\*Postconditions:\*\***

\* User should receive the reward after completing the challenge successfully.

**\*\*Expected Result:\*\*** Challenges should be accessible to users, with clear descriptions, reward details, and rewards granted upon successful completion.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Gamification\_04

**\*\*Test Case Title:\*\*** Verify that rewards can include virtual goods, discounts, or exclusive content.

**\*\*Test Case Description:\*\*** This test case verifies that different types of rewards, such as virtual goods, discounts, or exclusive content, are available and can be claimed by users.

**\*\*Test Suite:\*\*** Gamification and Rewards

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

**\*\*Preconditions:\*\***

\* Users have earned rewards by completing challenges or achieving milestones.

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

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

1. Access the rewards section of the platform.
2. Verify that various rewards, including virtual goods, discounts, or exclusive content, are available.
3. Check if the reward descriptions and claim conditions are clear.

4. Attempt to claim a reward by fulfilling the claim conditions.

5. Verify that the reward is successfully claimed and credited to the user's account.

**\*\*Postconditions:\*\***

\* The claimed reward should be accessible to the user in their account.

**\*\*Expected Result:\*\*** Users should be able to claim different types of rewards based on their achievements, and the claimed rewards should be accessible in their accounts.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Gamification\_05

**\*\*Test Case Title:\*\*** Verify that users can view their progress and achievements in a dedicated section.

**\*\*Test Case Description:\*\*** This test case checks if users can access and view their progress and achievements in a dedicated section within their user profiles.

**\*\*Test Suite:\*\*** Gamification and Rewards

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

**\*\*Preconditions:\*\***

\* Users have earned points, badges, and rewards.

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

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

1. Access the user's profile section.

2. Navigate to the dedicated section for progress and achievements.

3. Verify that the section displays all earned points, badges, and rewards.

4. Check if the section displays information related to completed challenges and unlocked achievements.

5. Ensure that the progress information is up-to-date and reflects the latest activity and achievements.

**\*\*Postconditions:\*\***



\* Users should be able to access and view their progress and achievements in the dedicated section.

**\*\*Expected Result:\*\*** Users should be able to view their progress and achievements in a dedicated section within their profile, including earned points, badges, rewards, completed challenges, and unlocked achievements.

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

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

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

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

These test cases comprehensively cover the acceptance criteria for the gamification and rewards feature. They are designed to identify potential defects and ensure a high-quality software product.

-----

## **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 Creation Feature

**\*\*Project Name:\*\* Task Management Tool**

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

**\*\*Acceptance Criteria:\*\***

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

### \*\*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-01	Verify that users can enter a task title, description, due date, and priority level	This test case verifies				

that the user can successfully input all required information for a new task, including the task title, description, due date, and priority level. | Task Creation | High | The user is logged in. | Task Title: "Write project report", Description: "Complete the final project report for the client.", Due Date: 2024-03-15, Priority: High | 1.

Navigate to the "Create Task" page. 2. Enter the task title, description, due date, and priority level. 3. Click the "Create Task" button. | The task is created and displayed on the task list. | The task is created with the correct information. | Minor | High | Functional Testing | Positive |

| TC-02 | Verify that tasks can be assigned to specific team members | This test case verifies that a user can assign a task to a specific team member from the list of available team members. | Task Creation | High | The user is logged in, and a team exists with at least 2 members. | Team member: "John Doe", Task Title: "Design Website Mockup" | 1. Navigate to the "Create Task" page. 2. Enter the task title, description, due date, and priority level. 3. Select "John Doe" from the "Assign to" dropdown list. 4. Click the "Create Task" button. | The task is created and assigned to John Doe. | The task is successfully assigned to the selected team member. | Minor | High | Functional Testing | Positive |

| TC-03 | Verify that the system saves and displays created tasks | This test case verifies that the system saves the newly created task and displays it on the task list. | Task Creation | High | The user is logged in. | Task Title: "Schedule team meeting", Description: "Plan the next team meeting agenda", Due Date: 2024-03-20, Priority: Medium | 1. Navigate to the "Create Task" page. 2. Enter the task title, description, due date, and priority level. 3. Click the "Create Task" button. 4. Navigate to the "Task List" page. | The created task should appear on the task list with all entered details. | The created task is saved and displayed on the task list. | Minor | High | Functional Testing | Positive |

| TC-04 | Verify that users can edit task details after creation | This test case verifies that a user can modify the details of a previously created task. | Task Creation | Medium | The user is logged in, and a task exists. | Task Title: "Write project report", Description: "Complete the final project report for the client.", Due Date: 2024-03-15, Priority: High | 1. Navigate to the "Task List" page. 2. Select the "Write project report" task. 3. Click the "Edit Task" button. 4. Change the due date to 2024-03-22. 5. Click the "Save Task" button. | The task details are updated with the new due date. | The task details are successfully updated. | Minor | Medium | Functional Testing | Positive |

| TC-05 | Verify that the system validates required fields (e.g., task title) | This test case verifies that the system prevents the creation of a task without a valid task title. | Task Creation | High | The user is logged in. | No test data needed | 1. Navigate to the "Create Task" page. 2. Leave the task title field empty. 3. Click the "Create Task" button. | The system should display an error message indicating that the task title is required. | The system displays an error message indicating that the task title field is required. | Minor | High | Functional Testing | Negative |

---

## **Data - 22**

### **Project name -**

Task Management Tool

### **Project description -**

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

### **Feature name -**

Task Assignment

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

['Users can select team members to assign tasks.', 'Assignees receive notifications about their tasks.', 'The

system should display assigned tasks in the assignee's task list.", 'Users can reassign tasks to different team members.', 'Task assignment history should be tracked.']

### Generated Test Cases:

## ## Test Cases for Task Assignment Feature

**\*\*Project Name:\*\* Task Management Tool**

**\*\*Feature Name:\*\* Task Assignment**

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

**\*\*Acceptance Criteria:\*\***

1. Users can select team members to assign tasks.
2. Assignees receive notifications about their tasks.
3. The system should display assigned tasks in the assignee's task list.
4. Users can reassign tasks to different team members.
5. Task assignment history should be tracked.

### \*\*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
High	High	Unit Testing	White Box
Medium	Medium	Integration Testing	Black Box
Low	Low	Acceptance Testing	Black Box

-----

| TC\_TA\_01 | Verify that users can select team members to assign tasks. | This test case verifies that users can select team members from a list to assign tasks. | Task Assignment | High | - A task is created. <br> - At least two team members are available in the system. | No test data needed | 1. Create a new task. <br> 2. Navigate to the "Assign" section of the task. <br> 3. Click on the dropdown menu to select a team member. <br> 4. Choose a team member from the list. | - The selected team member is assigned to the task. | The chosen team member is successfully assigned to the task, and their name is displayed in the "Assigned To" field. | Major | High | Functional Testing | Positive |

| TC\_TA\_02 | Verify that assignees receive notifications about their assigned tasks. | This test case ensures that the system sends notifications to team members when they are assigned a task. | Task Assignment | High | - A user is logged into the system. <br> - A task is created and assigned to the user. | No test data needed | 1. Log in as a user. <br> 2. Create a new task and assign it to the logged-in user. <br> 3. Check the user's notifications within the system. | - The notification should be displayed in the notification center. | The user receives a notification about the newly assigned task. | Major | High | Functional Testing | Positive |

| TC\_TA\_03 | Verify that the system displays assigned tasks in the assignee's task list. | This test case verifies that assigned tasks are correctly displayed in the assigned team member's task list. | Task Assignment | High | - A user is logged into the system. <br> - A task is created and assigned to the user. | No test data needed | 1. Log in as the assigned user. <br> 2. Navigate to the user's task list. <br> 3. Check if the assigned task is displayed in the list. | - The task should appear in the user's task list. | The assigned task is visible in the assignee's task list with all relevant details. | Major | High | Functional Testing | Positive |

| TC\_TA\_04 | Verify that users can reassign tasks to different team members. | This test case ensures that users can reassign tasks to different team members after initial assignment. | Task Assignment | Medium | - A task is created and assigned to a team member. | No test data needed | 1. Create a task and assign it to a team member. <br> 2. Navigate to the task details page. <br> 3. Click on the "Reassign" button. <br> 4. Select a different team member from the list. | - The task is reassigned to the newly selected team member. | The task is successfully reassigned to the new team member, and the assigned team member's name updates accordingly. | Minor | Medium | Functional Testing | Positive |

| TC\_TA\_05 | Verify that task assignment history is tracked. | This test case validates that the system tracks all

the assignments and reassignments of a task. | Task Assignment | Medium | - A task is created. <br> - The task is assigned to multiple team members over time. | No test data needed | 1. Create a task and assign it to a team member. <br> 2. Reassign the task to a different team member. <br> 3. Access the task's history. | - The task history should show all assignments and reassignments. | The task history displays a complete record of all assignments and reassignments with dates and involved team members. | Minor | Medium | Functional Testing | Positive |

---

## **Data - 23**

### **Project name -**

Task Management Tool

### **Project description -**

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

### **Feature name -**

Task Deadline Reminders

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

**Generated Test Cases:**

## Test Cases for Task Deadline Reminders Feature

**Project:** Task Management Tool

**Feature:** Task Deadline Reminders

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

**Acceptance Criteria:**

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

**Test Case Template:**

**Test Case ID:** TC-TDR-XXX

**Test Case Title:** Verify that... (Feature/function being tested)

**Test Case Description:** (Brief description of the test case)



**\*\*Test Suite:\*\*** Task Deadline Reminders

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

**\*\*Preconditions:\*\***

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

\* A task with a deadline has been created.

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "No test data needed")

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

\* (List steps for executing the test case)

**\*\*Postconditions:\*\***

\* (Any follow-up actions after test execution)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Priority:\*\*** (High, Medium, Low)

**\*\*Type of Testing:\*\*** (Functional Testing)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

---

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

**\*\*Test Case ID:\*\*** TC-TDR-01

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

**\*\*Test Case Description:\*\*** This test case checks if a user can set a reminder for a task based on its deadline.

**\*\*Preconditions:\*\***

- \* User is logged in to the Task Management Tool.
- \* A task with a deadline has been created.

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

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

1. Access the task details page.
2. Locate the 'Reminder' setting option.
3. Enable the 'Reminder' setting.
4. Verify that the reminder is set for the task deadline.

**\*\*Postconditions:\*\*** The reminder setting for the task is enabled.

**\*\*Expected Result:\*\*** The reminder is successfully set for the task based on the deadline.

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

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

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

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

---

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

**\*\*Test Case ID:\*\*** TC-TDR-02

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

**\*\*Test Case Description:\*\*** This test case checks if users can adjust the urgency level of reminders.

**\*\*Preconditions:\*\***

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

\* A task with a deadline has been set with a reminder.

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

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

1. Access the task details page.
2. Locate the 'Reminder' settings.
3. Select a different urgency level (e.g., High, Medium, Low).
4. Save the settings.

**\*\*Postconditions:\*\*** The reminder urgency level for the task is updated.

**\*\*Expected Result:\*\*** The urgency level of the reminder is changed successfully.

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

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

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

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

---

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

**\*\*Test Case ID:\*\*** TC-TDR-03

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

**\*\*Test Case Description:\*\*** This test case checks if the system sends notifications for upcoming deadlines.

**\*\*Preconditions:\*\***

- \* User is logged in to the Task Management Tool.
- \* A task with a deadline has been set with a reminder.
- \* User has provided a valid email address or in-app notification setting.

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

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

1. Set a task deadline to be within the next 24 hours.
2. Monitor the user's email inbox or in-app notifications.

**\*\*Postconditions:\*\*** The user's email inbox is checked.

**\*\*Expected Result:\*\*** The system sends a notification about the upcoming deadline to the user's chosen delivery method.

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

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

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

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

---

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

**\*\*Test Case ID:\*\* TC-TDR-04**

**\*\*Test Case Title:\*\* Verify that users can choose the delivery method for reminders.**

**\*\*Test Case Description:\*\* This test case checks if users can select their preferred method for receiving task deadline reminders.**

**\*\*Preconditions:\*\***

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

\* A task with a deadline has been created.

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

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

1. Access the 'Settings' page.
2. Locate the 'Reminder Settings' section.
3. Select a preferred delivery method (e.g., email, in-app notification).
4. Save the settings.

**\*\*Postconditions:\*\*** The user's preferred reminder delivery method is saved in the system.

**\*\*Expected Result:\*\*** The system saves the chosen delivery method for reminders.

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

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

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

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

---

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

**\*\*Test Case ID:\*\*** TC-TDR-05

**\*\*Test Case Title:\*\*** Verify that reminder settings can be adjusted or disabled.

**\*\*Test Case Description:\*\*** This test case checks if users can adjust or disable reminder settings for tasks.

**\*\*Preconditions:\*\***

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

\* A task with a deadline has been set with a reminder.

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

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

1. Access the task details page.
2. Locate the 'Reminder' settings.
3. Change the reminder settings (e.g., change urgency level, delivery method, or disable the reminder completely).
4. Save the settings.

**\*\*Postconditions:\*\*** The reminder settings are updated or disabled according to the user's changes.

**\*\*Expected Result:\*\*** The reminder settings are adjusted or disabled successfully based on the user's actions.

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

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

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

**\*\*Test Case Approach:\*\*** Positive, Negative (for disabling reminders)

---

**\*\*Note:\*\*** This is a basic set of test cases. More specific test cases should be designed to cover different scenarios, edge cases, and user interactions. For example, you might want to test:



- \* Different types of notification formats (e.g., email, push notification, in-app message).
  - \* The functionality of reminder settings when multiple users are assigned to a task.
  - \* The behavior of reminder settings when the task deadline is changed.
  - \* The impact of user settings for time zones on reminder delivery.
  - \* The behavior of reminders for tasks with recurring deadlines.
- 

## **Data - 24**

### **Project name -**

Task Management Tool

### **Project description -**

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

### **Feature name -**

Task Prioritization

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

Generated Test Cases:

## Test Cases for Task Prioritization Feature

\*\*Project:\*\* Task Management Tool

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

\*\*Acceptance Criteria:\*\*

- 1. Users can set priority levels for tasks (High, Medium, Low).
- 2. Tasks can be sorted based on priority levels.
- 3. Users can change the priority level of a task.
- 4. The system visually distinguishes tasks based on priority.
- 5. Tasks without a specified priority default to Medium.

\*\*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-001	Verify that users can set priority levels for tasks	This test case verifies that users can assign priority levels (High, Medium, Low) to tasks within the task management tool.	Task Prioritization	High	User is logged in.	A new task with a title "Project Proposal"	1. Create a new task. 2. Select the "Priority" field. 3. Choose the "High" priority level. 4. Save the task.	Task is saved with the assigned priority.	The task "Project Proposal" is successfully created with the "High" priority level.	Minor	High	Functional Testing	Positive

| TC-002 | Verify that tasks can be sorted based on priority levels | This test case ensures that tasks can be sorted by priority levels, allowing users to prioritize their workload. | Task Prioritization | High | User is logged in. At least 3 tasks exist with different priority levels. | 3 tasks with different priority levels (High, Medium, Low). | 1. Access the task list. 2. Select the "Priority" sorting option. 3. Verify that tasks are displayed in ascending order of priority (Low, Medium, High). | Tasks are successfully sorted by priority level. | The tasks are displayed in the correct order based on their priority levels (Low, Medium, High). | Minor | High | Functional Testing | Positive |

| TC-003 | Verify that users can change the priority level of a task | This test case verifies that users can modify the priority of an existing task. | Task Prioritization | High | User is logged in. A task with an existing priority level exists. | A task titled "Meeting Preparation" with a "Medium" priority. | 1. Open the task "Meeting Preparation". 2. Select the "Priority" field. 3. Change the priority level to "High". 4. Save the task. | Task is saved with the updated priority level. | The task "Meeting Preparation" has its priority level successfully changed to "High". | Minor | High | Functional Testing | Positive |

| TC-004 | Verify that the system visually distinguishes tasks based on priority | This test case ensures that tasks are visually differentiated based on their priority levels to provide clear visual cues for users. | Task Prioritization | High | User is logged in. At least one task exists for each priority level. | Tasks with High, Medium, and Low priority levels. | 1. Access the task list. 2. Verify that tasks with "High" priority are visually distinct (e.g., different color, bold text). 3. Verify that tasks with "Medium" priority are visually distinct (e.g., different color, normal text). 4. Verify that tasks with "Low" priority are visually distinct (e.g., different color, less prominent text). | Tasks are visually distinct based on their priority levels. | Minor | High | Usability Testing | Positive |

| TC-005 | Verify that tasks without a specified priority default to Medium | This test case ensures that new tasks without a designated priority are automatically assigned the "Medium" priority level. | Task Prioritization | High | User is logged in. | No specific test data needed. | 1. Create a new task without selecting a priority level. 2. Save the task. 3. Verify that the task is automatically assigned a "Medium" priority level. | Task is automatically assigned a "Medium" priority level. | The newly created task has the default priority level set to "Medium". | Minor | High | Functional Testing | Positive |

**\*\*Note:\*\*** This test case template can be used as a base for creating additional test cases that cover various scenarios, edge cases, and negative testing aspects for the Task Prioritization feature.

-----

**Data - 25**

**Project name -**

Task Management Tool

**Project description -**

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

**Feature name -**

Subtasks

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

## ## Test Cases for Subtasks Feature

**\*\*Project Name:\*\* Task Management Tool**

**\*\*User Story Title:\*\* As a user, I want to create subtasks so that I can break down complex tasks into manageable parts.**

**\*\*Acceptance Criteria:\*\***

1. Users can create subtasks under main tasks.
2. Subtasks can have their own due dates and assignees.
3. The system tracks the progress of each subtask.
4. Subtasks can be marked as complete independently of the main task.
5. Dependencies between subtasks can be set.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC-SUB-01**

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

**\*\*Test Case Description:\*\* This test case verifies that users can create subtasks associated with a main task within the task management tool.**

**\*\*Test Suite:\*\* Subtasks Functionality**

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

**\*\*Preconditions:\*\***

- \* A user is logged in.
- \* A main task is created.

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

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

1. Navigate to the main task.
2. Click on the 'Add Subtask' button.
3. Enter a name for the subtask.
4. Click on the 'Save' button.

**\*\*Postconditions:\*\***

- \* A new subtask is created and displayed under the main task.

**\*\*Expected Result:\*\*** A new subtask is successfully created and associated with the main task.

**\*\*Severity:\*\*** Blocker

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

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

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

---

**\*\*Test Case ID:\*\* TC-SUB-02**

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

**\*\*Test Case Description:\*\* This test case verifies that subtasks can have independent due dates and assignees, distinct from the main task.**

**\*\*Test Suite:\*\* Subtasks Functionality**

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

**\*\*Preconditions:\*\***

- \* A user is logged in.
- \* A main task and a subtask are created.

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

- \* Subtask due date: "2024-03-15"
- \* Subtask assignee: "John Doe"

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

1. Navigate to the subtask.
2. Enter "2024-03-15" in the 'Due Date' field.
3. Select "John Doe" from the 'Assignee' dropdown.
4. Click on the 'Save' button.

**\*\*Postconditions:\*\***

\* The subtask's due date and assignee are updated accordingly.

**\*\*Expected Result:\*\*** The subtask's due date and assignee are updated and displayed correctly, separate from the main task's details.

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

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

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

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

---

**\*\*Test Case ID:\*\*** TC-SUB-03

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

**\*\*Test Case Description:\*\*** This test case verifies that the system accurately displays the progress status of each subtask, independent of the main task.

**\*\*Test Suite:\*\*** Subtasks Functionality

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

**\*\*Preconditions:\*\***



\* A user is logged in.

\* A main task and a subtask are created.

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

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

1. Navigate to the subtask.

2. Click on the 'Mark as Complete' button.

**\*\*Postconditions:\*\***

\* The subtask's status is updated to 'Completed'.

**\*\*Expected Result:\*\*** The subtask's status is correctly updated to 'Completed' and displayed in the system.

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

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

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

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

---

**\*\*Test Case ID:\*\*** TC-SUB-04

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

**\*\*Test Case Description:\*\*** This test case verifies that marking a subtask as complete does not automatically mark the main task as complete.

**\*\*Test Suite:\*\*** Subtasks Functionality

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

**\*\*Preconditions:\*\***

- \* A user is logged in.
- \* A main task and a subtask are created.

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

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

1. Navigate to the subtask.
2. Click on the 'Mark as Complete' button.
3. Navigate to the main task.

**\*\*Postconditions:\*\***

- \* The subtask's status is updated to 'Completed'.
- \* The main task's status remains unchanged.

**\*\*Expected Result:\*\*** The subtask is marked as completed without affecting the status of the main task.

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

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

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

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

---

**\*\*Test Case ID:\*\* TC-SUB-05**

**\*\*Test Case Title:\*\* Verify that dependencies between subtasks can be set.**

**\*\*Test Case Description:\*\* This test case verifies that users can establish dependencies between subtasks, ensuring that a subtask can only be marked as complete after its dependent subtask is completed.**

**\*\*Test Suite:\*\* Subtasks Functionality**

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

**\*\*Preconditions:\*\***

- \* A user is logged in.
- \* Two subtasks are created under the same main task.

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

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

1. Navigate to the first subtask.

2. Select the 'Add Dependency' option.
3. Choose the second subtask as the dependent subtask.
4. Click on the 'Save' button.
5. Navigate to the dependent subtask.
6. Click on the 'Mark as Complete' button.
7. Navigate back to the first subtask.
8. Try to mark the first subtask as complete.

**\*\*Postconditions:\*\***

- \* The dependency is successfully established.
- \* The first subtask cannot be marked as complete before the dependent subtask is marked as complete.

**\*\*Expected Result:\*\*** The dependency is established and enforced, preventing the completion of the first subtask until the dependent subtask is completed.

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

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

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

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

-----

**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

**\*\*Project Name:\*\* Task Management Tool**

**\*\*User Story Title:\*\* As a user, I want to set task dependencies so that tasks are completed in the correct sequence.**

**\*\*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\_01\_TD | Verify that users can set dependencies between tasks. | This test case verifies the user's ability to establish a dependency relationship between two tasks. The user will create two tasks and define one as dependent on the other. | Task Dependencies | High | User is logged in. | Task 1: "Write a proposal" | Task 2: "Send the proposal" | 1. Create a task named "Write a proposal". | 2. Create a task named "Send the proposal". | 3. Select "Send the proposal" task and set a dependency on "Write a proposal". | - The system should allow the user to set the dependency. | The dependency should be established successfully. | - Dependency relationship should be visible on both task details. | Minor | High | Functional Testing | Positive |

| TC\_02\_TD | Verify that the system displays dependencies in the task details. | This test case checks if the system accurately displays the dependency relationship between tasks on their individual detail views. | Task Dependencies | High | User is logged in. | Task 1: "Write a proposal" | Task 2: "Send the proposal" | 1. Create two tasks with dependency set as described in TC\_01\_TD. | 2. Navigate to the "Send the proposal" task detail view. | - The "Send the proposal" task details should display the dependency on "Write a proposal". | Dependency should be correctly displayed in the task details. | Minor | High | Functional Testing | Positive |

| TC\_03\_TD | Verify that tasks cannot be marked as complete if dependent tasks are not completed. | This test case ensures that the system prevents users from marking a task as complete if its dependent task remains incomplete. | Task Dependencies | High | User is logged in. | Task 1: "Write a proposal" | Task 2: "Send the proposal" | 1. Create two tasks with dependency set as described in TC\_01\_TD. | 2. Attempt to mark "Send the proposal" as complete without completing "Write a proposal". | - The system should not allow marking "Send the proposal" as complete. | An error message or notification should appear indicating the dependency requirement. | Minor | High | Functional Testing | Negative |

| TC\_04\_TD | Verify that users can update or remove dependencies as needed. | This test case validates the user's ability to modify or delete existing dependencies between tasks. | Task Dependencies | Medium | User is logged in. | Task 1: "Write a proposal" | Task 2: "Send the proposal" | 1. Create two tasks with dependency set as described in TC\_01\_TD. | 2. Update the dependency relationship to remove the dependency between tasks.

| - The system should allow the user to modify or remove the dependency. | - The dependency should be updated or removed successfully. | Minor | Medium | Functional Testing | Positive |

| TC\_05\_TD | Verify that the system provides alerts or warnings if a dependent task is not completed. | This test case ensures that the system provides appropriate notifications or warnings when a dependent task is overdue or remains incomplete. | Task Dependencies | High | User is logged in. | Task 1: "Write a proposal" | Task 2: "Send the proposal" | 1. Create two tasks with dependency set as described in TC\_01\_TD. | 2. Set a due date for "Write a proposal" and let it pass. | - The system should display a visual alert or notification indicating the overdue dependent task. | - The notification should clearly communicate the status of the dependent task and its impact on the other tasks. | Minor | High | Functional Testing | 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 Cases for Task Management Tool - Task Comments and Collaboration

**Project Name:** Task Management Tool

**User Story:** As a user, I want to comment on tasks so that I can collaborate with my team.

**Acceptance Criteria:**

1. Users can add comments to tasks.
2. Comments can include text, attachments, and links.
3. Users can reply to and edit their comments.
4. The system notifies users of new comments.
5. Comments are displayed in chronological order.

**Test Case Template:**

**Test Case ID:** TC-TaskComment-XXX

**Test Case Title:** Verify that...

**Test Case Description:** A brief description of the test case.

**Test Suite:** Task Management Tool - Collaboration



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

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* A task is created.

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

\* Sample text comment.

\* A sample image file.

\* A sample link to an external website.

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

\* Steps to execute the test case.

**\*\*Postconditions:\*\***

\* No postconditions required.

**\*\*Expected Result:\*\*** The desired outcome of the test case.

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

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

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

**\*\*Test Case Approach:\*\*** Positive/Negative

**\*\*Test Cases:\*\***

**\*\*1. TC-TaskComment-01\*\***

**\*\*Test Case Title:\*\*** Verify that users can add comments to tasks.

**\*\*Test Case Description:\*\*** This test case verifies that a user can successfully add a text comment to an existing task.

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

1. Navigate to the task details page.

2. Click on the comment section.
3. Enter a sample text comment.
4. Click "Post Comment".

**\*\*Expected Result:\*\*** The comment is successfully added to the task, displayed in the comment section, and the comment count is updated.

**\*\*2. TC-TaskComment-02\*\***

**\*\*Test Case Title:\*\*** Verify that comments can include text, attachments, and links.

**\*\*Test Case Description:\*\*** This test case verifies that users can add comments with text, attachments, and links.

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

1. Navigate to the task details page.
2. Click on the comment section.
3. Enter a sample text comment.
4. Attach a sample image file.
5. Enter a sample link to an external website.
6. Click "Post Comment".

**\*\*Expected Result:\*\*** The comment is successfully added to the task with the text, attachment, and link displayed correctly.

**\*\*3. TC-TaskComment-03\*\***

**\*\*Test Case Title:\*\*** Verify that users can reply to and edit their comments.

**\*\*Test Case Description:\*\*** This test case verifies that a user can reply to an existing comment and edit their own comments.

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

1. Navigate to the task details page.
2. Click on the comment section.

3. Post a comment.
4. Click on the reply button under the comment.
5. Enter a reply comment.
6. Click "Post Reply".
7. Click on the edit button under the comment.
8. Edit the comment.
9. Click "Save Changes".

**\*\*Expected Result:\*\*** The reply comment is successfully added and displayed below the original comment.  
The edited comment is updated and displayed correctly.

#### **\*\*4. TC-TaskComment-04\*\***

**\*\*Test Case Title:\*\*** Verify that the system notifies users of new comments.

**\*\*Test Case Description:\*\*** This test case verifies that the system sends a notification to the user when a new comment is added to a task they are assigned to or following.

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

1. Create a task and assign it to another user.
2. Log in as the assigned user.
3. Navigate to the task details page.
4. Log in as the first user.
5. Add a comment to the task.

**\*\*Expected Result:\*\*** The assigned user receives a notification that a new comment has been added to the task.

#### **\*\*5. TC-TaskComment-05\*\***

**\*\*Test Case Title:\*\*** Verify that comments are displayed in chronological order.

**\*\*Test Case Description:\*\*** This test case verifies that comments are displayed in chronological order, with the most recent comment at the top.

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

1. Navigate to the task details page.
2. Add a comment.
3. Wait a few seconds.
4. Add another comment.

**\*\*Expected Result:\*\*** The comments are displayed in chronological order, with the second comment displayed above the first comment.

**\*\*Notes:\*\***

- \* These test cases are just examples and can be expanded upon to cover various scenarios and edge cases.
- \* The specific steps and expected results may vary depending on the implementation of the software.
- \* It's important to test the feature in different browsers and devices to ensure compatibility.
- \* Remember to include negative test cases as well to check the system's handling of invalid data or unexpected inputs.

-----

**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

**\*\*Project Name:\*\* Task Management Tool**

**\*\*User Story:\*\*** As a user, I want to attach files to tasks so that all relevant information is easily accessible.

**\*\*Acceptance Criteria:\*\***

1. Users can attach multiple files to a task.
2. Attachments can include various file types (documents, images, etc.).
3. Users can view and download attachments.
4. The system limits the size of attachments to prevent data overload.
5. Users can remove attachments if needed.

**\*\*Test Case 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-01 | Verify that multiple files can be attached to a task. | This test case verifies that users can attach more than one file to a task. | File Attachments | High | \* User is logged in and has created a task. | \* Two different image files (e.g., JPG, PNG).<br> \* One PDF document. | 1. Open a task. <br> 2. Click on the "Attach File" button. <br> 3. Select the two image files. <br> 4. Click "Upload". <br> 5. Select the PDF document. <br> 6. Click "Upload". | \* The uploaded files should appear in the "Attachments" section of the task. | All three files are successfully uploaded and displayed in the attachments section. | Minor | High | Functional Testing | Positive |

| TC-02 | Verify that various file types can be attached. | This test case verifies that users can attach various types of files, including documents, images, and other common formats. | File Attachments | High | \* User is logged in and has created a task. | \* A Word document (.docx). <br> \* A spreadsheet (.xlsx). <br> \* A text file (.txt). | 1. Open a task. <br> 2. Click on the "Attach File" button. <br> 3. Select the Word document, spreadsheet, and text file. <br> 4. Click "Upload". | \* The uploaded files should appear in the "Attachments" section of the task. | All three files are successfully uploaded and displayed in the attachments section. | Minor | High | Functional Testing | Positive |

| TC-03 | Verify that users can view and download attachments. | This test case verifies that users can view the contents of attached files and download them to their local device. | File Attachments | High | \* User is logged in and has a task with attached files. | \* A document file (e.g., PDF). | 1. Open a task with an attached document file. <br> 2. Click on the document file name. <br> 3. Verify that the document preview is displayed. <br> 4. Click on the "Download" button. | \* The file should be downloaded to the user's device. | The document preview is displayed, and the file is successfully downloaded. | Minor | High | Functional Testing | Positive |

| TC-04 | Verify that the system limits attachment size. | This test case verifies that the system prevents users from uploading files exceeding the defined size limit. | File Attachments | High | \* User is logged in and has

created a task. | \* A large file exceeding the defined attachment size limit. | 1. Open a task. <br> 2. Click on the "Attach File" button. <br> 3. Select the large file. <br> 4. Click "Upload". | \* The system should display an error message indicating that the file size exceeds the allowed limit. | An error message is displayed, and the file upload is prevented. | Minor | High | Functional Testing | Negative |

| TC-05 | Verify that users can remove attachments. | This test case verifies that users can remove attached files from a task. | File Attachments | Medium | \* User is logged in and has a task with attached files. | \* A document file attached to the task. | 1. Open a task with an attached document file. <br> 2. Hover over the attached file. <br> 3. Click on the "Remove" button. | \* The attached file should be removed from the task. | The attached file is successfully removed from the task. | Minor | High | Functional Testing | Positive |

-----

## **Data - 29**

### **Project name -**

Task Management Tool

### **Project description -**

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

### **Feature name -**

Recurring Tasks

### **Feature description -**

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

### **Userstory title -**

As a user, I want to set recurring tasks so that regular activities are automated.

## Acceptance criteria -

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

## Generated Test Cases:

### ## Test Cases for Recurring Tasks Feature

**\*\*Project Name:\*\* Task Management Tool**

**\*\*User Story:\*\* As a user, I want to set recurring tasks so that regular activities are automated.**

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC\_RecurringTasks\_01**

**\*\*Test Case Title:\*\* Verify that users can set tasks to recur at specified intervals (daily, weekly, monthly).**

**\*\*Test Case Description:\*\* This test case verifies that the user interface provides options for setting the recurrence interval of a task to daily, weekly, or monthly.**

**\*\*Test Suite:\*\* Recurring Tasks**

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

**\*\*Preconditions:\*\***

\* A user is logged into the task management tool.

\* A task is created.

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

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

1. Navigate to the task details page.
2. Locate the "Recurrence" setting.



3. Verify that options for "Daily", "Weekly", and "Monthly" recurrence are available.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The user interface should display options for setting the recurrence interval to daily, weekly, or monthly.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_RecurringTasks\_02

**\*\*Test Case Title:\*\*** Verify that the system generates instances of recurring tasks automatically.

**\*\*Test Case Description:\*\*** This test case verifies that the system automatically creates instances of a recurring task based on the selected recurrence interval.

**\*\*Test Suite:\*\*** Recurring Tasks

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

**\*\*Preconditions:\*\***

\* A user is logged into the task management tool.

\* A task is created with a daily recurrence interval.

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

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

1. Create a task with a daily recurrence interval.

2. Wait for 24 hours.

3. Check the task list.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The system should have generated a new instance of the recurring task.

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

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

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

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

**\*\*Test Case ID:\*\* TC\_RecurringTasks\_03**

**\*\*Test Case Title:\*\* Verify that users can modify the recurrence pattern or cancel the recurrence.**

**\*\*Test Case Description:\*\* This test case verifies that users can change the recurrence interval or stop the recurring task.**

**\*\*Test Suite:\*\* Recurring Tasks**

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

**\*\*Preconditions:\*\***

\* A user is logged into the task management tool.

\* A task with a weekly recurrence interval is created.

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

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

1. Navigate to the task details page of the recurring task.
2. Modify the recurrence interval from weekly to monthly.
3. Alternatively, click on "Cancel Recurrence".

**\*\*Postconditions:\*\* None**

**\*\*Expected Result:\*\* The system should update the recurrence interval or stop generating instances of the recurring task.**

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

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

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

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

**\*\*Test Case ID:\*\* TC\_RecurringTasks\_04**

**\*\*Test Case Title:\*\* Verify that the system notifies users of upcoming instances of recurring tasks.**

**\*\*Test Case Description:\*\*** This test case verifies that the system sends notifications to the user before the due date of upcoming recurring task instances.

**\*\*Test Suite:\*\*** Recurring Tasks

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

**\*\*Preconditions:\*\***

\* A user is logged into the task management tool.

\* User notifications are enabled.

\* A task with a daily recurrence interval is created.

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

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

1. Create a recurring task with a daily interval.

2. Wait for the next day.

3. Check the user's notification settings.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The system should send a notification to the user regarding the upcoming instance of the recurring task.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_RecurringTasks\_05

**\*\*Test Case Title:\*\*** Verify that users can view a list of all instances of a recurring task.

**\*\*Test Case Description:\*\*** This test case verifies that users can view a list of all past and future instances of a recurring task.

**\*\*Test Suite:\*\*** Recurring Tasks

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

**\*\*Preconditions:\*\***

- \* A user is logged into the task management tool.
- \* A task with a weekly recurrence interval is created.

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

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

1. Navigate to the task details page of the recurring task.
2. Check for a "View Instances" option.
3. Click on the "View Instances" option.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The system should display a list of all past and future instances of the recurring task, including their due dates and completion status.

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

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

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

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

-----

## **Data - 30**

### **Project name -**

Task Management Tool

### **Project description -**

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

### **Feature name -**

Task Tags and Labels

## **Feature description -**

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

## **Userstory title -**

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

## **Acceptance criteria -**

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

## **Generated Test Cases:**

## Test Cases for Task Tags and Labels Feature

**\*\*Project Name:\*\* Task Management Tool**

**\*\*Feature Name:\*\* Task Tags and Labels**

**\*\*User Story 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:\*\***

1. Users can add tags and labels to tasks.
2. Tags can be customized and edited.
3. Tasks can be filtered and sorted based on tags and labels.
4. The system displays tags and labels prominently in the task list.
5. Users can remove tags and labels if needed.

**\*\*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-01	Verify that users can add tags and labels to tasks	This test case verifies that users can add tags and labels to tasks within the task management tool. It checks the functionality of adding tags and labels to new tasks and existing tasks.	Task Management	High	User is logged in and has access to the task list.	Task Name: "Complete Project Report"	1. Create a new task. 2. Enter the "Task Name" and "Description" of the task. 3. Click on the "Add Tag" or "Add Label" option. 4. Enter a new tag or label, such as "Project Report" or "High Priority." 5. Add the tag/label to the task.	The tag/label is successfully added to the task and is visible in the task details.	Passed	Trivial	Functional Testing	Positive	
TC-02	Verify that tags can be customized and edited	This test case verifies the ability to customize and edit tags within the task management tool. It checks if users can change the name of existing tags and add or remove tags.	Task Management	Medium	User is logged in and has access to the task list.	Tag Name: "Urgent"	1. Go to the "Tag Management" section. 2. Select the tag "Urgent." 3. Click on the "Edit Tag" option. 4. Change the tag name to "High Priority." 5. Save the changes.	The tag name is updated to "High Priority" and reflected throughout the task list.	Passed	Trivial	Functional Testing	Positive	
TC-03	Verify that tasks can be filtered and sorted based on tags and labels	This test case verifies that users can filter and sort tasks based on tags and labels. It checks the functionality of filtering and sorting tasks based on multiple tags and labels.	Task Management	High	User is logged in and has access to the task list.	Task 1: "Project Report" - Tagged: "Project Report," "High Priority." Task 2: "Code Review" - Tagged: "Code Review," "Medium Priority." Task 3: "Design Proposal" - Tagged: "Design Proposal," "Low Priority."	1. Navigate to the task list. 2. Select the filter option and choose "Project Report." 3. Verify that only tasks tagged with "Project Report" are displayed. 4. Apply a sort filter based on priority and verify that tasks are sorted correctly.	Only tasks tagged with "Project Report" are displayed in the list, and they are sorted based					

on priority. | Passed | Minor | Functional Testing | Positive |

| TC-04 | Verify that the system displays tags and labels prominently in the task list | This test case verifies the prominent display of tags and labels within the task list. It checks that tags and labels are easily identifiable and accessible in the task list view. | Task Management | High | User is logged in and has access to the task list. | Task Name: "Meeting Preparation" - Tagged: "Meeting," "Important," "Urgent." | 1. Navigate to the task list. 2. Verify that tags and labels are visible next to each task. 3. Ensure that tags and labels are clearly visible and easily distinguishable from the task description. | Tags and labels are prominently displayed in the task list, adjacent to the task name or description. | Passed | Minor | Usability Testing | Positive |

| TC-05 | Verify that users can remove tags and labels from tasks | This test case verifies the ability to remove tags and labels from tasks. It checks if users can remove existing tags and labels from both new and existing tasks. | Task Management | Medium | User is logged in and has access to the task list. | Task Name: "Update Website" - Tagged: "Website," "High Priority." | 1. Select the "Update Website" task. 2. Click on the "Remove Tag" or "Remove Label" option next to the tag/label. 3. Confirm the removal of the tag/label. | The tag/label is successfully removed from the task, and the task details are updated accordingly. | Passed | Trivial | Functional Testing | Positive |

-----

## Data - 31

### Project name -

Task Management Tool

### Project description -

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

### Feature name -

Task Status Tracking

## **Feature description -**

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

## **Userstory title -**

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

## **Acceptance criteria -**

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

## **Generated Test Cases:**

### **## Task Management Tool - Task Status Tracking Test Cases**

**\*\*Project Context:\*\***

**\* \*\*Project Name:\*\*** Task Management Tool

**\* \*\*Feature Name:\*\*** Task Status Tracking

**\* \*\*User Story Title:\*\*** As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

**\*\*Acceptance Criteria:\*\***

1. Users can update the status of tasks (e.g., To Do, In Progress, Completed).
2. The system displays task status prominently in the task list.
3. Users can view the progress percentage of tasks.
4. The system provides notifications for status changes.



5. Task status can be filtered and sorted.

**\*\*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-TST-001 | Verify that users can update the status of tasks | This test case verifies that users can change the status of a task from one predefined state to another, such as from "To Do" to "In Progress." | Task Status Tracking | High | User is logged in and has created a task. | A task with status "To Do" | 1. Open the task details page. 2. Click the status dropdown menu. 3. Select "In Progress" from the options. 4. Click "Save" or "Update." | Task status is updated to "In Progress" in the task list and task details page. | Task status is successfully updated. | Minor | High | Functional Testing | Positive |

| TC-TST-002 | Verify that the system displays task status prominently in the task list | This test case verifies that the task status is clearly visible and easily identifiable within the task list view. | Task Status Tracking | High | User is logged in and has a list of tasks with different statuses. | N/A | 1. Navigate to the task list view. 2. Observe the display of task statuses within the list. | Task status is displayed in a prominent and easily recognizable way (e.g., distinct color, icon, or label) within the task list view. | Minor | High | Functional Testing | Positive |

| TC-TST-003 | Verify that users can view the progress percentage of tasks | This test case verifies that users can view the progress percentage of tasks, indicating their completion level. | Task Status Tracking | High | User is logged in and has a task with a defined progress percentage. | A task with a progress percentage of 50%. | 1. Navigate to the task details page. 2. Observe the progress percentage displayed. | Progress percentage is displayed accurately and clearly for the task. | Minor | High | Functional Testing | Positive |

| TC-TST-004 | Verify that the system provides notifications for status changes | This test case verifies that the system sends notifications to relevant users when a task status is updated. | Task Status Tracking | High | User is logged in and has a task assigned to them. | A task assigned to the user with a status change from "To Do" to

"In Progress". | 1. Change the status of a task assigned to the user from "To Do" to "In Progress". 2. Check the user's notification settings and/or inbox. | The user receives a notification about the task status change. | Minor | High | Functional Testing | Positive |

| TC-TST-005 | Verify that task status can be filtered and sorted | This test case verifies that users can filter and sort tasks based on their status, allowing them to organize tasks according to their progress. | Task Status Tracking | Medium | User is logged in and has a list of tasks with different statuses. | N/A | 1. Navigate to the task list view. 2. Utilize the filtering options to display only tasks with a specific status (e.g., "In Progress"). 3. Sort the task list by status. | The task list is successfully filtered and sorted based on the chosen status. | Minor | Medium | Functional Testing | Positive |

**\*\*Objective:\*\*** These test cases aim to ensure the functionality and usability of the task status tracking feature within the Task Management Tool. They cover all the acceptance criteria and aim to identify potential issues early in the development cycle, contributing to a high-quality software product.

**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

**\*\*Project:\*\*** Task Management Tool

**\*\*Feature:\*\*** Task Filtering and Sorting

**\*\*User Story:\*\*** 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:\*\***

1. Users can filter tasks based on criteria such as due date, priority, assignee, or project.
2. Tasks can be sorted by various attributes.
3. The system provides multiple filter and sort options in the task list.
4. Users can save and apply custom filter and sort settings.
5. Filters and sorts do not affect the actual data, only the view.

## **\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC\_FS\_01**

**\*\*Test Case Title:\*\*** Verify that users can filter tasks based on due date.

**\*\*Test Case Description:\*\*** This test case verifies that users can filter tasks based on the due date criteria. The user will filter tasks based on a specific date range and ensure only tasks falling within that range are displayed.

**\*\*Test Suite:\*\*** Filtering and Sorting

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

**\*\*Preconditions:\*\***

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

\* At least 5 tasks exist with different due dates.

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

\* A specific date range for filtering (e.g., tasks due between March 10th and March 20th).

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

1. Navigate to the task list view.
2. Select the "Due Date" filter option.
3. Enter the specified date range in the filter input field.
4. Apply the filter.

**\*\*Postconditions:\*\***

\* The task list should display only tasks with due dates within the specified date range.

**\*\*Expected Result:\*\*** Only tasks with due dates within the specified date range are displayed.

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

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

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

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

## **\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TC\_FS\_02**

**\*\*Test Case Title:\*\* Verify that tasks can be sorted by priority.**

**\*\*Test Case Description:\*\* This test case verifies that tasks can be sorted in ascending or descending order based on priority level.**

**\*\*Test Suite:\*\* Filtering and Sorting**

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

**\*\*Preconditions:\*\***

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

\* At least 5 tasks exist with different priority levels (e.g., High, Medium, Low).

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

\* No test data needed.

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

1. Navigate to the task list view.
2. Select the "Priority" sort option.
3. Choose either ascending or descending order.
4. Apply the sort.

**\*\*Postconditions:\*\***

\* The task list should display tasks sorted by priority in the chosen order.

**\*\*Expected Result:\*\* Tasks are displayed sorted by priority in the chosen ascending or descending order.**

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

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

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

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

## **\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* TC\_FS\_03**

**\*\*Test Case Title:\*\* Verify that the system provides multiple filter and sort options in the task list.**

**\*\*Test Case Description:\*\* This test case verifies the availability of multiple filter and sort options in the task list view. The test will verify if options like due date, priority, assignee, project, and other relevant criteria are present.**

**\*\*Test Suite:\*\* Filtering and Sorting**

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

**\*\*Preconditions:\*\***

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

\* At least 5 tasks exist with different attributes for filtering and sorting.

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

\* No test data needed.

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

1. Navigate to the task list view.
2. Verify the presence of filter options for due date, priority, assignee, project, and other relevant criteria.
3. Verify the presence of sort options for priority, due date, assignee, project, and other relevant criteria.

**\*\*Postconditions:\*\***

\* No postconditions required.

**\*\*Expected Result:\*\* Multiple filter and sort options for different attributes are present in the task list view.**

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

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

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

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

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* TC\_FS\_04**

**\*\*Test Case Title:\*\* Verify that users can save and apply custom filter and sort settings.**

**\*\*Test Case Description:\*\* This test case verifies that users can save their custom filter and sort settings and apply them later. The user will create a custom filter, save it, and then apply the saved filter settings.**

**\*\*Test Suite:\*\* Filtering and Sorting**

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

**\*\*Preconditions:\*\***

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

\* At least 5 tasks exist with different attributes for filtering and sorting.

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

\* Custom filter and sort settings (e.g., filter by project "A" and sort by priority in descending order).

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

1. Navigate to the task list view.
2. Apply a custom filter and sort combination.
3. Save the filter and sort settings with a descriptive name.
4. Navigate away from the task list view.
5. Return to the task list view.
6. Select the saved custom filter and sort settings.

**\*\*Postconditions:\*\***

\* The task list should display tasks according to the saved filter and sort settings.

**\*\*Expected Result:\*\* The task list displays tasks according to the saved custom filter and sort settings.**

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

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

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

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

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC\_FS\_05**

**\*\*Test Case Title:\*\* Verify that filters and sorts do not affect the actual data, only the view.**

**\*\*Test Case Description:\*\* This test case verifies that filtering and sorting operations only affect the displayed view of the task list and do not modify the actual task data.**

**\*\*Test Suite:\*\* Filtering and Sorting**

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

**\*\*Preconditions:\*\***

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

\* At least 5 tasks exist with different attributes.

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

\* No test data needed.

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

1. Navigate to the task list view.
2. Apply a filter or sort to the task list.
3. Verify the task list view reflects the applied filter or sort.
4. Navigate to the detail view of a specific task within the filtered or sorted list.
5. Verify that the task details remain unchanged and do not reflect the applied filter or sort.

**\*\*Postconditions:\*\***

\* No postconditions required.

**\*\*Expected Result:\*\* The applied filter or sort only affects the view of the task list, and the actual task data remains unchanged.**

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

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

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

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



---

## 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 Title:\*\*** As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

**\*\*Acceptance Criteria:\*\***

1. Users can start and stop a timer for each task.
2. The system records and displays total time spent on tasks.
3. Users can manually enter or adjust time entries.
4. Time tracking data is stored and can be exported.
5. The system provides reports on time spent per task or project.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC\_TT\_01

**\*\*Test Case Title:\*\*** Verify that users can start and stop a timer for each task.

**\*\*Test Case Description:\*\*** This test case verifies that a user can initiate and terminate a timer associated with a specific task.

**\*\*Test Suite:\*\*** Time Tracking

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

**\*\*Preconditions:\*\***

\* A user is logged into the system.

\* A task is created and selected.

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

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

1. Navigate to the task and click the "Start Timer" button.
2. Observe that the timer starts and displays elapsed time.
3. After a few seconds, click the "Stop Timer" button.

4. Verify that the timer stops and the total elapsed time is recorded.

**\*\*Postconditions:\*\*** The timer is stopped, and the elapsed time is displayed.

**\*\*Expected Result:\*\*** The timer starts and stops successfully, accurately recording the elapsed time.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_TT\_02

**\*\*Test Case Title:\*\*** Verify that the system records and displays the total time spent on tasks.

**\*\*Test Case Description:\*\*** This test case verifies that the system accurately accumulates and presents the total time spent on each task.

**\*\*Test Suite:\*\*** Time Tracking

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

**\*\*Preconditions:\*\***

\* A user is logged into the system.

\* A task is created with multiple time entries.

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

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

1. Start and stop the timer on the task multiple times.
2. Manually enter or adjust some time entries for the task.
3. Navigate to the task details or a dedicated time tracking view.
4. Verify that the total time spent on the task is displayed accurately.

**\*\*Postconditions:\*\*** The total time spent on the task is displayed.

**\*\*Expected Result:\*\*** The total time spent on the task reflects the sum of all timer recordings and manual entries.

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

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

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

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

**\*\*Test Case ID:\*\* TC\_TT\_03**

**\*\*Test Case Title:\*\* Verify that users can manually enter or adjust time entries.**

**\*\*Test Case Description:\*\* This test case verifies that users have the ability to manually input or modify time entries for tasks, independent of the timer function.**

**\*\*Test Suite:\*\* Time Tracking**

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

**\*\*Preconditions:\*\***

\* A user is logged into the system.

\* A task is created.

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

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

1. Navigate to the task and click the "Add Time Entry" button.
2. Enter a start time, end time, and optionally a description for the time entry.
3. Verify that the entered time entry is displayed correctly.
4. Adjust the time entry and verify the changes are reflected.

**\*\*Postconditions:\*\* A new time entry is added and can be modified as needed.**

**\*\*Expected Result:\*\* Time entries can be manually created and edited accurately.**

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

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

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

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

**\*\*Test Case ID:\*\* TC\_TT\_04**

**\*\*Test Case Title:\*\* Verify that time tracking data is stored and can be exported.**

**\*\*Test Case Description:\*\* This test case verifies that the system stores time tracking data persistently and allows users to export it in a suitable format.**

**\*\*Test Suite:\*\* Time Tracking**

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

**\*\*Preconditions:\*\***

\* A user is logged into the system.

\* Multiple tasks with time entries exist.

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

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

1. Navigate to the time tracking section or report view.
2. Select an appropriate time range for data export.
3. Click the "Export" button and choose the desired format (e.g., CSV, PDF).
4. Download the exported file.
5. Verify that the exported file contains the correct time tracking data, including task names, dates, start/end times, and total time spent.

**\*\*Postconditions:\*\* The exported file is downloaded.**

**\*\*Expected Result:\*\* Time tracking data is stored persistently and can be exported in a chosen format, accurately reflecting the recorded information.**

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

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

**\*\*Type of Testing:\*\* System Testing, Integration Testing**

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

**\*\*Test Case ID:\*\* TC\_TT\_05**

**\*\*Test Case Title:\*\* Verify that the system provides reports on time spent per task or project.**

**\*\*Test Case Description:\*\* This test case verifies that the system generates reports that summarize time spent on individual tasks or projects, allowing for analysis and insights.**

**\*\*Test Suite:\*\* Time Tracking**

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

**\*\*Preconditions:\*\***

\* A user is logged into the system.

\* Multiple tasks are created, some grouped within projects.

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

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

1. Navigate to the reporting section or time tracking dashboard.
2. Select the desired reporting option (e.g., "Time Spent per Task", "Time Spent per Project").
3. Choose a time range and other filtering options as needed.
4. View the generated report.
5. Verify that the report accurately displays the time spent on each task or project within the chosen time frame.

**\*\*Postconditions:\*\* A report is generated and displayed.**

**\*\*Expected Result:\*\* The report provides a clear and accurate summary of time spent on tasks or projects, enabling analysis and insights into time management.**

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

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

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

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

-----

**Project name -**

Task Management Tool

**Project description -**

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

**Feature name -**

Task 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 Templates Feature

**\*\*Project:\*\*** Task Management Tool

**\*\*User Story:\*\*** As a user, I want to use task templates so that I can streamline task creation and ensure

consistency.

**\*\*Acceptance Criteria:\*\***

- 1. Users can create and save task templates.
- 2. Templates can include predefined task structures, descriptions, assignees, and due dates.
- 3. Users can apply templates to new tasks or projects.
- 4. Templates can be edited or deleted as needed.
- 5. The system displays a list of available templates for selection.

**\*\*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-TM-01	Verify that users can create and save task templates	This test case verifies that a user can create a new task template with a name, description, and predefined fields.	Task Templates	High	User is logged in	Template Name: "Project Kickoff", Template Description: "Initial tasks for project setup", Predefined Fields: Assignee, Due Date, Description	1. Navigate to the Task Templates section. 2. Click on "Create New Template". 3. Enter the template name and description. 4. Select "Assignee" and "Due Date" as predefined fields. 5. Click on "Save Template".	The template is saved successfully and is displayed in the list of available templates.	Template is successfully created and saved.	Minor	Medium	Functional Testing	Positive
TC-TM-02	Verify that templates can include predefined task structures, descriptions, assignees, and due dates	This test case verifies that a user can set a default assignee, due date, and description when creating a task template.	Task Templates	High	User is logged in, a template is created	Template Name: "Bug							



Report", Predefined Fields: Assignee: "Development Team", Due Date: "3 days", Description: "Provide detailed steps to reproduce the bug." | 1. Navigate to the "Bug Report" template. 2. Click on "Edit Template". 3. Set the Assignee to "Development Team". 4. Set the Due Date to "3 days". 5. Enter the default description. 6. Click on "Save Template". | The template is updated with the predefined fields. | Predefined fields are successfully saved to the template. | Minor | Medium | Functional Testing | Positive |

| TC-TM-03 | Verify that users can apply templates to new tasks or projects | This test case verifies that a user can select an existing template and apply it to a new task or project. | Task Templates | High | User is logged in, a template exists | Template Name: "Project Kickoff" | 1. Navigate to the "Create New Task" section. 2. Select "Use Template". 3. Choose the "Project Kickoff" template. 4. Click on "Create Task". | A new task is created with the predefined fields from the selected template. | Task is created with the predefined fields from the selected template. | Minor | Medium | Functional Testing | Positive |

| TC-TM-04 | Verify that templates can be edited or deleted as needed | This test case verifies that a user can edit and delete existing templates. | Task Templates | High | User is logged in, a template exists | Template Name: "Project Kickoff" | 1. Navigate to the "Project Kickoff" template. 2. Click on "Edit Template". 3. Change the template description. 4. Click on "Save Template". 5. Navigate to the "Project Kickoff" template. 6. Click on "Delete Template". | The template is successfully updated with the new description. The template is then deleted from the list of available templates. | Template is successfully edited and deleted. | Minor | Medium | Functional Testing | Positive |

| TC-TM-05 | Verify that the system displays a list of available templates for selection | This test case verifies that the system displays a list of existing templates to the user when creating a new task or project. | Task Templates | High | User is logged in, multiple templates exist | No test data needed | 1. Navigate to the "Create New Task" or "Create New Project" section. 2. Click on "Use Template". | The system displays a list of existing templates. | The list of available templates is displayed to the user. | Minor | Medium | Functional Testing | Positive |

**\*\*Note:\*\*** This is a sample set of test cases. Further test cases can be added based on the specific functionalities of the application and the needs of the project.

---

## Data - 35

### Project name -

Task Management Tool

### Project description -

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

### Feature name -

Calendar Integration

### Feature description -

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

### Userstory title -

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

### Acceptance criteria -

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

### Generated Test Cases:

## Test Cases for Calendar Integration Feature

\*\*Project Name:\*\* Task Management Tool

\*\*Feature Name:\*\* Calendar Integration

**\*\*User Story 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:\*\***

- 1. Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).
- 2. The system displays task deadlines in the calendar view.
- 3. Users can set reminders for tasks via the calendar.
- 4. Calendar events update automatically when tasks are changed.
- 5. Users can manage calendar sync settings within the app.

**\*\*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_01_CI	Verify that users can sync tasks with external calendars	This test case verifies that users can connect their Task Management Tool accounts to external calendars like Google Calendar and Outlook.	Calendar Integration	High	User is logged in to the Task Management Tool and has an active external calendar account.	N/A	1. Navigate to the settings section of the Task Management Tool. 2. Select the "Calendar Integration" option. 3. Choose the desired external calendar provider (e.g., Google Calendar, Outlook). 4. Enter login credentials for the external calendar account. 5. Authorize the Task Management Tool to access the external calendar.	User is successfully logged in to the external calendar. The Task Management Tool displays a confirmation message about the successful connection.	All tasks synced with the Task Management Tool should appear in the external calendar.	Blocker	High	Functional Testing, Integration Testing	Positive

| TC\_02\_CI | Verify that the system displays task deadlines in the calendar view | This test case validates that deadlines for tasks are displayed correctly within the external calendar. | Calendar Integration | High | User has synced tasks with an external calendar. | A task with a specific deadline. | 1. Create a task within the Task Management Tool with a deadline. 2. Verify that the task is synced to the external calendar. 3. Check the external calendar for the event corresponding to the task. 4. Verify that the event date and time match the task deadline. | The deadline should appear as an event on the external calendar with the correct date and time. | Major | High | Functional Testing, Integration Testing | Positive |

| TC\_03\_CI | Verify that users can set reminders for tasks via the calendar | This test case verifies that users can set reminders for tasks through the external calendar interface. | Calendar Integration | Medium | User has synced tasks with an external calendar. | A task with a deadline. | 1. Create a task with a deadline in the Task Management Tool. 2. Navigate to the external calendar and locate the corresponding event. 3. Set a reminder for the event. 4. Verify that the reminder is successfully set. | The reminder for the task should be set in the external calendar. The user should receive the reminder notification at the specified time. | Minor | Medium | Functional Testing, Integration Testing | Positive |

| TC\_04\_CI | Verify that calendar events update automatically when tasks are changed | This test case ensures that any changes made to a task in the Task Management Tool are automatically reflected in the external calendar. | Calendar Integration | High | User has synced tasks with an external calendar. | A task with a deadline. | 1. Create a task with a deadline in the Task Management Tool. 2. Modify the deadline of the task. 3. Check the corresponding event in the external calendar. 4. Verify that the event deadline has been updated accordingly. | The event in the external calendar should reflect the changes made to the task deadline. | Major | High | Functional Testing, Integration Testing | Positive |

| TC\_05\_CI | Verify that users can manage calendar sync settings within the app | This test case validates that users can control the sync settings for calendar integration. | Calendar Integration | Medium | User has synced tasks with an external calendar. | N/A | 1. Navigate to the settings section of the Task Management Tool. 2. Select the "Calendar Integration" option. 3. Modify the sync settings, such as the type of events to sync, sync frequency, etc. 4. Verify that the changes are applied correctly. | The sync settings should be adjusted based on the user's selections. The system should update the external calendar accordingly. | Minor | Medium |

**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

**\*\*Project:\*\* Task Management Tool**

**\*\*Feature:\*\* Task Duplication**

**\*\*User Story:\*\* As a user, I want to duplicate tasks so that I can save time when creating similar tasks.**

**\*\*Acceptance Criteria:\*\***

1. **\*\*Users can duplicate tasks or task templates.\*\***
2. **\*\*The duplicated task retains all original details, including descriptions, assignees, and deadlines.\*\***
3. **\*\*Users can modify the duplicated task if needed.\*\***
4. **\*\*The system clearly indicates that a task has been duplicated.\*\***
5. **\*\*Duplicated tasks are saved and displayed in the task list.\*\***

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC\_DUP\_01**

**\*\*Test Case Title:\*\* Verify that users can duplicate tasks.**

**\*\*Test Case Description:\*\* This test case verifies that a user can duplicate an existing task, creating a new copy of the task.**

**\*\*Test Suite:\*\* Task Management - Task Duplication**

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

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* A task exists within the user's task list.

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

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

1. Open the task list and select an existing task.

2. Click the "Duplicate Task" button or icon (if available).

3. Confirm the duplicate task creation prompt.

**\*\*Postconditions:\*\***

\* A new task with the same name as the original task, followed by "(copy)" should be created in the task list.

**\*\*Expected Result:\*\*** A duplicate task should be created and displayed in the task list.

**\*\*Severity:\*\*** Blocker

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

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

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

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** TC\_DUP\_02

**\*\*Test Case Title:\*\*** Verify that the duplicated task retains all original details.

**\*\*Test Case Description:\*\*** This test case verifies that the duplicated task retains all original details such as description, assignee, and deadline.

**\*\*Test Suite:\*\*** Task Management - Task Duplication

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

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* A task with a description, assignee, and deadline exists within the user's task list.

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

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

1. Open the task list and select an existing task with a description, assignee, and deadline.

2. Click the "Duplicate Task" button or icon (if available).

3. Confirm the duplicate task creation prompt.

4. View the newly created duplicated task.

**\*\*Postconditions:\*\***

- \* Compare the original task and the duplicated task.

**\*\*Expected Result:\*\*** The duplicated task should have the same description, assignee, and deadline as the original task.

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

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

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

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

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TC\_DUP\_03

**\*\*Test Case Title:\*\*** Verify that users can modify the duplicated task.

**\*\*Test Case Description:\*\*** This test case verifies that a user can modify the duplicated task's details, such as description, assignee, or deadline.

**\*\*Test Suite:\*\*** Task Management - Task Duplication

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

**\*\*Preconditions:\*\***

- \* User is logged into the task management tool.

- \* A task exists within the user's task list.

- \* The task has been duplicated.

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

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

1. Open the task list and select the duplicated task.
2. Modify the task's description, assignee, or deadline.
3. Save the changes.

**\*\*Postconditions:\*\***



\* Verify that the modified task is saved and displayed correctly.

**\*\*Expected Result:\*\*** The duplicated task should be editable, allowing users to make changes to its details.

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

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

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

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

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TC\_DUP\_04

**\*\*Test Case Title:\*\*** Verify that the system clearly indicates that a task has been duplicated.

**\*\*Test Case Description:\*\*** This test case verifies that the system provides a clear visual or textual indication that a task has been duplicated.

**\*\*Test Suite:\*\*** Task Management - Task Duplication

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

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* A task exists within the user's task list.

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

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

1. Open the task list and select an existing task.
2. Click the "Duplicate Task" button or icon (if available).
3. Confirm the duplicate task creation prompt.
4. Observe the system's response.

**\*\*Postconditions:\*\***

\* Verify the presence of a confirmation message, visual indicator (e.g., a notification), or a change in the duplicated task's name (e.g., "Task Name (Copy)") indicating successful duplication.

**\*\*Expected Result:\*\*** The system should provide a clear visual or textual indication that the task has been duplicated.

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

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

**\*\*Type of Testing:\*\*** Usability Testing

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

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** TC\_DUP\_05

**\*\*Test Case Title:\*\*** Verify that duplicated tasks are saved and displayed in the task list.

**\*\*Test Case Description:\*\*** This test case verifies that duplicated tasks are properly saved and displayed in the user's task list.

**\*\*Test Suite:\*\*** Task Management - Task Duplication

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

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* A task exists within the user's task list.

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

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

1. Open the task list and select an existing task.
2. Click the "Duplicate Task" button or icon (if available).
3. Confirm the duplicate task creation prompt.
4. Close and reopen the task list.

**\*\*Postconditions:\*\***

\* Verify that the duplicated task is still present in the task list.

**\*\*Expected Result:\*\*** The duplicated task should be saved and displayed correctly in the task list after

closing and reopening the list.

**\*\*Severity:\*\*** Blocker

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

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

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

-----

## **Data - 37**

### **Project name -**

Task Management Tool

### **Project description -**

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

### **Feature name -**

Task History

### **Feature description -**

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

### **Userstory title -**

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

### **Acceptance criteria -**

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

## Generated Test Cases:

### ## Task Management Tool - Task History Test Cases

#### \*\*Project Context:\*\*

\* Project Name: Task Management Tool

\* Feature Name: Task History

\* User Story Title: As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

#### \*\*Acceptance Criteria:\*\*

1. Users can view a detailed history of changes to tasks.
2. The history includes modifications, comments, and other relevant activity.
3. Users can filter the history by type of change or date.
4. The system provides timestamps for all changes.
5. Task history is accessible from the task details page.

#### \*\*Test Case Creation Template:\*\*

#### \*\*Test Case 1:\*\*

\* \*\*Test Case ID:\*\* TC-TH-01

\* \*\*Test Case Title:\*\* Verify that users can view a detailed history of changes to tasks.

\* \*\*Test Case Description:\*\* This test case validates the ability of users to access and view the complete history of changes made to a task. It verifies that all modifications, including updates to title, description,

assignee, due date, priority, and status are recorded and displayed in the history.

\* \*\*Test Suite:\*\* Task History

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

\* \*\*Preconditions:\*\*

- \* A user is logged into the Task Management Tool.

- \* A task has been created with multiple changes made to it (e.g., title updates, due date changes, comment additions, etc.)

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

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

1. Navigate to the task details page.

2. Click on the "History" tab or button.

3. Verify that a list of all changes made to the task is displayed.

4. Confirm that each change includes a description of the modification (e.g., "Title changed from 'Task A' to 'Task B'") and the timestamp of the change.

\* \*\*Postconditions:\*\* None

\* \*\*Expected Result:\*\* The history view displays a comprehensive and chronologically ordered list of all changes made to the task, including timestamps and descriptions for each modification.

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

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

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

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

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

\* \*\*Test Case ID:\*\* TC-TH-02

\* \*\*Test Case Title:\*\* Verify that the history includes modifications, comments, and other relevant activity.

\* \*\*Test Case Description:\*\* This test case checks that the task history accurately records different types of

user interactions, including task modifications, comments, and other actions like adding attachments or assigning collaborators.

\* \*\*Test Suite:\*\* Task History

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

\* \*\*Preconditions:\*\*

- \* A user is logged into the Task Management Tool.

- \* A task has been created with multiple changes made to it (e.g., adding a comment, attaching a file, changing the assignee, etc.)

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

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

1. Navigate to the task details page.

2. Click on the "History" tab or button.

3. Verify that the history includes entries for:

- \* Modifications: Title, description, assignee, due date, priority, status changes.

- \* Comments: Added comments by users.

- \* Other activity: Added attachments, added collaborators, etc.

4. Confirm that each entry includes a timestamp and a clear description of the activity.

\* \*\*Postconditions:\*\* None

\* \*\*Expected Result:\*\* The task history displays records for modifications, comments, and other relevant activity associated with the task, providing a complete audit trail.

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

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

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

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

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

\* \*\*Test Case ID:\*\* TC-TH-03

\* \*\*Test Case Title:\*\* Verify that users can filter the history by type of change or date.

\* \*\*Test Case Description:\*\* This test case validates the filtering functionality of the task history. It ensures that users can efficiently narrow down the history view by selecting specific types of changes or specifying a date range.

\* \*\*Test Suite:\*\* Task History

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

\* \*\*Preconditions:\*\*

- \* A user is logged into the Task Management Tool.

- \* A task has been created with a history of multiple changes over a period of time.

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

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

1. Navigate to the task details page.
2. Click on the "History" tab or button.
3. Select a filter option based on type of change (e.g., "Comment," "Assignee Change," "Due Date Change").
4. Verify that the history displays only the filtered changes.
5. Select a filter option based on date range (e.g., "Last Week," "This Month," "Custom Range").
6. Verify that the history displays only changes within the specified date range.

\* \*\*Postconditions:\*\* None

\* \*\*Expected Result:\*\* The history view accurately filters based on selected type of change or date range, displaying only the relevant entries.

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

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

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

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

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

**\* \*\*Test Case ID:\*\* TC-TH-04**

**\* \*\*Test Case Title:\*\* Verify that the system provides timestamps for all changes.**

**\* \*\*Test Case Description:\*\* This test case ensures that all changes recorded in the task history are associated with a timestamp, allowing for accurate tracking of when modifications occurred.**

**\* \*\*Test Suite:\*\* Task History**

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

**\* \*\*Preconditions:\*\***

- \* A user is logged into the Task Management Tool.**

- \* A task has been created with multiple changes made to it.**

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

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

- 1. Navigate to the task details page.**

- 2. Click on the "History" tab or button.**

- 3. Verify that each change in the history includes a timestamp indicating the time and date of the modification.**

**\* \*\*Postconditions:\*\* None**

**\* \*\*Expected Result:\*\* Every change entry in the task history displays a timestamp with an accurate date and time indicating when the modification occurred.**

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

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

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

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

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



- \* \*\*Test Case ID:\*\* TC-TH-05
- \* \*\*Test Case Title:\*\* Verify that task history is accessible from the task details page.
- \* \*\*Test Case Description:\*\* This test case verifies that the task history feature is readily accessible to users directly from the task details page.
- \* \*\*Test Suite:\*\* Task History
- \* \*\*Test Priority:\*\* High
- \* \*\*Preconditions:\*\*
  - \* A user is logged into the Task Management Tool.
  - \* A task has been created with a history of changes.
- \* \*\*Test Data:\*\* No test data needed.
- \* \*\*Test Steps:\*\*
  1. Navigate to the task details page.
  2. Verify that a dedicated "History" tab, button, or link is present on the page.
  3. Click on the "History" element.
  4. Verify that the task history view is displayed.
- \* \*\*Postconditions:\*\* None
- \* \*\*Expected Result:\*\* The task history is accessible through a clearly marked and easily located "History" element on the task details page.
- \* \*\*Severity:\*\* Critical
- \* \*\*Priority:\*\* High
- \* \*\*Type of Testing:\*\* Usability Testing
- \* \*\*Test Case Approach:\*\* Positive

These 5 test cases comprehensively cover all the acceptance criteria for the "Task History" feature, ensuring a robust testing process and contributing to the overall quality of the Task Management Tool.

-----

**Project name -**

Task Management Tool

**Project description -**

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

**Feature name -**

Task Notifications

**Feature description -**

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

**Userstory title -**

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

**Acceptance criteria -**

['Users receive notifications for task updates, comments, mentions, and other relevant activities.',  
'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.',  
'The system logs notification history for future reference.']

**Generated Test Cases:**

## Test Cases for Task Notifications Feature

**\*\*Project Name:\*\* Task Management Tool**

**\*\*Feature Name:\*\* Task Notifications**

**\*\*User Story Title:\*\* As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.**

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC\_TN\_01**

**\*\*Test Case Title:\*\* Verify that users receive notifications for task updates.**

**\*\*Test Case Description:\*\* This test case verifies that users receive notifications when tasks assigned to them are updated. It involves updating a task and checking if the assigned user receives a notification.**

**\*\*Test Suite:\*\* Task Notifications**

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

**\*\*Preconditions:\*\***

\* User is logged in to the system.

\* A task is assigned to the user.

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

\* Existing task with a user assigned.

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

1. Log in as the assigned user.
2. Open the assigned task and update the description.
3. Save the changes.
4. Verify that the user receives a notification about the task update.

**\*\*Postconditions:\*\***

\* User can view the notification in their notification center.

\* The updated task description is reflected in the notification.

**\*\*Expected Result:\*\* The assigned user receives a notification about the updated task description.**

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

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

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

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

**\*\*Test Case ID:\*\* TC\_TN\_02**

**\*\*Test Case Title:\*\* Verify that users receive notifications for comments on tasks.**

**\*\*Test Case Description:\*\*** This test case verifies that users assigned to a task receive notifications when someone comments on it. It involves adding a comment to a task and checking if the assigned user receives a notification.

**\*\*Test Suite:\*\* Task Notifications**

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

**\*\*Preconditions:\*\***

\* User is logged in to the system.

\* A task is assigned to the user.

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

\* Existing task with a user assigned.

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

1. Log in as a different user than the assigned user.
2. Open the assigned task and add a comment.
3. Save the comment.
4. Verify that the assigned user receives a notification about the new comment.

**\*\*Postconditions:\*\***

\* User can view the notification in their notification center.

\* The comment content is reflected in the notification.

**\*\*Expected Result:\*\*** The assigned user receives a notification about the new comment on the task.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_TN\_03**

**\*\*Test Case Title:\*\* Verify that users can customize notification settings.**

**\*\*Test Case Description:\*\* This test case verifies that users can customize notification settings based on their preferences, enabling or disabling different types of notifications.**

**\*\*Test Suite:\*\* Task Notifications**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

**\* User is logged in to the system.**

**\*\*Test Data:\*\***

**\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the user settings page.
2. Locate the notification settings section.
3. Enable/disable specific notification types (e.g., task updates, comments, mentions).
4. Save the changes.
5. Verify that the notification settings are updated according to the user's preferences.

**\*\*Postconditions:\*\***

**\* User's notification preferences are saved.**

**\*\*Expected Result:\*\* The notification settings are updated based on user preferences.**

**\*\*Severity:\*\* Minor**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_TN\_04**

**\*\*Test Case Title:\*\* Verify that notifications provide detailed information about the activity.**

**\*\*Test Case Description:\*\* This test case verifies that notifications provide detailed information about the activity that triggered them, allowing users to understand the context and take necessary actions.**

**\*\*Test Suite:\*\* Task Notifications**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* User is logged in to the system.

\* A task is assigned to the user.

**\*\*Test Data:\*\***

\* Existing task with a user assigned.

**\*\*Test Steps:\*\***

1. Log in as the assigned user.

2. Open a task and add a comment.

3. Verify that the notification received by the assigned user includes the following information:

\* Task title.

\* Task description (briefly).

\* Comment content.

\* The commenter's name or username.

\* The timestamp of the comment.

**\*\*Postconditions:\*\***

\* User can view the detailed information in the notification.

**\*\*Expected Result:\*\* The notification provides all relevant information about the activity.**

**\*\*Severity:\*\* Minor**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_TN\_05**

**\*\*Test Case Title:\*\* Verify that the system logs notification history for future reference.**

**\*\*Test Case Description:\*\* This test case verifies that the system logs all notification history, allowing users to review past notifications for reference.**

**\*\*Test Suite:\*\* Task Notifications**

**\*\*Test Priority:\*\* Low**

**\*\*Preconditions:\*\***

\* User is logged in to the system.

\* The user has received at least one notification.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Log in as the user.
2. Navigate to the notification history section.
3. Verify that all previously received notifications are logged.
4. Check the details of a logged notification (e.g., timestamp, activity type, content).

**\*\*Postconditions:\*\***

\* User can view the complete notification history.

**\*\*Expected Result:\*\* All received notifications are logged in the notification history.**

**\*\*Severity:\*\* Minor**

**\*\*Priority:\*\* Low**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

These test cases cover all acceptance criteria mentioned in the user story for the Task Notifications feature.

They aim to ensure the functionality meets the user needs and provides a comprehensive and robust

notification system.

---

## **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:**

## Test Cases for Task Archiving Feature



**\*\*Project Name:\*\* Task Management Tool**

**\*\*Feature Name:\*\* Task Archiving**

**\*\*User Story:\*\* As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks.**

**\*\*Acceptance Criteria:\*\***

1. Users can archive completed or inactive tasks.
2. Archived tasks are moved to a separate section for easy access.
3. Users can search and view archived tasks.
4. The system provides an option to restore archived tasks to the active list.
5. Archived tasks retain all original details and history.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\* Task Archiving**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged in to the task management tool.

\* User has created at least one completed or inactive task.

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

\* The archived task should be removed from the active task list.

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Cases:\*\***

**\*\*Test Case ID:\*\* TC\_01**

**\*\*Test Case Title:\*\* Verify that users can archive completed tasks.**

**\*\*Test Case Description:\*\* This test case verifies that a user can successfully archive a completed task.**

**\*\*Test Data:\*\* A completed task with a title, description, and deadline.**

**\*\*Test Steps:\*\***

1. Open the task management tool and navigate to the task list.
2. Select the completed task to be archived.
3. Click on the "Archive" option for the selected task.
4. Confirm the archive action.

**\*\*Expected Result:\*\* The task should be moved to the archived section.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_02**

**\*\*Test Case Title:\*\* Verify that archived tasks are moved to a separate section.**

**\*\*Test Case Description:\*\* This test case ensures that archived tasks are moved to a dedicated section and are**

distinct from active tasks.

**\*\*Test Data:\*\*** A completed task that has been archived.

**\*\*Test Steps:\*\***

1. Archive a completed task as per TC\_01.
2. Navigate to the "Archived Tasks" section.
3. Verify that the archived task is displayed in the "Archived Tasks" section.
4. Verify that the archived task is no longer present in the active task list.

**\*\*Expected Result:\*\*** The archived task should be present only in the "Archived Tasks" section and not in the active task list.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_03

**\*\*Test Case Title:\*\*** Verify that users can search and view archived tasks.

**\*\*Test Case Description:\*\*** This test case checks if users can search for and view specific archived tasks based on keywords.

**\*\*Test Data:\*\*** Multiple archived tasks with different titles and descriptions.

**\*\*Test Steps:\*\***

1. Navigate to the "Archived Tasks" section.
2. Enter a keyword from the title or description of an archived task into the search bar.
3. Perform the search operation.

**\*\*Expected Result:\*\*** The search should return only those archived tasks that contain the specified keyword in their title or description.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_04**

**\*\*Test Case Title:\*\* Verify that the system provides an option to restore archived tasks to the active list.**

**\*\*Test Case Description:\*\* This test case verifies that users can easily restore an archived task back to the active task list.**

**\*\*Test Data:\*\* An archived task.**

**\*\*Test Steps:\*\***

1. Navigate to the "Archived Tasks" section.
2. Select the archived task to be restored.
3. Click on the "Restore" option for the selected task.

**\*\*Expected Result:\*\* The selected archived task should be moved back to the active task list.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_05**

**\*\*Test Case Title:\*\* Verify that archived tasks retain all original details and history.**

**\*\*Test Case Description:\*\* This test case ensures that all original details and history associated with an archived task are preserved.**

**\*\*Test Data:\*\* An archived task with a title, description, deadline, and comments.**

**\*\*Test Steps:\*\***

1. Archive a task with details as per test data.
2. Navigate to the "Archived Tasks" section.
3. View the archived task and verify that all original details (title, description, deadline, comments) are

present.

**\*\*Expected Result:\*\*** The archived task should retain all original details and history, including any comments or updates made before archiving.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

These test cases cover all the acceptance criteria for the task archiving feature and provide a thorough evaluation of the functionality. By executing these tests, you can ensure that the feature works as expected and meets the user's needs.

-----

**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:\*\*** Task Management Tool

**\*\*Feature:\*\*** 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.

**\*\*Acceptance Criteria:\*\***

1. Users can invite external collaborators as guests with limited access.
2. Guests can view and collaborate on specific tasks or projects.
3. The system provides clear distinctions between guest and full access roles.
4. Users can manage and revoke guest access at any time.
5. Guests cannot access sensitive or restricted areas of the platform.

**\*\*Test Case 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-GA-01 | Verify that users can invite external collaborators as guests with limited access | This test case verifies the functionality of inviting external collaborators as guests and assigning limited access to them. The user will attempt to invite a guest with a specific email address and set their access level to "Guest". | Guest Access | High | User is logged in with an active account. | Guest email address: guest@example.com | 1. Navigate to the "Team" section. 2. Click on the "Invite Guest" button. 3. Enter the guest email address. 4. Select "Guest" access level. 5. Click "Invite". | Guest receives an invitation email with a link to access the platform. | The guest receives an email with a unique invitation link and is successfully added as a guest with limited access to the platform. | Minor | Medium | Functional Testing | Positive |

| TC-GA-02 | Verify that guests can view and collaborate on specific tasks or projects | This test case verifies that guests can access specific tasks or projects and participate in collaboration activities like adding comments and attachments. | Guest Access | High | A guest user is logged in with a valid invitation link. | Task ID: 1234, Project ID: 5678 | 1. Access the platform using the guest invitation link. 2. Navigate to the designated task or project. 3. View the task details, comments, and attachments. 4. Add a comment to the task. 5. Upload an attachment to the task. | The guest can successfully access the assigned task or project, view details, add comments, and upload attachments. | Minor | Medium | Functional Testing | Positive |

| TC-GA-03 | Verify that the system provides clear distinctions between guest and full access roles | This test case verifies that the system clearly distinguishes between guest and full access users by visually differentiating their interfaces or roles displayed on the platform. | Guest Access | High | User is logged in with a full access account, Guest user is logged in with a guest account. | No test data needed | 1. Log in as a full access user. 2. Log in as a guest user. 3. Observe the user interfaces for both roles. | The system visually differentiates the interface or displays the user's role as "Guest" or "Full Access" accordingly. | Major | Medium | Functional Testing | Positive |

| TC-GA-04 | Verify that users can manage and revoke guest access at any time | This test case verifies that

users with full access can manage and revoke guest access for invited collaborators. | Guest Access | High |

User is logged in with a full access account. | Guest email address: guest@example.com | 1. Navigate to the "Team" section. 2. Locate the guest user. 3. Click on the "Manage Access" button. 4. Change the access level to "No Access". 5. Click "Save". | The guest user's access is revoked, and they can no longer access the platform. | Minor | Medium | Functional Testing | Negative |

| TC-GA-05 | Verify that guests cannot access sensitive or restricted areas of the platform | This test case verifies that guests are restricted from accessing sensitive areas of the platform, such as user profiles, account settings, or billing information. | Guest Access | High | Guest user is logged in with a guest account. | No test data needed | 1. Attempt to access the user profile section. 2. Attempt to access the account settings section. 3. Attempt to access the billing information section. | Guests are prevented from accessing sensitive areas and redirected to an appropriate message or error page. | Critical | High | Functional Testing, Security Testing | 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:**

## Test Cases for Task Management Tool - Task Import/Export Feature

**\*\*Project Context:\*\***

\* **\*\*Project Name:\*\*** Task Management Tool

\* **\*\*Feature Name:\*\*** Task Import/Export

\* **\*\*User Story:\*\*** As a user, I want to import and export tasks so that I can migrate data between different task management tools.

**\*\*Acceptance Criteria:\*\***

1. Users can import tasks from external sources in various formats (e.g., CSV, Excel).
2. Users can export tasks to other task management tools or file formats.
3. The system provides clear mapping of fields during import/export.
4. Users can preview imported data before finalizing the process.
5. The system logs import/export activities for future reference.

**\*\*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-IMP-01	Verify that users can import tasks from CSV files	This test case verifies the functionality of importing tasks from a CSV file containing task details. The test case will assess if the system correctly imports the data and maps it to the appropriate fields.	Import/Export	High	User is logged in, a CSV file containing task details is available	Sample CSV file with task data (title, description, due date, assignee, etc.)	1. Navigate to the "Import" section. 2. Select CSV as the import format. 3. Choose the sample CSV file. 4. Review the field mapping and confirm the import.	The system successfully imports the tasks from the CSV file, and the data is accurately mapped to the corresponding fields.	Minor	High	Functional Testing	Positive	
TC-EXP-02	Verify that users can export tasks in JSON format	This test case verifies the functionality of exporting tasks in JSON format. The test case assesses if the system correctly generates a JSON file containing the selected tasks' data.	Import/Export	High	User is logged in, tasks are created in the system	No test data needed	1. Select multiple tasks to export. 2. Choose JSON as the export format. 3. Download the exported JSON file.	The system successfully creates a JSON file containing the exported tasks' data in the desired format.	Minor	High	Functional Testing	Positive	
TC-IMP-03	Verify that the system provides clear mapping of fields during CSV import	This test case verifies that the system provides clear mapping of fields during the import process from a CSV file. The test case assesses if the system correctly identifies and maps the CSV columns to the corresponding task fields.	Import/Export	Medium	User is logged in, a CSV file containing task details is available	Sample CSV file with task data (title, description, due date, assignee, etc.)	1. Navigate to the "Import" section. 2. Select CSV as the import format. 3. Choose the sample CSV file. 4. Review the field mapping options.	The system accurately identifies and displays the mapping of each CSV column to the corresponding task field, providing					

clear and understandable labels. | Minor | Medium | Functional Testing | Positive |

| TC-IMP-04 | Verify that users can preview imported data before finalizing the import process | This test case verifies that the user can preview the imported data before confirming the import process. The test case assesses if the system displays a preview of the imported data, allowing the user to verify accuracy before proceeding. | Import/Export | High | User is logged in, a CSV file containing task details is available | Sample CSV file with task data (title, description, due date, assignee, etc.) | 1. Navigate to the "Import" section. 2. Select CSV as the import format. 3. Choose the sample CSV file. 4. Review the field mapping and click "Preview". | The system displays a preview of the imported data, allowing the user to review the imported task details and ensure accuracy before confirming the import. | Minor | High | Functional Testing | Positive |

| TC-IMP-05 | Verify that the system logs import/export activities | This test case verifies that the system logs all import and export activities for future reference. The test case assesses if the system records details such as the file used, date and time of the operation, and user who performed the action. | Import/Export | Medium | User is logged in, import/export actions are performed | No test data needed | 1. Import tasks from a CSV file. 2. Export tasks in JSON format. | The system records details about the import and export activities, including the file used, date and time of the operation, and the user who performed the action. | Minor | Medium | Functional Testing, System Testing | Positive |

**\*\*Note:\*\*** These are just examples, and you can add more test cases based on the specific requirements of your project.

-----

**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

**\*\*Project Name:\*\* Task Management Tool**

**\*\*User Story:\*\*** As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

**\*\*Acceptance Criteria:\*\***

1. Users can set and adjust task priorities.
2. Tasks can be sorted and filtered based on priority levels.

- 3. The system provides visual indicators of task priority.
- 4. Users receive notifications for high-priority tasks.
- 5. Priority changes are logged and tracked.

**\*\*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-TP-01	Verify that users can set task priorities	This test case verifies that users can successfully assign priority levels to tasks.	Task Prioritization	High	User is logged in and has created a task.	Task: "Write Project Proposal", Priority levels: High, Medium, Low	1. Navigate to the task "Write Project Proposal". 2. Select the "Priority" dropdown. 3. Choose "High" priority level. 4. Save the task.	Task priority is updated in the task list and the task details page.	The "Write Project Proposal" task should now display as high priority.	Major	High	Functional Testing	Positive
TC-TP-02	Verify that users can adjust task priorities	This test case checks that users can modify task priorities after initial assignment.	Task Prioritization	High	User is logged in and has a task with a set priority.	Task: "Draft Marketing Plan", Initial priority: Medium	1. Navigate to the "Draft Marketing Plan" task. 2. Click the "Priority" dropdown. 3. Select "Low" priority level. 4. Save the task.	Task priority is updated in the task list and the task details page.	The "Draft Marketing Plan" task should now display as low priority.	Major	High	Functional Testing	Positive
TC-TP-03	Verify that tasks can be sorted and filtered based on priority levels	This test case checks that the task list can be sorted and filtered based on different priority levels.	Task Prioritization	High	User is logged in and has tasks with various priorities.	Multiple tasks with different priority levels.	1. Navigate to the task list. 2. Click the "Priority" filter dropdown. 3. Select "High" priority level.	The task list displays only tasks with "High" priority.	Only tasks with "High" priority are displayed in the list.	Major	High	Functional Testing	Positive

| TC-TP-04 | Verify that the system provides visual indicators of task priority | This test case checks that the system clearly displays task priorities using visual indicators. | Task Prioritization | High | User is logged in and has a task with a set priority. | Task: "Schedule Team Meeting", Priority: High | 1. Navigate to the "Schedule Team Meeting" task. 2. Observe the visual indicators representing the task priority. | Visual indicators like color-coding or icons clearly show the "High" priority of the task. | The task "Schedule Team Meeting" should be visually distinguished from other tasks based on its priority level. | Major | High | Usability Testing | Positive |

| TC-TP-05 | Verify that users receive notifications for high-priority tasks | This test case checks that users receive notifications for upcoming deadlines for high-priority tasks. | Task Prioritization | High | User is logged in and has a high-priority task with a deadline set. | Task: "Finalize Project Report", Deadline: Today, Priority: High | 1. Set a deadline for the "Finalize Project Report" task as today. 2. Observe notification channels (e.g., email, in-app notifications). | Users receive notifications (email and/or in-app) regarding the high-priority "Finalize Project Report" task deadline. | Users should receive notifications about the approaching deadline for the "Finalize Project Report" task. | Major | High | System Testing | Positive |

-----

**Data - 43**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Collaboration

**Feature description -**

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

#### **Userstory title -**

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay updated on task progress.

#### **Acceptance criteria -**

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

#### **Generated Test Cases:**

## Test Cases for Task Collaboration Feature:

**\*\*Project:\*\*** Task Management Tool

**\*\*Feature:\*\*** Task Collaboration

**\*\*User Story:\*\*** 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:\*\***

1. Users can assign tasks to team members.
2. Users can share updates and communicate through comments or messages.
3. Task updates and comments are visible to assigned team members.
4. Users receive notifications for task updates and comments.

5. The system provides a clear view of task collaboration history.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-COLLAB-01

**\*\*Test Case Title:\*\*** Verify that users can assign tasks to team members.

**\*\*Test Case Description:\*\*** This test case verifies that a user can assign a task to another team member. It will check if the assignment is reflected in the task details and if the assigned user is notified about the assignment.

**\*\*Test Suite:\*\*** Task Collaboration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User A and User B are members of the same team.

\* User A has created a task.

**\*\*Test Data:\*\***

\* Task Title: "Draft project proposal"

\* Assignee: User B

**\*\*Test Steps:\*\***



1. User A opens the task "Draft project proposal".
2. User A clicks on the "Assign" button.
3. User A selects User B from the list of team members.
4. User A saves the task assignment.

**\*\*Postconditions:\*\***

- \* The task details should show User B as the assigned user.
- \* User B should receive a notification about the task assignment.

**\*\*Expected Result:\*\*** User B is successfully assigned to the task, and both User A and User B can see the assignment reflected in the task details.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*\_\_\_\_\_\*\***

**\*\*Test Case ID:\*\*** TC-COLLAB-02

**\*\*Test Case Title:\*\*** Verify that users can share updates and communicate through comments or messages.

**\*\*Test Case Description:\*\*** This test case verifies that users can add comments and updates to a task. It will check if the comments and updates are saved and visible to assigned team members.

**\*\*Test Suite:\*\*** Task Collaboration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* User A and User B are members of the same team.
- \* User A has created a task and assigned it to User B.

**\*\*Test Data:\*\***

- \* Task Title: "Draft project proposal"
- \* Update: "Completed first draft, awaiting review"
- \* Comment: "Looks good so far, just a few minor edits needed"

**\*\*Test Steps:\*\***

1. User A opens the task "Draft project proposal".
2. User A adds an update: "Completed first draft, awaiting review".
3. User A adds a comment: "Looks good so far, just a few minor edits needed".
4. User A saves the updates and comment.

**\*\*Postconditions:\*\***

\* The updates and comment should be visible to both User A and User B in the task details.

**\*\*Expected Result:\*\*** Both the update and comment are successfully added to the task and visible to the assigned users.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*-----\*\***

**\*\*Test Case ID:\*\*** TC-COLLAB-03

**\*\*Test Case Title:\*\*** Verify that task updates and comments are visible to assigned team members.

**\*\*Test Case Description:\*\*** This test case verifies that only assigned team members can see the task updates and comments. It will check if a user who is not assigned to the task can access the updates and comments.

**\*\*Test Suite:\*\*** Task Collaboration

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User A, User B, and User C are members of the same team.

\* User A created a task and assigned it to User B.

\* User A added an update and a comment to the task.

**\*\*Test Data:\*\***

\* No test data needed

**\*\*Test Steps:\*\***

1. User C attempts to access the task details.

**\*\*Postconditions:\*\***

\* User C should not be able to see the task updates and comments.

**\*\*Expected Result:\*\*** Only the assigned user (User B) and the creator (User A) can view the updates and comments. User C, not assigned to the task, should not be able to see the updates and comments.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Negative

\*\*-----\*\*

\*\*Test Case ID:\*\* TC-COLLAB-04

\*\*Test Case Title:\*\* Verify that users receive notifications for task updates and comments.

\*\*Test Case Description:\*\* This test case verifies that users receive notifications when updates and comments are added to a task they are assigned to. It will check if notifications are delivered to the user's email address or within the application interface.

\*\*Test Suite:\*\* Task Collaboration

\*\*Test Priority:\*\* High

\*\*Preconditions:\*\*

- \* User A and User B are members of the same team.
- \* User A created a task and assigned it to User B.
- \* User B has set up notification preferences for task updates and comments.

\*\*Test Data:\*\*

- \* No test data needed

\*\*Test Steps:\*\*

1. User A adds an update and a comment to the task.

**\*\*Postconditions:\*\***

\* User B should receive a notification about the update and comment through email or within the application interface.

**\*\*Expected Result:\*\*** User B receives a notification for both the update and the comment, indicating the task name, update content, and comment content.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*\_\_\_\_\_\*\***

**\*\*Test Case ID:\*\*** TC-COLLAB-05

**\*\*Test Case Title:\*\*** Verify that the system provides a clear view of task collaboration history.

**\*\*Test Case Description:\*\*** This test case verifies that the system provides a clear history of all task updates, comments, and assignments. It will check if the history is organized chronologically and includes details about the action performed, the user who performed it, and the time of the action.

**\*\*Test Suite:\*\*** Task Collaboration

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- \* User A and User B are members of the same team.
- \* User A created a task and assigned it to User B.
- \* User A added updates and comments to the task.
- \* User B completed the task.

**\*\*Test Data:\*\***

- \* No test data needed

**\*\*Test Steps:\*\***

1. User A opens the task details.
2. User A accesses the task collaboration history.

**\*\*Postconditions:\*\***

- \* The task collaboration history should display all actions taken on the task, including assignments, updates, comments, and task completion.
- \* The history should be organized chronologically, with the latest action at the top.
- \* Each action should show the user who performed it, the time of the action, and the content of the update or

comment.

**\*\*Expected Result:\*\*** The system provides a clear and detailed history of all collaboration activities on the task, including updates, comments, and assignments, organized chronologically and displaying relevant information about each action.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*-----\*\***

These test cases cover the key aspects of the task collaboration feature and will help identify potential defects early in the development cycle. This will ensure the functionality meets the acceptance criteria and provides a smooth and effective collaborative experience for users.

-----

**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:**

## Test Cases for Task Tracking Feature

**\*\*Project:\*\*** Task Management Tool

**\*\*User Story:\*\*** As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

**\*\*Acceptance Criteria:\*\***

1. Users can view task statuses.
2. Users can track task progress and deadlines.
3. Tasks are updated in real-time.
4. System provides visual indicators for task progress.

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_01

**\*\*Test Case Title:\*\*** Verify that users can view task statuses.

**\*\*Test Case Description:\*\*** This test case verifies that users can see the current status of a task.

**\*\*Test Suite:\*\*** Task Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* A task has been created and assigned a status.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list.

2. Select a task.

3. Observe the task details.

**\*\*Postconditions:\*\***

\* Task details are displayed.

**\*\*Expected Result:\*\*** The task status should be clearly visible within the task details.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_02

**\*\*Test Case Title:\*\*** Verify that users can track task progress and deadlines.

**\*\*Test Case Description:\*\*** This test case verifies that users can see the progress of a task towards completion and its associated deadline.

**\*\*Test Suite:\*\*** Task Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* A task has been created with a deadline and assigned a status.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list.
2. Select a task.
3. Observe the task details.
4. Check the task progress bar/indicator and the remaining time until deadline.

**\*\*Postconditions:\*\***

\* Task details are displayed.

**\*\*Expected Result:\*\*** The task details should display a progress bar or indicator reflecting task progress, and the deadline should be clearly visible.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_03

**\*\*Test Case Title:\*\*** Verify that tasks are updated in real-time.

**\*\*Test Case Description:\*\*** This test case verifies that changes to task status, progress, and deadline are reflected immediately in the system.

**\*\*Test Suite:\*\*** Task Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* A task has been created and assigned a status.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list.
2. Select a task.
3. Change the task status or update the progress.
4. Observe the task details for immediate reflection of the changes.

**\*\*Postconditions:\*\***

\* Task details are updated in real-time.

**\*\*Expected Result:\*\*** The task status, progress, and deadline should update in real-time without requiring a page refresh.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_04

**\*\*Test Case Title:\*\*** Verify that the system provides visual indicators for task progress.

**\*\*Test Case Description:\*\*** This test case verifies that the system utilizes visual cues to indicate task progress

and completion status.

**\*\*Test Suite:\*\*** Task Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged into the task management tool.

\* Multiple tasks with different progress statuses are available.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list.

2. Observe the tasks and their visual indicators.

**\*\*Postconditions:\*\***

\* Task list is displayed.

**\*\*Expected Result:\*\*** Tasks with different statuses should have distinct visual indicators (e.g., colors, icons, progress bars) that clearly convey their progress and completion status.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Usability Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_05

**\*\*Test Case Title:\*\*** Verify that tasks are updated in real-time for multiple users.

**\*\*Test Case Description:\*\*** This test case verifies that task updates made by one user are reflected in real-time for other users collaborating on the task.

**\*\*Test Suite:\*\*** Task Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* Two users are logged into the task management tool.
- \* A task has been created and shared between the two users.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. User 1 navigates to the task list and selects the shared task.
2. User 1 updates the task status or progress.
3. User 2 observes the task details.

**\*\*Postconditions:\*\***

- \* Task details are updated in real-time for both users.

**\*\*Expected Result:\*\*** User 2 should see the updated task status and progress immediately after User 1 makes the changes.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Integration 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.

**Acceptance Criteria:**

1. Users receive notifications for new tasks.
2. Users receive notifications for task updates and comments.
3. Users receive reminders for upcoming deadlines.
4. Notification settings can be customized by users.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-TN-01

**\*\*Test Case Title:\*\*** Verify that users receive notifications for new tasks.

**\*\*Test Case Description:\*\*** This test case verifies that users receive notifications 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.

\* The user has enabled notifications for new tasks in their settings.

**\*\*Test Data:\*\***

\* New task details: title, description, due date, assigned user.

**\*\*Test Steps:\*\***

1. Create a new task and assign it to the test user.
2. Verify that the test user receives a notification about the new task.
3. Check the notification content for accuracy: task title, due date, assigned user.

**\*\*Postconditions:\*\***

\* The newly created task should appear in the user's task list.

**\*\*Expected Result:\*\*** The test user receives a notification about the new task within a reasonable timeframe.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-TN-02

**\*\*Test Case Title:\*\*** Verify that users receive notifications for task updates and comments.



**\*\*Test Case Description:\*\*** This test case verifies that users receive notifications when a task they are involved in is updated or receives comments.

**\*\*Test Suite:\*\*** Task Notifications

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* A user account is created and logged in.
- \* A task is created and assigned to the user.
- \* The user has enabled notifications for task updates and comments in their settings.

**\*\*Test Data:\*\***

- \* Task updates: Change due date, add/edit description, add attachment, mark as complete.
- \* Comments: Add a new comment to the task.

**\*\*Test Steps:\*\***

1. Update the existing task by changing its due date, description, adding an attachment, or marking it as complete.
2. Add a comment to the existing task.
3. Verify that the test user receives a notification about each update and comment.
4. Check the notification content for accuracy: task title, type of update, comment content.

**\*\*Postconditions:\*\***

- \* The task should reflect the updates made in the user's task list.
- \* The comment should be visible in the task's comment section.

**\*\*Expected Result:\*\*** The test user receives notifications about each task update and comment within a reasonable timeframe.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\* TC-TN-03**

**\*\*Test Case Title:\*\* Verify that users receive reminders for upcoming deadlines.**

**\*\*Test Case Description:\*\* This test case verifies that users receive timely reminders for tasks nearing their deadlines.**

**\*\*Test Suite:\*\* Task Notifications**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* A task with a deadline is created and assigned to the user.

\* The user has enabled notifications for deadlines in their settings.

**\*\*Test Data:\*\***

\* Task with deadline set to within a specific timeframe (e.g., 24 hours, 1 week).

**\*\*Test Steps:\*\***

1. Create a task with a deadline set for within a specific timeframe (e.g., 24 hours, 1 week).

2. Verify that the user receives a reminder notification about the upcoming deadline before the actual deadline.

3. Check the notification content for accuracy: task title, due date, time until deadline.

**\*\*Postconditions:\*\***

\* The task should remain in the user's task list with the deadline marked.

**\*\*Expected Result:\*\* The test user receives a timely reminder notification about the task deadline.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-TN-04**

**\*\*Test Case Title:\*\* Verify that users can customize notification settings.**

**\*\*Test Case Description:\*\*** This test case verifies that users can adjust their notification settings to suit their preferences.

**\*\*Test Suite:\*\*** Task Notifications

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the user's settings page.
2. Access the notification settings section.
3. Verify that options to customize notifications for different events (new tasks, updates, comments, deadlines) are available.
4. Adjust notification settings for different events (enable/disable, choose frequency).
5. Save the updated notification settings.
6. Verify that the notification settings have been successfully updated.

**\*\*Postconditions:\*\***

\* The user's notification preferences should be reflected in subsequent notifications.

**\*\*Expected Result:\*\*** The user can customize notification settings for different task events to their preference.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-TN-05

**\*\*Test Case Title:\*\*** Verify that users cannot receive notifications for events they have disabled.

**\*\*Test Case Description:\*\*** This test case verifies that users do not receive notifications for events that they

have specifically disabled in their notification settings.

**\*\*Test Suite:\*\*** Task Notifications

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* A task with a deadline is created and assigned to the user.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the user's settings page and disable notifications for task deadlines.
2. Create a task with a deadline set within a reasonable timeframe.
3. Verify that the user does not receive a deadline reminder notification for the new task.

**\*\*Postconditions:\*\***

\* The user's notification settings should remain unchanged.

**\*\*Expected Result:\*\*** The test user does not receive notifications for events they have disabled in their settings.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Negative

**\*\*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:**

## Test Cases for Task Categorization Feature

**Project:** Task Management Tool

**Feature:** Task Categorization

**User Story:** As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

**Acceptance Criteria:**

1. Users can create and assign tasks to different categories or projects.
2. Users can view tasks grouped by categories or projects.
3. Tasks can be moved between categories or projects.
4. Category or project information is accurately reflected in task views.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-TaskCat-01

**\*\*Test Case Title:\*\*** Verify that users can create and assign tasks to different categories.

**\*\*Test Case Description:\*\*** This test case verifies that a user can create a new category and assign a task to it.

**\*\*Test Suite:\*\*** Task Categorization

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in to the task management tool.

\* User has the permission to create categories.

**\*\*Test Data:\*\***

\* Category Name: "Project A"

\* Task Title: "Write project proposal"

**\*\*Test Steps:\*\***

1. Navigate to the "Categories" section.
2. Click on the "Create Category" button.
3. Enter "Project A" as the category name and click "Save".
4. Navigate to the "Tasks" section.
5. Click on the "Create Task" button.
6. Enter "Write project proposal" as the task title and select "Project A" from the category dropdown.
7. Click on "Save".

**\*\*Postconditions:\*\***

\* The new category "Project A" should be visible in the "Categories" section.

\* The new task "Write project proposal" should be visible in the "Tasks" section and should be assigned to the "Project A" category.

**\*\*Expected Result:\*\*** The user should be able to successfully create a new category and assign a task to it.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-TaskCat-02

**\*\*Test Case Title:\*\*** Verify that users can view tasks grouped by categories.

**\*\*Test Case Description:\*\*** This test case verifies that a user can view a list of tasks grouped by different categories.

**\*\*Test Suite:\*\*** Task Categorization

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in to the task management tool.

\* At least two categories and two tasks exist, with each task assigned to a different category.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the "Tasks" section.
2. Select the "Category" filter dropdown.
3. Choose a specific category from the dropdown.
4. Verify that only the tasks belonging to the selected category are displayed.

**\*\*Postconditions:\*\*** No postconditions.

**\*\*Expected Result:\*\*** The user should be able to view a list of tasks grouped by the selected category.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-TaskCat-03**

**\*\*Test Case Title:\*\* Verify that tasks can be moved between categories.**

**\*\*Test Case Description:\*\* This test case verifies that a user can move a task from one category to another.**

**\*\*Test Suite:\*\* Task Categorization**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged in to the task management tool.

\* At least two categories and one task exist, with the task assigned to one of the categories.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the "Tasks" section.
2. Select a task assigned to a specific category.
3. Click on the "Edit" button for the selected task.
4. Change the category of the task to a different category.
5. Click on "Save".

**\*\*Postconditions:\*\***

\* The task should be removed from its original category.

\* The task should be visible in the new category.

**\*\*Expected Result:\*\* The user should be able to successfully move a task from one category to another.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**



**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-TaskCat-04**

**\*\*Test Case Title:\*\* Verify that category information is accurately reflected in task views.**

**\*\*Test Case Description:\*\* This test case verifies that the category information assigned to a task is displayed correctly in the task view.**

**\*\*Test Suite:\*\* Task Categorization**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged in to the task management tool.

\* At least one task assigned to a category exists.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the "Tasks" section.
2. Select a task that is assigned to a category.
3. Verify that the category name is correctly displayed in the task details view.

**\*\*Postconditions:\*\* No postconditions.**

**\*\*Expected Result:\*\* The category information for the task should be accurately displayed in the task view.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-TaskCat-05**

**\*\*Test Case Title:\*\* Verify that users can create a project and assign tasks to it.**

**\*\*Test Case Description:\*\* This test case verifies that a user can create a new project and assign a task to it.**

**\*\*Test Suite:\*\* Task Categorization**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged in to the task management tool.

\* User has the permission to create projects.

**\*\*Test Data:\*\***

\* Project Name: "Website Development"

\* Task Title: "Design website homepage"

**\*\*Test Steps:\*\***

1. Navigate to the "Projects" section.

2. Click on the "Create Project" button.

3. Enter "Website Development" as the project name and click "Save".

4. Navigate to the "Tasks" section.

5. Click on the "Create Task" button.

6. Enter "Design website homepage" as the task title and select "Website Development" from the project dropdown.

7. Click on "Save".

**\*\*Postconditions:\*\***

\* The new project "Website Development" should be visible in the "Projects" section.

\* The new task "Design website homepage" should be visible in the "Tasks" section and should be assigned to the "Website Development" project.

**\*\*Expected Result:\*\*** The user should be able to successfully create a new project and assign a task to it.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

-----

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Search

**Feature description -**

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

**Userstory title -**

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

**Acceptance criteria -**

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

**Generated Test Cases:**

## Task Management Tool - Task Search Test Cases

**\*\*Project Name:\*\*** Task Management Tool

**\*\*Feature Name:\*\*** Task Search

**\*\*User Story 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.

**\*\*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-TS-01 | Verify that users can search for tasks by name. | This test case checks the functionality of searching for tasks based on their name. It includes a positive scenario with a matching task and a negative scenario where no task matches the search term. | Task Search | High | - A task with the name "Meeting with Client" exists in the system. | - Search Term: "Meeting with Client" | 1. Open the task management tool. 2. Enter "Meeting with Client" in the search bar. 3. Click the search button. | - The search results should display. | - The search results should display the task "Meeting with Client". | Minor | High | Functional Testing | Positive and Negative |

| TC-TS-02 | Verify that users can search for tasks by description. | This test case ensures that users can search for tasks based on their descriptions. It includes a positive scenario where the search term matches a task description and a negative scenario where no task description matches the search term. | Task Search | High | - A task with the description "Discuss project proposal" exists in the system. | - Search Term: "project proposal" | 1. Open the task management tool. 2. Enter "project proposal" in the search bar. 3. Click the search button. | - The search results should display. | - The search results should display the task with the description "Discuss project proposal". | Minor | High | Functional Testing | Positive and Negative |

| TC-TS-03 | Verify that users can filter search results by category. | This test case verifies the functionality of filtering search results based on the task category. It checks if the filter works for a specific category and displays only tasks belonging to that category. | Task Search | High | - Tasks are categorized into "Development", "Design", and "Marketing". - At least one task exists in each category. | - Category filter: "Design" | 1. Open the task management tool. 2. Enter a search term (e.g., "Task" or leave it empty). 3. Select "Design" from the category filter. 4. Click the filter button. | - The search results should be filtered. | - Only

tasks belonging to the "Design" category should be displayed. | Minor | High | Functional Testing | Positive |

| TC-TS-04 | Verify that users can filter search results by priority. | This test case checks the functionality of filtering search results by priority level. It verifies if the filter works correctly for a specific priority level and displays only tasks with that priority. | Task Search | High | - Tasks are prioritized as "High", "Medium", and "Low". - At least one task exists with each priority level. | - Priority filter: "High" | 1. Open the task management tool. 2. Enter a search term (e.g., "Task" or leave it empty). 3. Select "High" from the priority filter. 4. Click the filter button. | - The search results should be filtered. | - Only tasks with "High" priority should be displayed. | Minor | High | Functional Testing | Positive |

| TC-TS-05 | Verify that users can filter search results by due date. | This test case checks if users can filter search results based on due dates. It verifies if the filter works correctly for a specific date range and displays only tasks within that date range. | Task Search | High | - Tasks have various due dates. | - Due date range: From "2023-12-15" to "2023-12-22" | 1. Open the task management tool. 2. Enter a search term (e.g., "Task" or leave it empty). 3. Set the due date range from "2023-12-15" to "2023-12-22". 4. Click the filter button. | - The search results should be filtered. | - Only tasks with due dates between "2023-12-15" and "2023-12-22" should be displayed. | Minor | High | Functional Testing | 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 Collaboration Feature

**Project Name:** Task Management Tool

**User Story 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.

**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-01	Verify that users can add comments to tasks	This test case verifies that a user can add a comment to an existing task.	Task Collaboration	High	User must be logged in and have access to a task.	- Comment text: "This is a test comment."	1. Navigate to a task. 2. Click on the "Add Comment" field. 3. Enter the comment text. 4. Click on the "Submit" button.	- Comment should be visible on the task.	The comment is successfully added and displayed on the task.	Minor	High	Functional Testing	Positive

| TC-02 | Verify that users can attach files to tasks | This test case verifies that a user can attach a file to an existing task. | Task Collaboration | High | User must be logged in and have access to a task. | - File to be attached: "test\_document.pdf" | 1. Navigate to a task. 2. Click on the "Attach File" button. 3. Select the file "test\_document.pdf". 4. Click on the "Upload" button. | - Attached file should be visible on the task. | The file is successfully attached and displayed on the task. | Minor | High | Functional Testing | Positive |

| TC-03 | Verify that users can share tasks with other users | This test case verifies that a user can share an existing task with another user. | Task Collaboration | High | User must be logged in and have access to a task. | - User to share with: "user2@example.com" | 1. Navigate to a task. 2. Click on the "Share Task" button. 3. Enter the email address "user2@example.com". 4. Click on the "Share" button. | - User "user2@example.com" should receive an email notification about the shared task. | The task is successfully shared with the specified user, and a notification email is sent. | Minor | High | Functional Testing | Positive |

| TC-04 | Verify that task collaboration features work seamlessly | This test case verifies that adding comments, attaching files, and sharing tasks can be performed seamlessly without errors or interruptions. | Task Collaboration | Medium | Users must be logged in and have access to a task. | - Comment text: "Test comment" - File to be attached: "test\_image.jpg" - User to share with: "user3@example.com" | 1. Navigate to a task. 2. Add a comment. 3. Attach a file. 4. Share the task with "user3@example.com". 5. Verify that all actions are performed successfully without any errors. | - All actions should be completed without errors or interruptions. | All actions related to task collaboration are performed seamlessly without any errors or interruptions. | Minor | Medium | Functional Testing | Positive |

| TC-05 | Verify that users cannot share tasks with non-existing users | This test case verifies that the system prevents sharing tasks with invalid email addresses. | Task Collaboration | Medium | User must be logged in and have access to a task. | - Invalid email address: "invalid\_email@example.com" | 1. Navigate to a task. 2. Click on the "Share Task" button. 3. Enter the invalid email address "invalid\_email@example.com". 4. Click on the "Share" button. | - An error message should be displayed indicating that the email address is invalid. | An error message is displayed, and the task is not shared with the invalid email address. | Minor | Medium | Functional Testing | Negative |

---

## **Data - 49**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Course Creation Tools

### **Feature description -**

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to structure content effectively, ensuring an engaging and informative learning experience for students.

### **Userstory title -**

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

### **Acceptance criteria -**

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

### **Generated Test Cases:**

## Test Cases for Course Creation Tools

\*\*Project Name:\*\* Online Learning Platform

\*\*Feature Name:\*\* Course Creation Tools



**\*\*User Story Title:\*\*** As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

**\*\*Acceptance Criteria:\*\***

1. **\*\*Instructors can create courses using videos, presentations, quizzes, and assignments.\*\***
2. **\*\*The tools are user-friendly and require minimal technical skills.\*\***
3. **\*\*Courses can be previewed before publishing.\*\***
4. **\*\*Instructors can save courses as drafts.\*\***
5. **\*\*Courses can be edited post-publishing.\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_001

**\*\*Test Case Title:\*\*** Verify that instructors can create courses using videos, presentations, quizzes, and assignments.

**\*\*Test Case Description:\*\*** This test case verifies that instructors are able to add different types of content, including videos, presentations, quizzes, and assignments, to their courses.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* Instructor account is created and logged in.

\* Course creation page is accessible.

**\*\*Test Data:\*\***

\* Sample video file.

\* Sample presentation file (e.g., PowerPoint).

\* Sample quiz with multiple-choice questions.

\* Sample assignment with instructions and due date.

**\*\*Test Steps:\*\***

1. Navigate to the course creation page.
2. Create a new course.
3. Add a video file to the course content.
4. Add a presentation file to the course content.
5. Add a quiz with multiple-choice questions to the course content.
6. Add an assignment with instructions and due date to the course content.

**\*\*Postconditions:\*\***

\* The course is saved with all added content.

**\*\*Expected Result:\*\*** All added content is successfully integrated into the course, and the course can be saved and viewed with all content elements.

**\*\*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 course creation tools are user-friendly and require minimal technical skills.

**\*\*Test Case Description:\*\*** This test case validates that instructors without advanced technical skills can easily create and manage courses using the intuitive interface of the course creation tools.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* Instructor account is created and logged in.

\* Course creation page is accessible.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the course creation page.
2. Attempt to create a new course without referring to any instructions or help guides.
3. Add content elements (video, presentation, quiz, assignment) to the course.
4. Preview the course content.
5. Save the course as a draft.
6. Publish the course.

**\*\*Postconditions:\*\***

\* The course is created successfully.

**\*\*Expected Result:\*\***

\* The instructor is able to complete all steps without encountering difficulties.

\* The interface is easy to understand and navigate.

\* The tools are intuitive and self-explanatory.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Usability Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_003

**\*\*Test Case Title:\*\*** Verify that courses can be previewed before publishing.

**\*\*Test Case Description:\*\*** This test case ensures that instructors can preview their courses before making them available to students.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* Instructor account is created and logged in.

\* A course is created with at least one content element.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the course creation page.

2. Select a course.

3. Click on the "Preview" button.

**\*\*Postconditions:\*\***

\* The course content is displayed in a preview mode.

**\*\*Expected Result:\*\***

\* The course preview accurately reflects the content, layout, and functionality of the published course.

\* The instructor can view the course from the student's perspective.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_004**

**\*\*Test Case Title:\*\* Verify that instructors can save courses as drafts.**

**\*\*Test Case Description:\*\* This test case ensures that instructors can save their incomplete courses as drafts, allowing them to resume editing later.**

**\*\*Test Suite:\*\* Course Creation Tools**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- \* Instructor account is created and logged in.
- \* A course is created with at least one content element.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the course creation page.
2. Select a course.
3. Click on the "Save as Draft" button.

**\*\*Postconditions:\*\***

- \* The course is saved as a draft.

**\*\*Expected Result:\*\***

- \* The course is saved with all the current content and progress.
- \* The course can be accessed later from the "Drafts" section of the course creation page.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_005

**\*\*Test Case Title:\*\*** Verify that courses can be edited post-publishing.

**\*\*Test Case Description:\*\*** This test case ensures that instructors can modify their courses after they are published.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* Instructor account is created and logged in.

\* A course is created and published.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the published course page.

2. Click on the "Edit" button.

3. Make changes to the course content (e.g., add a new video, edit the quiz questions).

4. Save the changes.

**\*\*Postconditions:\*\***

\* The course is updated with the latest changes.

**\*\*Expected Result:\*\***

\* The course is updated with the new content or modifications.

\* The changes are reflected in the published course.

\* The published date of the course is not altered.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 50**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Interactive Lessons

### **Feature description -**

Lessons are thoughtfully designed to foster learner engagement through interactive components such as quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these interactive elements enhance comprehension and retention of course material.

### **Userstory title -**

As a learner, I want interactive lessons so that I can engage with the content more effectively.

### **Acceptance criteria -**

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided after each interactive component.', 'Interactive lessons can be accessed on various devices.']

### **Generated Test Cases:**

## Test Cases for Interactive Lessons Feature

**\*\*Project Name:\*\*** Online Learning Platform

**\*\*Feature Name:\*\*** Interactive Lessons

**\*\*User Story Title:\*\*** As a learner, I want interactive lessons so that I can engage with the content more effectively.

**\*\*Acceptance Criteria:\*\***

1. Lessons include quizzes, polls, simulations, and discussions.
2. Interactive elements are integrated seamlessly into the lesson content.
3. Learners can track their progress through these activities.

4. Feedback is provided after each interactive component.

5. Interactive lessons can be accessed on various devices.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-IL-01

**\*\*Test Case Title:\*\*** Verify that lessons include quizzes, polls, simulations, and discussions.

**\*\*Test Case Description:\*\*** This test case verifies the presence of various interactive elements within a lesson.

**\*\*Test Suite:\*\*** Interactive Lessons

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A user is logged into the platform.

\* A course with interactive lessons is selected.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to a selected lesson within a course.

2. Examine the lesson content for the presence of quizzes.

3. Examine the lesson content for the presence of polls.

4. Examine the lesson content for the presence of simulations.

5. Examine the lesson content for the presence of discussions.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The lesson should contain quizzes, polls, simulations, and discussion forums.

**\*\*Severity:\*\*** Blocker

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive



**\*\*Test Case ID:\*\* TC-IL-02**

**\*\*Test Case Title:\*\* Verify that interactive elements are integrated seamlessly into the lesson content.**

**\*\*Test Case Description:\*\* This test case verifies that interactive elements are seamlessly integrated with the lesson flow and content.**

**\*\*Test Suite:\*\* Interactive Lessons**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A user is logged into the platform.

\* A course with interactive lessons is selected.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to a selected lesson within a course.
2. Attempt to participate in a quiz, poll, simulation, or discussion.
3. Observe the flow of the lesson content and the interaction with the chosen interactive element.
4. Verify that the interactive element is presented at a relevant point in the lesson.
5. Verify that the transition back to the lesson content after completing the interactive element is smooth.

**\*\*Postconditions:\*\* None**

**\*\*Expected Result:\*\* The interactive elements should integrate seamlessly with the lesson content, maintaining a smooth learning experience.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Usability Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-IL-03**

**\*\*Test Case Title:\*\* Verify that learners can track their progress through interactive activities.**

**\*\*Test Case Description:\*\*** This test case verifies that learners can track their progress through interactive activities within a lesson.

**\*\*Test Suite:\*\*** Interactive Lessons

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A user is logged into the platform.

\* A course with interactive lessons is selected.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to a selected lesson within a course.
2. Attempt to participate in a quiz, poll, simulation, or discussion.
3. Observe the progress tracking mechanism (e.g., completion status, score, progress bar).
4. Verify that the tracking mechanism accurately reflects the user's progress in the chosen interactive activity.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** Learners should be able to track their progress through interactive activities within a lesson.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-IL-04

**\*\*Test Case Title:\*\*** Verify that feedback is provided after each interactive component.

**\*\*Test Case Description:\*\*** This test case verifies that feedback is provided to learners after completing each interactive element.

**\*\*Test Suite:\*\*** Interactive Lessons

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* A user is logged into the platform.
- \* A course with interactive lessons is selected.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to a selected lesson within a course.
2. Attempt to participate in a quiz, poll, simulation, or discussion.
3. Upon completion of the interactive element, observe for feedback or results.
4. Verify that the feedback is provided in a clear and understandable manner.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** Learners should receive feedback after completing each interactive component, such as correct/incorrect answers, results, or insights.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-IL-05

**\*\*Test Case Title:\*\*** Verify that interactive lessons can be accessed on various devices.

**\*\*Test Case Description:\*\*** This test case verifies that interactive lessons are accessible and function correctly across different devices (desktop, mobile, tablet).

**\*\*Test Suite:\*\*** Interactive Lessons

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* A user is logged into the platform.
- \* A course with interactive lessons is selected.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access the interactive lesson on a desktop browser.
2. Access the interactive lesson on a mobile browser.
3. Access the interactive lesson on a tablet browser.
4. Verify that all interactive elements (quizzes, polls, simulations, discussions) are accessible and function correctly on each device.
5. Verify that the layout and user interface are responsive and optimized for each device type.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** Interactive lessons should be accessible and function correctly on various devices with responsive design.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Compatibility Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 51**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Personalized Learning Paths

**Feature description -**

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

**Userstory title -**

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

**Acceptance criteria -**

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

**Generated Test Cases:**

## Test Cases for Personalized Learning Paths Feature

**\*\*Project:\*\*** Online Learning Platform

**\*\*Feature:\*\*** Personalized Learning Paths

**\*\*User Story:\*\*** As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

**\*\*Acceptance Criteria:\*\***

1. Learners can choose courses and modules based on their interests.
2. Personalized learning paths adjust dynamically based on learner progress.
3. Learners receive recommendations for new courses and modules.

- 4. Progress tracking is personalized to each learner's path.
- 5. Learners can adjust their learning paths at any time.

**\*\*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-PLP-01	Verify that learners can choose courses and modules based on their interests.	This test case verifies that learners can browse and select courses and modules based on their interests and preferences. The test checks if the system provides filtering options and recommendations aligned with the learner's selected interests.	Functional Testing	High	User has an account on the learning platform	No test data needed	1. Log in as a learner. 2. Navigate to the "Explore Courses" section. 3. Use the filtering options to select courses based on subject, level, and other relevant criteria. 4. View the list of courses and modules matching the selected interests. 5. Select a course and view its details, including the modules. 6. Choose specific modules within the selected course.	The system should display courses and modules relevant to the selected interests.	Minor	High	Functional Testing	Positive	
TC-PLP-02	Verify that personalized learning paths adjust dynamically based on learner progress.	This test case assesses if the system dynamically updates the personalized learning path based on the learner's progress through courses and modules. It verifies if the system recommends relevant content and adapts to the learner's evolving learning journey.	Functional Testing	High	User has an account on the learning platform and has started a personalized learning path	No test data needed	1. Log in as a learner with an active personalized learning path. 2. Start a course and complete a module within the path. 3. View the updated personalized learning path. 4. Observe if the system recommends new courses or modules based on the completed content.	The system should update the personalized learning path based on completed courses and modules. The system should also recommend relevant courses and modules.	Major	High	Functional Testing	Positive	

| TC-PLP-03 | Verify that learners receive recommendations for new courses and modules. | This test case verifies that the system provides relevant recommendations for new courses and modules based on the learner's interests, progress, and learning history. It checks if the recommendations align with the learner's learning journey and offer valuable learning opportunities. | Functional Testing | Medium | User has an account on the learning platform and has started a personalized learning path | No test data needed | 1. Log in as a learner with an active personalized learning path. <br> 2. View the "Recommended Courses" or "Suggested Modules" section. <br> 3. Check if the recommendations align with the learner's interests, completed courses, and learning objectives. | The system should provide personalized recommendations for courses and modules based on the learner's profile and progress. | Minor | Medium | Functional Testing | Positive |

| TC-PLP-04 | Verify that progress tracking is personalized to each learner's path. | This test case checks if the progress tracking system accurately reflects the learner's advancement through their personalized learning path. The test ensures that the system displays the learner's completion status for each course and module within their chosen path. | Functional Testing | High | User has an account on the learning platform and has started a personalized learning path | No test data needed | 1. Log in as a learner with an active personalized learning path. <br> 2. Access the "My Progress" section. <br> 3. Verify that the progress tracking displays the learner's completion status for each course and module in their chosen learning path. | The progress tracking should accurately reflect the learner's progress through their personalized learning path. | Major | High | Functional Testing | Positive |

| TC-PLP-05 | Verify that learners can adjust their learning paths at any time. | This test case assesses if the system allows learners to modify their personalized learning path as needed. It checks if the system provides options for adding, removing, or changing courses and modules within the learner's path. | Functional Testing | Medium | User has an account on the learning platform and has started a personalized learning path | No test data needed | 1. Log in as a learner with an active personalized learning path. <br> 2. Navigate to the "My Learning Path" section. <br> 3. Attempt to add a new course or module to the path. <br> 4. Attempt to remove a course or module from the path. <br> 5. Attempt to replace a course or module with a different one. | The system should allow learners to add, remove, or change courses and modules within their personalized

learning path. | Minor | Medium | Functional Testing | Positive |

**\*\*Note:\*\*** These test cases are just examples and can be further expanded to cover edge cases and negative scenarios, ensuring comprehensive testing of the personalized learning paths feature.

-----

**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 Progress Tracking Feature

\*\*Project Name:\*\* Online Learning Platform

\*\*User Story:\*\* As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

\*\*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-01	Verify that progress tracking shows completed activities and grades	This test case checks if the progress tracking feature displays all the completed activities and their corresponding grades for a learner.	Progress Tracking	High	* Learner is enrolled in a course with completed modules and activities. * Learner has achieved grades for completed activities.	No test data needed	1. Log in as a learner. 2. Navigate to the course page. 3. Access the progress tracking feature. 4. Verify that all completed activities and their grades are displayed.	No postconditions	All completed activities and their grades are displayed accurately in the progress tracking section.	Major	High	Functional Testing	Positive
TC-02	Verify that learners can view detailed reports on their progress	This test case checks if learners can access detailed reports about their progress in the course.	Progress Tracking	High	* Learner is enrolled in a course with completed modules and activities. * Learner has achieved grades for completed activities.	No test data needed	1. Log in as a learner. 2. Navigate to the course page. 3. Access the progress tracking feature. 4. Click on the "View report" option. 5. Verify that a detailed report containing progress details like completed activities, grades, and time spent is displayed.	Report is displayed successfully.	Major	High	Functional Testing	Positive	

| TC-03 | Verify that achievements are highlighted in the progress tracker | This test case validates if achievements unlocked by a learner are highlighted in the progress tracking section. | Progress Tracking | High | \* Learner is enrolled in a course with achievements associated with modules or activities. \* Learner has completed modules and activities leading to unlocking achievements. | No test data needed | 1. Log in as a learner. 2. Navigate to the course page. 3. Access the progress tracking feature. 4. Verify that all unlocked achievements are prominently displayed with appropriate icons or visual cues. | Achievements are displayed clearly in the progress tracking section. | Major | High | Functional Testing | Positive |

| TC-04 | Verify that learners receive notifications about their progress | This test case checks if learners receive notifications upon completing activities or achieving milestones in a course. | Progress Tracking | High | \* Learner is enrolled in a course. \* Learner has completed an activity or unlocked an achievement. | No test data needed | 1. Log in as a learner. 2. Complete an activity or achieve a milestone in the course. 3. Verify that a notification is sent to the learner about the progress made. | Notification is received by the learner. | Major | High | Functional Testing | Positive |

| TC-05 | Verify that progress tracking can be exported as reports | This test case verifies if learners can export their progress data as a downloadable report. | Progress Tracking | Medium | \* Learner is enrolled in a course with completed activities and achievements. | No test data needed | 1. Log in as a learner. 2. Navigate to the course page. 3. Access the progress tracking feature. 4. Click on the "Export report" option. 5. Select the desired report format (e.g., PDF, CSV). 6. Download the report. 7. Verify that the downloaded report contains the correct progress data. | Report is exported successfully. | Minor | Medium | Functional Testing | Positive |

**\*\*Note:\*\*** These test cases are designed to provide a starting point. Additional test cases might be required based on the specific requirements and functionality of the online learning platform.

-----

**Data - 53**

**Project name -**

## Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Assessment and Grading

### **Feature description -**

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

### **Userstory title -**

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

### **Acceptance criteria -**

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

### **Generated Test Cases:**

## Test Cases for Online Learning Platform - Assessment and Grading Feature

**\*\*Project Context:\*\*** Online Learning Platform

**\*\*User Story:\*\*** As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

**\*\*Acceptance Criteria:\*\***

1. Instructors can create quizzes, tests, and assignments.
2. Assessments can include various question types (MCQs, short answers, essays, etc.).
3. Automatic grading is available for objective questions.
4. Manual grading is available for subjective questions.
5. Feedback is provided to learners after grading.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-01

**\*\*Test Case Title:\*\*** Verify that instructors can create quizzes, tests, and assignments.

**\*\*Test Case Description:\*\*** This test case verifies that instructors have the ability to create different types of assessments within the platform, ensuring flexibility in their evaluation methods.

**\*\*Test Suite:\*\*** Assessment Creation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* An instructor account is logged in.

\* A course is created.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the course page.
2. Select the "Assessments" tab.
3. Click on "Create Assessment".
4. Choose the assessment type: Quiz, Test, or Assignment.
5. Verify that the chosen assessment type is created successfully.

**\*\*Postconditions:\*\*** The created assessment should be visible in the assessment list.

**\*\*Expected Result:\*\*** The platform allows instructors to create quizzes, tests, and assignments successfully.

**\*\*Severity:\*\*** Blocker

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-02

**\*\*Test Case Title:\*\*** Verify that assessments can include various question types (MCQs, short answers, essays, etc.).

**\*\*Test Case Description:\*\*** This test case checks if the platform supports a range of question types for assessments, catering to diverse assessment needs and cognitive levels.

**\*\*Test Suite:\*\*** Question Types

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* An instructor account is logged in.

\* An assessment (Quiz, Test, or Assignment) is created.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the created assessment.
2. Select "Add Question".
3. Choose from available question types: MCQ, Short Answer, Essay, etc.
4. Add a question of the chosen type.
5. Verify that the question is added successfully and the chosen question type is displayed correctly.

**\*\*Postconditions:\*\*** The added question should be visible in the assessment.

**\*\*Expected Result:\*\*** The platform supports various question types for assessments.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-03**

**\*\*Test Case Title:\*\* Verify automatic grading is available for objective questions.**

**\*\*Test Case Description:\*\* This test case ensures that objective questions, such as MCQs, are automatically graded by the platform, providing immediate feedback to learners.**

**\*\*Test Suite:\*\* Automatic Grading**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* An instructor account is logged in.

\* An assessment with MCQ questions is created.

**\*\*Test Data:\*\* MCQ question with a correct answer.**

**\*\*Test Steps:\*\***

1. As a learner, attempt the MCQ question.
2. Submit the answer.
3. Verify that the question is automatically graded and the score is displayed immediately.

**\*\*Postconditions:\*\* The learner's score for the question should be recorded.**

**\*\*Expected Result:\*\* The platform automatically grades objective questions, providing immediate feedback.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-04**

**\*\*Test Case Title:\*\* Verify manual grading is available for subjective questions.**

**\*\*Test Case Description:\*\*** This test case validates that the platform allows instructors to manually grade subjective questions like short answers and essays, enabling more nuanced evaluation and feedback.

**\*\*Test Suite:\*\*** Manual Grading

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* An instructor account is logged in.

\* An assessment with short answer questions is created.

**\*\*Test Data:\*\*** Short answer question with a learner response.

**\*\*Test Steps:\*\***

1. As an instructor, view the submitted short answer question.

2. Enter a grade and feedback for the question.

3. Submit the grade and feedback.

4. Verify that the question is graded and feedback is saved.

**\*\*Postconditions:\*\*** The learner should receive the grade and feedback.

**\*\*Expected Result:\*\*** The platform allows instructors to manually grade subjective questions and provide feedback.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-05

**\*\*Test Case Title:\*\*** Verify feedback is provided to learners after grading.

**\*\*Test Case Description:\*\*** This test case checks if learners receive feedback after their assessments are graded, promoting learning and improvement through constructive feedback.

**\*\*Test Suite:\*\*** Feedback Delivery

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* An instructor account is logged in.
- \* A learner account is logged in.
- \* An assessment with graded questions is completed.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. As a learner, view the graded assessment.
2. Verify that feedback is provided for each question.
3. Check that the overall score and grade are displayed.

**\*\*Postconditions:\*\*** The learner should have access to the feedback and their final score.

**\*\*Expected Result:\*\*** The platform provides feedback to learners after grading their assessments.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 54**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Discussion Forums



## **Feature description -**

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge sharing, enriching the learning experience through diverse perspectives and active participation.

## **Userstory title -**

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

## **Acceptance criteria -**

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all devices.']

## **Generated Test Cases:**

## Test Cases for Online Learning Platform Discussion Forums

**Project:** Online Learning Platform

**Feature:** Discussion Forums

**User Story:** As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

**Test Case Creation Template:**

**Test Case ID:** TC\_DF\_01

**Test Case Title:** Verify that learners can post questions and responses in forums.

**Test Case Description:** This test case validates the functionality for learners to create new posts in forums, including adding text and any relevant attachments. It also ensures they can reply to existing posts, creating threaded discussions.

**Test Suite:** Discussion Forums

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A learner account is created and logged in.

\* A forum exists within the platform.

**\*\*Test Data:\*\***

\* Sample question text: "What are the best resources for learning about [topic]?"

\* Sample response text: "I found [link] to be helpful!"

**\*\*Test Steps:\*\***

1. Navigate to the forum.
2. Click on the "New Post" button.
3. Enter the sample question text into the post editor.
4. (Optional) Attach a relevant file or image.
5. Click on the "Post" button.
6. Verify the post appears in the forum.
7. Reply to the post with the sample response text.
8. Click on the "Reply" button.
9. Verify the reply appears below the original post.

**\*\*Postconditions:\*\***

\* The new post and reply are visible to other learners.

**\*\*Expected Result:\*\*** The learner successfully posts a question and responds to an existing post, creating a threaded discussion.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_DF\_02**

**\*\*Test Case Title:\*\* Verify that forums support threaded discussions.**

**\*\*Test Case Description:\*\* This test case verifies that forum posts can be nested as replies, creating a hierarchical structure for discussions. It ensures that users can easily follow the thread of a discussion and understand the context of each reply.**

**\*\*Test Suite:\*\* Discussion Forums**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A learner account is created and logged in.

\* A forum exists with at least one post and one reply.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the forum.
2. Locate a post with at least one reply.
3. Click on the "Reply" button under a reply to the original post.
4. Enter text in the reply editor.
5. Click on the "Reply" button.
6. Verify that the new reply is nested under the reply it was a response to.
7. Observe the indentation or visual cues indicating the hierarchical structure of the discussion.

**\*\*Postconditions:\*\***

\* The new reply is correctly nested under the parent reply.

**\*\*Expected Result:\*\* The forum displays threaded discussions, with replies nested under their corresponding parent posts.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_DF\_03**

**\*\*Test Case Title:\*\* Verify that instructors can moderate discussions.**

**\*\*Test Case Description:\*\* This test case verifies that instructors have the necessary permissions to moderate forum discussions, including deleting inappropriate posts, editing posts, and marking posts as resolved.**

**\*\*Test Suite:\*\* Discussion Forums**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* An instructor account is created and logged in.

\* A forum exists with posts and replies from learners.

**\*\*Test Data:\*\***

\* An inappropriate post containing offensive language.

\* A post marked as resolved.

**\*\*Test Steps:\*\***

1. Log in as the instructor.
2. Navigate to the forum.
3. Locate an inappropriate post.
4. Click on the "Moderate" option (or similar) for the post.
5. Select "Delete" (or similar) from the moderation options.
6. Verify the post is removed from the forum.
7. Locate a post requiring resolution.
8. Click on the "Moderate" option (or similar) for the post.
9. Select "Mark as Resolved" (or similar) from the moderation options.
10. Verify the post is marked as resolved with a visual indicator.

**\*\*Postconditions:\*\***

\* Inappropriate posts are removed from the forum.

\* Posts requiring resolution are marked as resolved.

**\*\*Expected Result:\*\*** Instructors can successfully moderate discussions by deleting inappropriate posts, editing posts, and marking posts as resolved.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_DF\_04

**\*\*Test Case Title:\*\*** Verify that learners receive notifications for new posts.

**\*\*Test Case Description:\*\*** This test case validates the notification system for new posts in forums. It ensures that learners are notified when new posts or replies are added to forums they are subscribed to.

**\*\*Test Suite:\*\*** Discussion Forums

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* A learner account is created and logged in.

\* The learner is subscribed to a forum.

\* Another learner posts a new message in the forum.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Log in as the learner.
2. Navigate to the forum the learner is subscribed to.
3. Verify that no new posts are visible.
4. Log out of the learner account.

5. Log in as a different learner account.
6. Post a new message in the forum the first learner is subscribed to.
7. Log out of the second learner account.
8. Log back in as the first learner account.
9. Check the notification center or any other designated area for notifications.
10. Verify that a notification is received indicating a new post in the subscribed forum.

**\*\*Postconditions:\*\***

- \* The notification center displays the new post notification.
- \* The learner can access the new post from the notification.

**\*\*Expected Result:\*\*** Learners receive notifications for new posts in forums they are subscribed to.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_DF\_05

**\*\*Test Case Title:\*\*** Verify that forums are accessible on all devices.

**\*\*Test Case Description:\*\*** This test case verifies the accessibility of forums across various devices, including desktops, laptops, tablets, and smartphones. It ensures that the user interface adapts to different screen sizes and resolutions, providing a seamless experience on all platforms.

**\*\*Test Suite:\*\*** Discussion Forums

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- \* The online learning platform is launched on different devices (desktop, laptop, tablet, smartphone).
- \* A learner account is created and logged in on each device.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Access the online learning platform on each device.
2. Navigate to the discussion forums section.
3. Verify that the forums are displayed correctly on each device, without any layout or functionality issues.
4. Check if the font size, spacing, and button placement are appropriate for each screen size.
5. Attempt to post a message, reply to a post, and navigate through the forum threads on each device.

**\*\*Postconditions:\*\***

\* The discussion forums are accessible and functional on all devices.

**\*\*Expected Result:\*\*** Users can seamlessly interact with the discussion forums on all supported devices, regardless of screen size or platform.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Compatibility Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Note:\*\*** These test cases provide a starting point. You can expand upon these and create more specific test cases depending on the complexity of your discussion forum implementation.

-----

**Data - 55**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Live Classes and Webinars

**Feature description -**

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

**Userstory title -**

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

**Acceptance criteria -**

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

**Generated Test Cases:**

## Test Cases for Live Classes and Webinars Feature

**\*\*Project:\*\*** Online Learning Platform

**\*\*Feature:\*\*** Live Classes and Webinars

**\*\*User Story:\*\*** As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

**\*\*Acceptance Criteria:\*\***



1. Live classes and webinars are conducted through video conferencing tools.
2. Learners can ask questions in real-time.
3. Sessions are recorded and available for later viewing.
4. Instructors can share screen and resources during sessions.
5. Notifications are sent before sessions start.

**\*\*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-001	Verify that live classes and webinars are conducted through video conferencing tools	This test case verifies that the platform integrates with a reliable video conferencing tool and enables instructors to conduct live sessions with learners.	Live Classes & Webinars	High	- User is logged into the platform.	- A live class is scheduled.	1. Navigate to the live class page. 2. Verify that the video conferencing tool is embedded within the platform. 3. Check if the tool functions correctly and allows for video/audio streaming, chat, and screen sharing.	No postconditions	The video conferencing tool is integrated within the platform and functions correctly, enabling live classes and webinars.	Blocker	High	Functional Testing, Integration Testing	Positive
TC-002	Verify that learners can ask questions in real-time	This test case ensures that learners can interact with the instructor during live sessions by asking questions in real-time through the video conferencing tool.	Live Classes & Webinars	High	- User is logged into the platform.	- A live class is in progress.	1. Join the live class using the video conferencing tool. 2. Type a question in the chat box of the video conferencing tool. 3. Verify that the question is displayed for the instructor and other learners to see.	No postconditions	The question is successfully sent to the instructor and other learners in the live session.	Major	High	Functional Testing, Integration Testing	Positive
TC-003	Verify that sessions are recorded and available for later viewing	This test case validates that live											

sessions are recorded and available for playback after the live session ends, allowing learners to review the content. | Live Classes & Webinars | Medium | - User is logged into the platform. <br> - A live class has been completed. | No test data needed | 1. Access the recorded session from the live class page or within the course materials. <br> 2. Verify that the recording is playable. <br> 3. Ensure that the recording quality is good and the content is clearly visible. | No postconditions | The recorded session is available for playback with good audio and video quality. | Minor | Medium | Functional Testing, Integration Testing | Positive |

| TC-004 | Verify that instructors can share screen and resources during sessions | This test case ensures that instructors can utilize the video conferencing tool to share their screen and upload/share resources during live sessions. | Live Classes & Webinars | High | - User is logged into the platform as an instructor. <br> - A live class is in progress. | - A presentation file (PDF, PPT). <br> - A document file (Word, Excel). | 1. Share screen from the instructor's device. <br> 2. Upload a presentation file and share it during the session. <br> 3. Upload a document file and share it during the session. | No postconditions | The instructor is able to successfully share their screen and upload/share relevant files during the live session. | Major | High | Functional Testing, Integration Testing | Positive |

| TC-005 | Verify that notifications are sent before sessions start | This test case ensures that learners receive timely notifications about upcoming live sessions. | Live Classes & Webinars | High | - User is logged into the platform. <br> - A live class is scheduled. | No test data needed | 1. Enroll in the live class. <br> 2. Verify that a notification is received before the scheduled start time of the class. <br> 3. Check the notification content, including class details and a link to join the session. | No postconditions | The learner receives a notification with accurate information about the upcoming live session. | Major | High | System Testing, Non-Functional Testing | 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

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Mobile Accessibility**

**\*\*User Story Title:\*\* As a learner, I want mobile access to the platform so that I can learn on the go.**

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC-MA-01**

**\*\*Test Case Title:\*\* Verify that the platform is accessible on smartphones and tablets.**

**\*\*Test Case Description:\*\* This test case verifies that the platform's website is accessible and functional on various smartphone and tablet devices with different screen sizes and operating systems.**

**\*\*Test Suite:\*\* Mobile Accessibility**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A smartphone (e.g., iPhone, Android) and a tablet (e.g., iPad, Android tablet) are available.

\* The platform is launched on the chosen devices.

\* Internet connection is active.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the online learning platform website using the smartphone's web browser.

2. Navigate through different sections of the platform, including the homepage, courses, lectures, tutorials, quizzes, and user profile.

3. Repeat the same steps using the tablet device.

**\*\*Postconditions:\*\***

\* No errors or warnings are displayed on the screen.

\* All elements on the platform load correctly and are responsive to touch interactions.

\* All functionalities work as expected on both smartphone and tablet devices.

**\*\*Expected Result:\*\* The platform should be accessible, functional, and responsive on both smartphone and tablet devices.**

**\*\*Severity:\*\* Blocker**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Compatibility Testing, Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-MA-02**

**\*\*Test Case Title:\*\* Verify that the mobile UI is optimized for touch navigation.**

**\*\*Test Case Description:\*\* This test case verifies that the platform's UI elements, such as buttons, links, and menus, are easily clickable and navigable using touch gestures on mobile devices.**

**\*\*Test Suite:\*\* Mobile Accessibility**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A smartphone or tablet is available.

\* The platform is launched on the chosen device.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the online learning platform website.
2. Tap on various buttons, links, and menu items on the platform.
3. Attempt to scroll through pages using touch gestures.
4. Try navigating through different sections of the platform using touch interactions.

**\*\*Postconditions:\*\***

\* All UI elements are responsive to touch gestures.

\* Scrolling and navigation through the platform are smooth and intuitive.

\* There are no issues with accidental clicks or unintended interactions.

**\*\*Expected Result:\*\* The platform's UI should be optimized for touch navigation, offering a smooth and responsive user experience.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Usability Testing, Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-MA-03**

**\*\*Test Case Title:\*\*** Verify that course materials are accessible and readable on mobile devices.

**\*\*Test Case Description:\*\*** This test case verifies that course materials, including text content, images, and videos, are displayed correctly and are easily readable on mobile devices.

**\*\*Test Suite:\*\*** Mobile Accessibility

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A smartphone or tablet is available.

\* The platform is launched on the chosen device.

\* A course is selected for testing.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Open a course on the platform.

2. Access different learning materials within the course, including text content, images, videos, and downloadable resources.

3. Verify that the content is displayed correctly, without any overlaps, distortion, or errors.

4. Check if the text font size and contrast are suitable for mobile devices.

5. Attempt to zoom in and out on the content to check for readability.

**\*\*Postconditions:\*\***

\* Course materials are displayed clearly and are readable on mobile devices.

\* The layout and formatting of the content are appropriate for mobile screens.

\* Content is easily scrollable and navigable.

**\*\*Expected Result:\*\*** Course materials should be fully accessible and readable on mobile devices, offering a comfortable and engaging learning experience.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Compatibility Testing, Usability Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\* TC-MA-04**

**\*\*Test Case Title:\*\* Verify that notifications are sent to mobile devices.**

**\*\*Test Case Description:\*\* This test case verifies that the platform sends notifications to mobile devices to inform learners about important events, updates, and activities within the platform.**

**\*\*Test Suite:\*\* Mobile Accessibility**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* The user is logged into the platform on a mobile device.

\* The platform's notification settings are enabled.

\* There are available notifications within the platform (e.g., new course updates, message notifications).

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Trigger an event that should generate a notification on the platform (e.g., new course release, message from another user).
2. Check if the notification is received on the mobile device.
3. Verify that the notification content is relevant and accurate.
4. Try accessing the notification content on the mobile device.

**\*\*Postconditions:\*\***

\* The notification is successfully sent and received on the mobile device.

\* The notification content is displayed correctly and is interactive.

\* The notification should not cause any issues or errors on the device.

**\*\*Expected Result:\*\* The platform should successfully send notifications to mobile devices, informing users about relevant events and activities.**

**\*\*Severity:\*\* Minor**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing, System Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-MA-05**

**\*\*Test Case Title:\*\* Verify that the platform supports offline access to downloaded materials.**

**\*\*Test Case Description:\*\* This test case verifies that learners can access downloaded course materials, such as lectures, tutorials, and documents, even when an internet connection is not available.**

**\*\*Test Suite:\*\* Mobile Accessibility**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- \* A smartphone or tablet is available.
- \* The platform is launched on the chosen device.
- \* The user has downloaded course materials to their device.
- \* The internet connection is turned off.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Download course materials (e.g., video lectures, documents) within a course.
2. Turn off the internet connection on the device.
3. Attempt to access the downloaded materials.
4. Verify that the materials can be accessed and played/viewed without an internet connection.

**\*\*Postconditions:\*\***

- \* The downloaded materials are accessible offline.
- \* The content plays or opens smoothly without any errors or buffering issues.

**\*\*Expected Result:\*\* The platform should provide offline access to downloaded course materials, allowing learners to continue their learning even without an internet connection.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing, System Testing**



**Data - 57**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Resource Library

**Feature description -**

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

**Userstory title -**

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

**Acceptance criteria -**

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

**Generated Test Cases:**

## ## Test Cases for Resource Library Feature

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Resource Library**

**\*\*User Story Title:\*\*** As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

**\*\*Acceptance Criteria:\*\***

1. Resource library includes e-books, articles, and reference guides.
2. Learners can search for specific materials.
3. Resources are categorized and tagged for easy navigation.
4. Learners can download resources for offline access.
5. Instructors can recommend resources to learners.

### \*\*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-RL-01	Verify that the Resource Library includes e-books, articles, and reference guides	This test case verifies that the resource library contains different types of materials, such as e-books, articles, and reference guides.	Resource Library	High	User is logged in to the platform	No test data needed
1. Navigate to the Resource Library section. 2. Check if the library displays different types of resources, including e-books, articles, and reference guides. 3. Verify the presence of each type of resource by clicking on an example of each.						
User returns to the resource library.						
The Resource Library displays a variety of resources, including e-books, articles, and reference guides.						
Major   High   Functional Testing   Positive						
TC-RL-02	Verify that learners can search for specific materials	This test case verifies that learners can				

search for specific materials within the resource library. | Resource Library | High | User is logged in to the platform | Search term "Machine Learning" | 1. Navigate to the Resource Library section. 2. Enter the search term "Machine Learning" in the search bar. 3. Click on the search button. | Search results are displayed. | The platform displays relevant search results based on the entered search term, including e-books, articles, and reference guides related to "Machine Learning". | Major | High | Functional Testing | Positive |

| TC-RL-03 | Verify that resources are categorized and tagged for easy navigation | This test case verifies that resources are categorized and tagged to allow learners to easily navigate the library. | Resource Library | High | User is logged in to the platform | No test data needed | 1. Navigate to the Resource Library section. 2. Verify that resources are grouped into different categories (e.g., "Computer Science", "Mathematics", "History"). 3. Check if each resource is tagged with relevant keywords. | User navigates through different categories and resource listings. | Resources are categorized and tagged to facilitate easy navigation and browsing by learners. | Major | High | Functional Testing | Positive |

| TC-RL-04 | Verify that learners can download resources for offline access | This test case verifies that learners can download resources from the library for offline access. | Resource Library | High | User is logged in to the platform | Selected resource (e.g., PDF e-book) | 1. Navigate to the Resource Library section. 2. Select an e-book or other downloadable resource. 3. Click on the "Download" option for the chosen resource. | Download completes and the resource is saved in the user's device. | The selected resource is downloaded successfully and can be accessed offline. | Major | High | Functional Testing | Positive |

| TC-RL-05 | Verify that instructors can recommend resources to learners | This test case verifies that instructors can recommend resources from the library to their learners. | Resource Library | High | Instructor is logged in to the platform | Selected resource, course, and student | 1. Login as an instructor. 2. Navigate to a course where recommendations are to be made. 3. Select a resource from the library. 4. Select a student from the course roster. 5. Click on "Recommend resource" to suggest the selected resource to the student. | Recommendation is displayed to the student. | The selected resource is recommended to the chosen student within the specified course. | Major | High | Functional Testing | 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 -**

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:**

## Test Cases for Peer Review Feature

**\*\*Project:\*\*** Online Learning Platform

**\*\*User Story:\*\*** As a learner, I want to participate in peer review so that I can give and receive feedback on

assignments.

**\*\*Acceptance Criteria:\*\***

1. Learners can submit assignments for peer review.
2. Peers can provide feedback on submitted assignments.
3. A structured review process guides the feedback.
4. Learners can view feedback on their work.
5. Reviewers remain anonymous to ensure unbiased feedback.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\*** TC\_PeerReview\_[Acceptance Criteria Number]

**\*\*Test Case Title:\*\*** Verify that... [Description of functionality being tested]

**\*\*Test Case Description:\*\*** A brief description of the test case, between 70-100 words.

**\*\*Test Suite:\*\*** Peer Review Functionality

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A learner account is created and enrolled in a course with an assignment requiring peer review.

\* The assignment is submitted by the learner.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\*** [List steps for executing the test case, min 3 to max 6]

**\*\*Postconditions:\*\*** [Any follow-up actions after test execution, max 3 items]

**\*\*Expected Result:\*\*** [Overall expected outcome of the test case]

**\*\*Severity:\*\*** [Blocker, Critical, Major, Minor, Trivial]

**\*\*Priority:\*\*** [High, Medium, Low]

**\*\*Type of Testing:\*\*** [Functional Testing]

**\*\*Test Case Approach:\*\*** [Positive/Negative]

**\*\*Test Cases:\*\***

**\*\*1. TC\_PeerReview\_1\*\***

**\*\*Test Case Title:\*\*** Verify that learners can submit assignments for peer review.

**\*\*Test Case Description:\*\*** This test case checks if learners can submit their assignments for peer review after completing them.

**\*\*Test Steps:\*\***

1. Login as a learner.
2. Navigate to the course containing the assignment.
3. Open the assignment and click on the "Submit" button.
4. Verify that the assignment is successfully submitted for peer review.

**\*\*Expected Result:\*\*** The assignment should be successfully submitted for peer review, and the learner should see a confirmation message.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*2. TC\_PeerReview\_2\*\***

**\*\*Test Case Title:\*\*** Verify that peers can provide feedback on submitted assignments.

**\*\*Test Case Description:\*\*** This test case validates if peers can provide feedback on the submitted assignments, including text comments and ratings.

**\*\*Test Steps:\*\***

1. Login as a learner who has access to the submitted assignment for peer review.
2. Open the assignment and access the "Review" section.
3. Provide feedback in the form of text comments and ratings.

4. Submit the feedback.

**\*\*Expected Result:\*\*** The feedback should be successfully submitted, and the learner should see a confirmation message.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*3. TC\_PeerReview\_3\*\***

**\*\*Test Case Title:\*\*** Verify that a structured review process guides the feedback.

**\*\*Test Case Description:\*\*** This test case ensures that the feedback process is structured with specific guidelines and prompts to encourage constructive and helpful feedback.

**\*\*Test Steps:\*\***

1. Login as a learner who has access to the submitted assignment for peer review.
2. Open the assignment and access the "Review" section.
3. Verify that the platform provides specific prompts or questions to guide feedback on different aspects of the assignment.
4. Check if the feedback form allows for rating and text comments based on a clear rubric or guidelines.

**\*\*Expected Result:\*\*** The feedback process should be structured with prompts and guidelines to ensure constructive feedback.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*4. TC\_PeerReview\_4\*\***

**\*\*Test Case Title:\*\*** Verify that learners can view feedback on their work.

**\*\*Test Case Description:\*\*** This test case checks if learners can access and view the feedback provided by their peers on their submitted assignments.

**\*\*Test Steps:\*\***

1. Login as a learner who has submitted an assignment for peer review.

2. Navigate to the submitted assignment.

3. Access the "Feedback" section.

4. Verify that the feedback provided by peers is visible to the learner.

**\*\*Expected Result:\*\*** The learner should be able to see the feedback from their peers, including comments and ratings.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*5. TC\_PeerReview\_5\*\***

**\*\*Test Case Title:\*\*** Verify that reviewers remain anonymous to ensure unbiased feedback.

**\*\*Test Case Description:\*\*** This test case ensures that reviewers remain anonymous to prevent bias and promote honest feedback.

**\*\*Test Steps:\*\***

1. Login as a learner who has submitted an assignment for peer review.

2. Navigate to the submitted assignment and access the "Feedback" section.

3. Verify that the reviewer's identity is not displayed, only their feedback.

4. Repeat the steps as a reviewer providing feedback to confirm the anonymity.

**\*\*Expected Result:\*\*** The reviewer's identity should be hidden to ensure unbiased feedback.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

-----

**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:**

## Test Cases for Certificates and Badges Feature

**\*\*Project:\*\*** Online Learning Platform

**\*\*User Story:\*\*** As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

**\*\*Acceptance Criteria:\*\***

1. Learners earn certificates upon course completion.
2. Badges are awarded for specific achievements and milestones.
3. Certificates and badges are accessible through learner profiles.
4. Learners can share their certificates and badges on social media.
5. Instructors can issue custom badges for special achievements.

**\*\*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-001	Verify that learners earn certificates upon course completion	This test case verifies that learners receive certificates upon completing a course successfully. The test case checks for certificate availability, content, and accuracy of details.	Certificates & Badges	High	* Learner is enrolled in a course. * Learner has completed all course requirements.	Course ID, Learner ID	1. Complete all course modules and assessments. 2. Navigate to the course dashboard. 3. Verify the availability of a certificate for the completed course. 4. Download the certificate. 5. Verify the certificate content including learner's name, course name, completion date.	Certificate downloaded successfully.	Learner receives a certificate upon successful course completion with accurate details.	Major	High	Functional Testing	Positive
TC-002	Verify that badges are awarded for specific achievements and milestones	This test case verifies that learners receive badges for specific achievements and milestones, such as completing a specific module, scoring a high grade, or contributing to a discussion forum.	Certificates & Badges	Medium	* Learner is enrolled in a course. * Learner has achieved a specific milestone.	Course ID, Learner ID, Milestone ID	1. Complete a specific milestone within a course (e.g., complete a module, score a high grade in an assessment). 2. Navigate to the learner's profile. 3. Verify the presence of the corresponding badge on the profile.	Badge displayed on the learner profile.	Badge is awarded for the specific achievement or milestone.	Minor	Medium	Functional Testing	Positive

Medium | Functional Testing | Positive |

| TC-003 | Verify that certificates and badges are accessible through learner profiles | This test case verifies that certificates and badges earned by a learner are accessible through their profile page. | Certificates & Badges | High | \* Learner has earned certificates and badges. | Learner ID | 1. Log in as the learner. 2. Navigate to the learner profile page. 3. Verify the availability of a "Certificates" or "Badges" section. 4. Access the certificates and badges sections. 5. Verify that all earned certificates and badges are displayed correctly. | Certificates and badges are displayed on the learner profile. | All earned certificates and badges are accessible through the learner profile. | Major | High | Functional Testing | Positive |

| TC-004 | Verify that learners can share their certificates and badges on social media | This test case verifies that learners can share their certificates and badges on social media platforms, such as Facebook, Twitter, or LinkedIn. | Certificates & Badges | Medium | \* Learner has earned certificates and badges. | Learner ID, Social Media Platform | 1. Log in as the learner. 2. Access the certificate or badge they want to share. 3. Verify the availability of social media sharing options. 4. Select the desired social media platform. 5. Verify that the certificate or badge is shared successfully on the chosen platform. | Certificate or badge shared on the selected social media platform. | Certificate or badge is successfully shared on social media. | Minor | Medium | Functional Testing | Positive |

| TC-005 | Verify that instructors can issue custom badges for special achievements | This test case verifies that instructors can issue custom badges to learners for special achievements or contributions, such as outstanding performance in a project or active participation in a discussion forum. | Certificates & Badges | Low | \* Instructor is managing a course. | Instructor ID, Course ID, Badge Details | 1. Log in as the instructor. 2. Navigate to the course management section. 3. Access the "Badges" section. 4. Create a new custom badge. 5. Assign the custom badge to a specific learner. 6. Verify that the custom badge is awarded to the learner. | Custom badge awarded to the learner. | Custom badge is successfully created and assigned to the learner. | Minor | Low | Functional Testing | 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 -**

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 - Online Learning Platform

**\*\*Project:\*\*** Online Learning Platform

**\*\*Feature:\*\*** Gamification Elements

**\*\*User Story:\*\*** As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC\_01

**\*\*Test Case Title:\*\*** Verify that learners earn points for completing activities

**\*\*Test Case Description:\*\*** This test case aims to verify that the platform accurately awards points to learners for successfully completing various learning activities, such as watching lectures, taking quizzes, and completing assignments.

**\*\*Test Suite:\*\*** Gamification Elements

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* A course with learning activities (e.g., lectures, quizzes, assignments) is enrolled.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access the enrolled course and navigate to a lecture.
2. Watch the complete lecture.
3. Complete a quiz within the course, ensuring all answers are correct.
4. Complete an assignment within the course and submit it.
5. Navigate to the user's gamification dashboard or profile.

**\*\*Postconditions:\*\***

\* The user is still logged in.

\* The gamification dashboard or profile is displayed.

**\*\*Expected Result:\*\*** The user's points should be updated to reflect the completion of the lecture, quiz, and assignment, according to the point system defined by the platform.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\* Functional Testing, Integration Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_02**

**\*\*Test Case Title:\*\* Verify that levels are achieved based on accumulated points**

**\*\*Test Case Description:\*\* This test case verifies that the platform correctly calculates and assigns levels to learners based on the points they accumulate by completing learning activities.**

**\*\*Test Suite:\*\* Gamification Elements**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* A course with learning activities (e.g., lectures, quizzes, assignments) is enrolled.

\* The user has completed enough activities to reach the next level based on the defined point system.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the enrolled course and complete various learning activities until the user reaches the point threshold required for the next level.

2. Navigate to the user's gamification dashboard or profile.

**\*\*Postconditions:\*\***

\* The user is still logged in.

\* The gamification dashboard or profile is displayed.

**\*\*Expected Result:\*\* The user's profile should reflect the newly achieved level, displaying the accumulated points and the level progress bar.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing, Integration Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_03**

**\*\*Test Case Title:\*\* Verify that leaderboards display top-performing learners**

**\*\*Test Case Description:\*\* This test case verifies that the platform accurately displays a leaderboard showcasing the top-performing learners based on their accumulated points.**

**\*\*Test Suite:\*\* Gamification Elements**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* Multiple user accounts are created and logged in.

\* Users have completed various learning activities in the same course.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the leaderboard section of the platform.

2. Observe the ranking of users on the leaderboard.

**\*\*Postconditions:\*\***

\* The user is still logged in.

\* The leaderboard is displayed.

**\*\*Expected Result:\*\* The leaderboard should accurately display the users ranked based on their accumulated points, showing their usernames, current levels, and points.**

**\*\*Severity:\*\* Minor**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing, Integration Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_04**

**\*\*Test Case Title:\*\* Verify that badges are awarded for reaching milestones**

**\*\*Test Case Description:\*\* This test case verifies that the platform correctly awards badges to learners upon**

reaching specific milestones, such as completing a course, achieving a certain level, or finishing a set number of activities.

**\*\*Test Suite:\*\*** Gamification Elements

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- \* A user account is created and logged in.

- \* A course with learning activities (e.g., lectures, quizzes, assignments) is enrolled.

- \* The user has completed activities or reached specific milestones in the course.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Complete activities or achieve milestones in the enrolled course (e.g., completing a course, achieving a specific level, completing a set number of activities).

2. Navigate to the user's gamification dashboard or profile.

**\*\*Postconditions:\*\***

- \* The user is still logged in.

- \* The gamification dashboard or profile is displayed.

**\*\*Expected Result:\*\*** The user's profile should display the earned badge, indicating the specific milestone achieved.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing, Integration Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_05

**\*\*Test Case Title:\*\*** Verify that learners can view their progress in a gamification dashboard

**\*\*Test Case Description:\*\*** This test case verifies that the platform provides a dedicated gamification dashboard where learners can track their progress, view their points, levels, badges, and other relevant



gamification elements.

**\*\*Test Suite:\*\*** Gamification Elements

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A user account is created and logged in.

\* The user has completed some learning activities.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the user's gamification dashboard.

2. Observe the available information on the dashboard.

**\*\*Postconditions:\*\***

\* The user is still logged in.

\* The gamification dashboard is displayed.

**\*\*Expected Result:\*\*** The gamification dashboard should display the user's current points, level, progress bar towards the next level, earned badges, and a summary of completed activities.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing, Integration Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 61**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Multilingual Support

**Feature description -**

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

**Userstory title -**

As a learner, I want multilingual support so that I can access the platform in my preferred language.

**Acceptance criteria -**

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated help and support resources.', 'Instructors can provide content in multiple languages.']

**Generated Test Cases:**

## Test Cases for Multilingual Support Feature

**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.

**Acceptance Criteria:**

1. The platform interface is available in multiple languages.
2. Course materials are available in different languages.
3. Learners can select their preferred language at any time.
4. Multilingual support includes translated help and support resources.
5. Instructors can provide content in multiple languages.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-ML-01

**\*\*Test Case Title:\*\*** Verify that the platform interface is available in multiple languages.

**\*\*Test Case Description:\*\*** This test case verifies the availability of the platform interface in different languages, ensuring that users can navigate the platform in their preferred language.

**\*\*Test Suite:\*\*** UI Testing

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- The platform is accessible and functional.
- The user is logged in to the platform.

**\*\*Test Data:\*\***

- List of supported languages (e.g., English, Spanish, French, German)

**\*\*Test Steps:\*\***

1. Access the platform using a supported language.
2. Navigate through different sections of the platform (e.g., homepage, course listing, user profile).
3. Verify that all interface elements (e.g., menus, buttons, text labels) are displayed in the chosen language.

**\*\*Postconditions:\*\***

- Logout from the platform.

**\*\*Expected Result:\*\*** The platform interface is displayed in the chosen language, including all elements, menus, buttons, and text labels.

**\*\*Severity:\*\* Blocker**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-ML-02**

**\*\*Test Case Title:\*\* Verify that course materials are available in different languages.**

**\*\*Test Case Description:\*\* This test case verifies that course materials, including videos, lectures, quizzes, and downloadable resources, are available in multiple languages.**

**\*\*Test Suite:\*\* Content Testing**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- The platform is accessible and functional.
- A course with multilingual content is available.

**\*\*Test Data:\*\***

- Course ID with multilingual content.
- List of supported languages for the course.

**\*\*Test Steps:\*\***

1. Select a course with multilingual content.
2. Access different course materials (e.g., videos, lectures, quizzes, downloadable resources).
3. Verify that each material is available in the chosen language.

**\*\*Postconditions:\*\***

- None

**\*\*Expected Result:\*\* All course materials are displayed in the selected language, ensuring content accessibility for learners.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-ML-03**

**\*\*Test Case Title:\*\* Verify that learners can select their preferred language at any time.**

**\*\*Test Case Description:\*\* This test case verifies that learners can easily select their preferred language from a list of available options.**

**\*\*Test Suite:\*\* User Interface Testing**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- The platform is accessible and functional.
- The user is logged in to the platform.

**\*\*Test Data:\*\***

- List of supported languages.

**\*\*Test Steps:\*\***

1. Navigate to the language settings section of the platform.
2. Select a language different from the current interface language.
3. Verify that the platform interface is immediately updated to the chosen language.

**\*\*Postconditions:\*\***

- None

**\*\*Expected Result:\*\* The platform interface language changes to the selected language without requiring page refreshes or reloading.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-ML-04**

**\*\*Test Case Title:\*\* Verify that multilingual support includes translated help and support resources.**

**\*\*Test Case Description:\*\* This test case verifies that help and support resources like FAQ, contact forms, and documentation are available in multiple languages.**

**\*\*Test Suite:\*\* Support & Documentation Testing**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- The platform is accessible and functional.
- The user is logged in to the platform.

**\*\*Test Data:\*\***

- List of supported languages.
- URL of the help & support section.

**\*\*Test Steps:\*\***

1. Access the help and support section of the platform.
2. Select a language different from the current interface language.
3. Verify that all content (FAQ, contact forms, documentation) is displayed in the chosen language.

**\*\*Postconditions:\*\***

- None

**\*\*Expected Result:\*\* All help and support resources are translated and displayed in the selected language.**

**\*\*Severity:\*\* Minor**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-ML-05**

**\*\*Test Case Title:\*\* Verify that instructors can provide content in multiple languages.**

**\*\*Test Case Description:\*\* This test case verifies that instructors can upload course materials in different languages, enabling them to reach a wider audience.**

**\*\*Test Suite:\*\* Instructor Functionality Testing**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- The platform is accessible and functional.
- An instructor is logged in to the platform.
- A course is created.

**\*\*Test Data:\*\***

- Instructor account credentials.
- Course ID.
- List of supported languages.

**\*\*Test Steps:\*\***

1. As an instructor, upload course materials in different languages (e.g., video lectures, quiz questions).
2. Verify that the uploaded materials are associated with the correct languages.
3. Check that learners can access the materials in their preferred language.

**\*\*Postconditions:\*\***

- None

**\*\*Expected Result:\*\* Instructors can upload course materials in multiple languages, and learners can access these materials based on their language preferences.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

-----

**Data - 62**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Accessibility Features

**Feature description -**

The platform prioritizes accessibility by incorporating features such as screen reader compatibility, adjustable font sizes, and color contrast options. These accessibility features ensure that learners with disabilities can access and navigate the platform with ease, promoting inclusivity and equitable access to educational opportunities.

**Userstory title -**

As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

**Acceptance criteria -**

['The platform is compatible with screen readers.', 'Learners can adjust font sizes for better readability.', 'Color contrast options are available for improved visibility.', 'Accessibility features are easy to activate and customize.', 'Support resources are available to assist learners with disabilities.']

**Generated Test Cases:**

## Test Cases for Online Learning Platform - Accessibility Features

**\*\*Project Context:\*\***



\* \*\*Project Name:\*\* Online Learning Platform

\* \*\*Project Description:\*\* An online platform offering courses, lectures, tutorials, and quizzes on various subjects.

\* \*\*Feature Name:\*\* Accessibility Features

\* \*\*User Story Title:\*\* As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

\*\*Acceptance Criteria:\*\*

- 1. The platform is compatible with screen readers.
- 2. Learners can adjust font sizes for better readability.
- 3. Color contrast options are available for improved visibility.
- 4. Accessibility features are easy to activate and customize.
- 5. Support resources are available to assist learners with disabilities.

\*\*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-001 | Verify that the platform is compatible with screen readers | This test case verifies that the platform can be navigated and interacted with using a screen reader. | Accessibility | High | The user has a screen reader installed and configured on their device. | No test data needed. | 1. Launch the platform using a screen reader. 2. Navigate through various sections of the platform, including course listings, lecture pages, quiz interfaces, and profile settings. 3. Verify that screen reader announcements accurately describe the content and functionality of each element. 4. Ensure that all interactive elements, like buttons and links, are properly

labelled for screen reader users. | None | The screen reader announces all platform elements and functionality correctly, allowing users to navigate and interact with the platform seamlessly. | Blocker | High | Functional Testing, Accessibility Testing | Positive |

| TC-002 | Verify that learners can adjust font sizes | This test case verifies that learners can easily adjust font sizes on the platform, improving readability. | Accessibility | High | The platform is launched. | No test data needed. | 1. Access the platform settings or preferences. 2. Locate the font size adjustment option. 3. Increase the font size to multiple levels. 4. Decrease the font size to multiple levels. 5. Verify that the font size changes are reflected across the platform content. | None | The font size changes seamlessly and affects all text elements on the platform. | Major | High | Functional Testing, Accessibility Testing | Positive |

| TC-003 | Verify the availability of color contrast options | This test case verifies the platform's ability to adjust color contrast levels, improving visibility for learners with visual impairments. | Accessibility | High | The platform is launched. | No test data needed. | 1. Access the platform settings or preferences. 2. Locate the color contrast adjustment options. 3. Select different color contrast modes (e.g., high contrast, low contrast, default). 4. Verify that the selected color contrast mode is applied to the entire platform. | None | Color contrast modes are applied correctly, making the platform content more visible and accessible for users with varying visual needs. | Major | High | Functional Testing, Accessibility Testing | Positive |

| TC-004 | Verify that accessibility features are easy to activate and customize | This test case ensures that accessibility features are readily accessible and easily customizable by learners. | Accessibility | High | The platform is launched. | No test data needed. | 1. Access the platform settings or preferences. 2. Locate the accessibility settings section. 3. Verify that accessibility features like font size, color contrast, screen reader compatibility, etc. are clearly labelled and easily toggled on/off. 4. Ensure that users can adjust the settings without requiring additional steps or navigating complex menus. | None | Accessibility features are conveniently located, clearly labelled, and readily customizable without requiring extensive navigation or effort. | Major | High | Usability Testing, Accessibility Testing | Positive |

| TC-005 | Verify that support resources for learners with disabilities are available | This test case ensures that learners with disabilities have access to appropriate support resources to address their needs. | Accessibility | High | The platform is launched. | No test data needed. | 1. Access the platform's help or support section. 2.

Locate information specifically related to accessibility features and support for learners with disabilities. 3. Verify the presence of resources such as FAQs, tutorials, contact details for accessibility specialists, and information about assistive technology compatibility. | None | Clear and readily accessible information regarding accessibility features, support resources, and assistive technology compatibility is available to learners with disabilities. | Major | High | Documentation Testing, Accessibility Testing | 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

**\*\*Project:\*\*** Online Learning Platform

**\*\*Feature:\*\*** Discussion Moderation

**\*\*User Story:\*\*** As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

**\*\*Acceptance Criteria:\*\***

1. Instructors can edit, delete, or highlight posts in discussions.
2. Instructors can mute or ban disruptive users.
3. Moderation actions are logged and reported.
4. Community guidelines are visible to all participants.
5. Notifications are sent for moderated actions.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Discussion Moderation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* Instructor logged in to the platform

\* A discussion thread with multiple posts exists

**\*\*Test Data:\*\***

\* Sample discussion post

\* Sample user account

**\*\*Test Steps:\*\***

\*

**\*\*Postconditions:\*\***

\*

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Priority:\*\***

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_DM\_01

**\*\*Test Case Title:\*\*** Verify that instructors can edit discussion posts.

**\*\*Test Case Description:\*\*** This test case verifies that an instructor can edit a post within a discussion thread, ensuring the functionality for correcting errors or improving content.

**\*\*Preconditions:\*\***

\* Instructor logged in to the platform

\* A discussion thread with multiple posts exists

**\*\*Test Data:\*\***

\* Sample discussion post with a typo

**\*\*Test Steps:\*\***

1. Navigate to the discussion thread.
2. Locate the post with the typo.
3. Click the edit button for the post.
4. Correct the typo within the post editor.
5. Click the "Save" or "Update" button.

**\*\*Postconditions:\*\***

\* The edited post should be reflected in the discussion thread.

**\*\*Expected Result:\*\*** The post should be successfully edited with the corrected content.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_DM\_02

**\*\*Test Case Title:\*\*** Verify that instructors can delete discussion posts.

**\*\*Test Case Description:\*\*** This test case verifies that an instructor can delete a post from a discussion thread, ensuring the functionality for removing inappropriate or irrelevant content.

**\*\*Preconditions:\*\***

\* Instructor logged in to the platform

\* A discussion thread with multiple posts exists

**\*\*Test Data:\*\***

\* Sample discussion post containing irrelevant content

**\*\*Test Steps:\*\***

1. Navigate to the discussion thread.
2. Locate the post containing irrelevant content.

3. Click the delete button for the post.

4. Confirm the deletion action.

**\*\*Postconditions:\*\***

\* The deleted post should be removed from the discussion thread.

**\*\*Expected Result:\*\*** The post should be successfully deleted from the discussion thread.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_DM\_03

**\*\*Test Case Title:\*\*** Verify that instructors can mute disruptive users in discussions.

**\*\*Test Case Description:\*\*** This test case verifies that an instructor can mute a user who is disrupting the discussion thread, preventing them from posting further until unmuted.

**\*\*Preconditions:\*\***

\* Instructor logged in to the platform

\* A discussion thread exists

\* A user account is actively posting disruptive content

**\*\*Test Data:\*\***

\* Sample user account

**\*\*Test Steps:\*\***

1. Navigate to the discussion thread.

2. Identify the user posting disruptive content.

3. Select the option to mute the user from the discussion.

4. Confirm the mute action.

**\*\*Postconditions:\*\***

- \* The disruptive user should be muted from the discussion.
- \* The user should be unable to post further in the discussion until unmuted.

**\*\*Expected Result:\*\*** The user should be successfully muted from the discussion thread.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_DM\_04

**\*\*Test Case Title:\*\*** Verify that community guidelines are visible to all participants in discussions.

**\*\*Test Case Description:\*\*** This test case verifies that the community guidelines are prominently displayed within the discussion forum, allowing participants to understand expected behavior and guidelines for respectful discourse.

**\*\*Preconditions:\*\***

- \* A user is logged in to the platform.
- \* A discussion thread exists.

**\*\*Test Data:\*\***

- \* No test data needed

**\*\*Test Steps:\*\***

1. Navigate to any discussion thread.
2. Locate the "Community Guidelines" section or link within the discussion area.
3. Read the displayed guidelines.

**\*\*Postconditions:\*\***

- \* No postconditions.



**\*\*Expected Result:\*\*** The community guidelines should be clearly visible within the discussion forum.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_DM\_05

**\*\*Test Case Title:\*\*** Verify that notifications are sent to instructors for moderated actions.

**\*\*Test Case Description:\*\*** This test case verifies that instructors receive notifications when they perform moderation actions within discussion forums, providing them with immediate feedback and transparency regarding their actions.

**\*\*Preconditions:\*\***

\* Instructor logged in to the platform.

\* A discussion thread exists.

**\*\*Test Data:\*\***

\* No test data needed

**\*\*Test Steps:\*\***

1. Perform a moderation action within a discussion thread, such as deleting a post or muting a user.
2. Check the instructor's notification center or email inbox.

**\*\*Postconditions:\*\***

\* No postconditions.

**\*\*Expected Result:\*\*** A notification should be sent to the instructor regarding the performed moderation action.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

-----

**Data - 64**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Learning Analytics

**Feature description -**

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement, performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning experience to meet learners' needs effectively.

**Userstory title -**

As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

**Acceptance criteria -**

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.', 'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.', 'Instructors can export analytics reports for further analysis.']

**Generated Test Cases:**

## Test Cases for Learning Analytics Feature

**Project:** Online Learning Platform

**Feature:** Learning Analytics

**User Story:** As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

**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-01	Verify that learning analytics dashboards display engagement metrics	This test case verifies that the learning analytics dashboard displays key engagement metrics such as video completion rates, forum participation, and resource downloads.	Learning Analytics	High	User is logged in as an instructor.	Course with student activity	1. Navigate to the learning analytics dashboard for a specific course. 2. Observe the dashboard for various engagement metrics such as video completion rates, forum participation, and resource downloads. 3. Verify that the data displayed is relevant to the course and represents real-time engagement levels.	None	Engagement metrics should be displayed accurately and in real-time, including video completion rates, forum participation, and resource downloads.	Major	High	Functional Testing	Positive
TC-02	Verify that performance trends are visualized over time	This test case verifies that the learning analytics dashboard visualizes performance trends over time, allowing instructors to identify patterns and areas of improvement.	Learning Analytics	High	User is logged in as an instructor.	Course with student activity	1. Navigate to the learning analytics dashboard for a specific course. 2. Select a time period (e.g., weekly, monthly, semester). 3. Observe the visualization of performance trends over the selected time period, such as average quiz scores, progress through modules, or overall course performance. 4. Verify that the						

visualization accurately represents the trends over time and includes relevant data points. | None | The performance trends should be visualized clearly and accurately over time, including data points and relevant trends. | Major | High | Functional Testing | Positive |

| TC-03 | Verify that instructors can access detailed reports on learner progress | This test case verifies that instructors can access detailed reports on individual learner progress, providing insights into their strengths and areas for improvement. | Learning Analytics | Medium | User is logged in as an instructor. | Course with student activity | 1. Navigate to the learning analytics dashboard for a specific course. 2. Select "Learner Progress" or "Individual Reports" option. 3. View detailed reports on individual student progress, including quiz scores, module completion, assignment submissions, and overall course performance. 4. Verify that the reports are comprehensive, accurate, and provide relevant information on individual learner progress. | None | Detailed reports on individual learner progress should be accessible, accurate, and provide comprehensive insights into their performance and learning journey. | Major | Medium | Functional Testing | Positive |

| TC-04 | Verify that analytics data is updated in real-time | This test case ensures that the learning analytics dashboard displays real-time data updates, reflecting changes in learner engagement and performance. | Learning Analytics | High | User is logged in as an instructor. | Course with active learners | 1. Navigate to the learning analytics dashboard for a specific course. 2. Observe the data displayed for a particular engagement metric (e.g., video completion rates). 3. Initiate a learning activity (e.g., watch a video, participate in a forum, complete a quiz). 4. Observe the analytics dashboard for immediate updates reflecting the changes in learner engagement and performance. | None | The analytics dashboard should display real-time updates reflecting changes in learner engagement and performance as soon as the activities are completed. | Critical | High | Functional Testing | Positive |

| TC-05 | Verify that instructors can export analytics reports for further analysis | This test case verifies that instructors can export analytics reports in various formats for offline analysis or integration with other tools. | Learning Analytics | Medium | User is logged in as an instructor. | Course with student activity | 1. Navigate to the learning analytics dashboard for a specific course. 2. Select "Export Report" or similar option. 3. Choose an export format (e.g., CSV, PDF). 4. Verify that the export process is successful and the report is downloaded in the selected format. 5. Open the exported report and verify that it contains all the relevant data from the

learning analytics dashboard. | None | Instructors should be able to export analytics reports in various formats (e.g., CSV, PDF) for further analysis or integration with other tools. The exported reports should contain accurate and comprehensive data. | Minor | Medium | Functional Testing | Positive |

-----

**Data - 65**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Social Learning Features

**Feature description -**

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

**Userstory title -**

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

**Acceptance criteria -**

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

**Generated Test Cases:**

## ## Test Cases for Social Learning Features

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*User Story Title:\*\* As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.**

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC\_SLF\_01**

**\*\*Test Case Title:\*\* Verify that learners can create and join groups for projects.**

**\*\*Test Case Description:\*\* This test case verifies that learners can create new groups and join existing groups for collaborative projects. It will test the functionality of group creation, joining, and viewing group members.**

**\*\*Test Suite:\*\* Social Learning Features**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- \* A user account is created and logged into the platform.
- \* At least one course with group project options is available.

**\*\*Test Data:\*\***

- \* Group name: "Data Structures Study Group"
- \* Course name: "Introduction to Data Structures"

**\*\*Test Steps:\*\***

1. Navigate to the "Course" page and select the course with group project options.
2. Click on the "Group Projects" tab.
3. Click on the "Create Group" button.
4. Enter the group name and a description (optional).
5. Select the relevant course from the dropdown list.
6. Click on the "Create Group" button.

7. Verify that the group is created successfully and the user is added as a member.
8. Navigate back to the "Group Projects" tab.
9. Click on the "Join Group" button for a group created by another user.
10. Confirm the group membership and view the list of members.

**\*\*Postconditions:\*\***

- \* The user is now a member of the created group.
- \* The user is now a member of the joined group.

**\*\*Expected Result:\*\***

- \* The user should be able to create a group for a selected course and add a description (optional).
- \* The user should be able to join an existing group and view the group members.

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_SLF\_02**

**\*\*Test Case Title:\*\* Verify that collaboration tools are available for group communication.**

**\*\*Test Case Description:\*\*** This test case ensures that the platform provides effective communication tools within groups, such as message boards, chat, and file sharing.

**\*\*Test Suite:\*\* Social Learning Features**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- \* A user account is created and logged into the platform.
- \* A user is a member of an existing group.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Group Projects" tab.
2. Select an existing group.
3. Access the group's communication tools, such as message boards, chat, or file sharing options.
4. Post a message on the message board.
5. Send a message through the chat functionality.
6. Upload a document to the file sharing section.
7. View the posted message on the message board, the sent chat message, and the uploaded file.

**\*\*Postconditions:\*\***

- \* The user has successfully interacted with the communication tools within the group.

**\*\*Expected Result:\*\***

- \* The platform should provide tools for communication within groups, allowing users to post messages, send chat messages, and share files.

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_SLF\_03**

**\*\*Test Case Title:\*\* Verify that learners can share resources and feedback within groups.**

**\*\*Test Case Description:\*\* This test case checks if learners can share relevant resources and provide feedback on shared materials within groups.**

**\*\*Test Suite:\*\* Social Learning Features**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- \* A user account is created and logged into the platform.

- \* A user is a member of an existing group.

**\*\*Test Data:\*\***



\* A relevant document to share.

\* A feedback message.

**\*\*Test Steps:\*\***

1. Navigate to the "Group Projects" tab.
2. Select an existing group.
3. Locate the resource sharing functionality within the group.
4. Upload a relevant document.
5. Provide feedback on an uploaded resource by other members.
6. View the shared resources and feedback within the group.

**\*\*Postconditions:\*\***

\* The user has shared a resource and provided feedback within the group.

**\*\*Expected Result:\*\***

\* The platform should allow users to share relevant resources with group members and provide feedback on shared content.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_SLF\_04**

**\*\*Test Case Title:\*\* Verify that instructors can assign group projects and monitor progress.**

**\*\*Test Case Description:\*\* This test case checks if instructors can assign group projects to their students and monitor their progress within assigned groups.**

**\*\*Test Suite:\*\* Social Learning Features**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A user account is created and logged into the platform with instructor privileges.

\* A course is created with group project options.

**\*\*Test Data:\*\***

\* A group project description.

\* A deadline for the project.

**\*\*Test Steps:\*\***

1. Navigate to the "Course" page and select the course with group project options.
2. Click on the "Group Projects" tab.
3. Click on the "Assign Group Project" button.
4. Enter the project description and set a deadline.
5. Select the relevant course from the dropdown list.
6. Assign the project to a specific group.
7. View the progress of the assigned group project and track student activity within the group.

**\*\*Postconditions:\*\***

\* The instructor has successfully assigned a group project and can track progress within the group.

**\*\*Expected Result:\*\***

\* The platform should allow instructors to assign group projects with deadlines, view project details, and monitor group activity.

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_SLF\_05**

**\*\*Test Case Title:\*\* Verify that social sharing features allow learners to share achievements.**

**\*\*Test Case Description:\*\* This test case checks the functionality of social sharing features, allowing learners to share their achievements (e.g., completing a course, achieving a high score on a quiz) with their social networks.**

**\*\*Test Suite:\*\* Social Learning Features**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- \* A user account is created and logged into the platform.
- \* A user has completed a course or achieved a high score on a quiz.

**\*\*Test Data:\*\***

- \* A user's social media profile link (optional).

**\*\*Test Steps:\*\***

1. Navigate to the user's profile page.
2. Locate the "Share Achievements" section.
3. Select the achievement to share (e.g., completed course, high score on quiz).
4. Choose the social media platform to share the achievement (e.g., Facebook, Twitter, LinkedIn).
5. (Optional) Enter a custom message for the shared post.
6. Click on the "Share" button.
7. Verify that the achievement is shared on the selected platform.

**\*\*Postconditions:\*\***

- \* The user's achievement is shared on the selected social media platform.

**\*\*Expected Result:\*\***

- \* The platform should provide a dedicated section for sharing achievements with social networks, allowing users to select their desired platform and customize the shared message.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

-----

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

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 Online Learning Platform - Content Authoring Tools

**\*\*Project Name:\*\*** Online Learning Platform

**\*\*Feature Name:\*\*** Content Authoring Tools

**\*\*User Story Title:\*\*** As an instructor, I want powerful content authoring tools so that I can create engaging

and interactive course materials.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-001

**\*\*Test Case Title:\*\*** Verify that instructors can create multimedia presentations.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of the content authoring tool to allow instructors to create multimedia presentations using various media types, including images, videos, and audio.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* Instructor account is created and logged in.

\* A course is created.

\* Content authoring tools are accessible.

**\*\*Test Data:\*\***

\* Images (jpg, png)

\* Videos (mp4)

\* Audio (mp3)

\* Text content

**\*\*Test Steps:\*\***

1. Access the course content creation area.
2. Select "Multimedia Presentation" as the content type.
3. Upload images, videos, and audio files into the presentation.
4. Add text content to the presentation.
5. Arrange media elements and text within the presentation.
6. Save the presentation.

**\*\*Postconditions:\*\***

\* The multimedia presentation is saved in the course content.

\* Preview option for the presentation is available.

**\*\*Expected Result:\*\*** The system allows instructors to create multimedia presentations by uploading various media types, adding text, and arranging content elements. The saved presentation can be previewed.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-002

**\*\*Test Case Title:\*\*** Verify that interactive modules can be developed and integrated.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of the content authoring tool to allow instructors to create and integrate interactive modules within their courses.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* Instructor account is created and logged in.

\* A course is created.

\* Content authoring tools are accessible.

**\*\*Test Data:\*\***

\* Pre-defined interactive module templates (e.g., quizzes, polls, surveys)

\* Custom questions and answer options for quizzes.

**\*\*Test Steps:\*\***

1. Access the course content creation area.

2. Select "Interactive Module" as the content type.

3. Choose a pre-defined module template or create a custom module.

4. Configure the interactive module with relevant content and parameters (e.g., question types, answer options, feedback messages).
5. Integrate the interactive module into the course content.
6. Save the course content.

**\*\*Postconditions:\*\***

- \* The interactive module is saved within the course content.
- \* The module is accessible for learners during the course.

**\*\*Expected Result:\*\*** The system allows instructors to create and integrate interactive modules using pre-defined templates or custom settings. The interactive modules are displayed correctly and function as intended within the course.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-003

**\*\*Test Case Title:\*\*** Verify that rich media resources, such as videos and animations, are supported.

**\*\*Test Case Description:\*\*** This test case verifies the system's ability to upload, embed, and display rich media resources, such as videos and animations, within course content.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* Instructor account is created and logged in.
- \* A course is created.
- \* Content authoring tools are accessible.

**\*\*Test Data:\*\***

\* Video files (mp4, webm)

\* Animation files (gif, webm)

\* External video links (YouTube, Vimeo)

**\*\*Test Steps:\*\***

1. Access the course content creation area.
2. Choose the desired content type (e.g., lecture, module, assignment).
3. Upload video files directly or embed external videos via URL.
4. Upload and embed animation files.
5. Preview the content to ensure proper video and animation playback.
6. Save the content.

**\*\*Postconditions:\*\***

\* Video and animation files are saved within the course content.

\* Video and animation playback is smooth and uninterrupted.

**\*\*Expected Result:\*\*** The system allows instructors to upload, embed, and display rich media resources like videos and animations within the course content. The media files are displayed correctly and play without errors.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-004

**\*\*Test Case Title:\*\*** Verify that content authoring tools are user-friendly and intuitive.

**\*\*Test Case Description:\*\*** This test case verifies the usability and intuitiveness of the content authoring tools by evaluating the interface design, navigation, and ease of use for instructors.

**\*\*Test Suite:\*\*** Content Authoring Tools



**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* Instructor account is created and logged in.

\* A course is created.

\* Content authoring tools are accessible.

**\*\*Test Data:\*\***

\* No test data needed

**\*\*Test Steps:\*\***

1. Access the content authoring tools interface.

2. Explore the available features, tools, and options.

3. Attempt to create a simple content item (e.g., text, image).

4. Navigate between different sections and tabs within the tools.

5. Evaluate the ease of use and understanding of the tool's functionality.

6. Seek feedback from instructors on the usability and intuitiveness of the tools.

**\*\*Postconditions:\*\***

\* No postconditions.

**\*\*Expected Result:\*\*** Instructors find the content authoring tools easy to use and navigate, with clear and understandable features and functions. The interface design is user-friendly and intuitive.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Usability Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-005**

**\*\*Test Case Title:\*\* Verify that instructors can preview and edit content before publishing.**

**\*\*Test Case Description:\*\* This test case verifies the functionality to allow instructors to preview and edit**

created content before publishing it to learners.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* Instructor account is created and logged in.

- \* A course is created.

- \* Content is created within the course using the authoring tools.

**\*\*Test Data:\*\***

- \* No test data needed

**\*\*Test Steps:\*\***

1. Access the created content within the course.

2. Utilize the "Preview" function to view the content as it will appear to learners.

3. Identify areas for improvement or editing.

4. Use the editing functions to modify the content as needed.

5. Save the edited content.

6. Re-preview the content to ensure desired changes are reflected.

**\*\*Postconditions:\*\***

- \* The content is saved with the applied edits.

- \* The instructor has the option to publish the content to learners.

**\*\*Expected Result:\*\*** The system allows instructors to preview and edit their created content before publishing. The preview function displays the content accurately, and the editing functions allow instructors to make necessary changes.

**\*\*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:**

## Test Cases for Learning Paths and Curriculum Feature

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Learning Paths and Curriculum**

**\*\*User Story Title:\*\* As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.**

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC-LP-01**

**\*\*Test Case Title:\*\* Verify that Learning Paths include sequential courses and modules.**

**\*\*Test Case Description:\*\* This test case verifies that the learning paths feature correctly presents courses and modules in a defined order, ensuring learners follow a structured progression through the content.**

**\*\*Test Suite:\*\* Learning Paths and Curriculum**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A learning path is created with multiple courses and modules.

\* A user is logged in and has access to the learning path.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the learning path.
2. Navigate through the courses and modules.
3. Observe the order of presentation for each course and module.
4. Verify that the order of courses and modules is consistent with the defined learning path sequence.

**\*\*Postconditions:\*\* None**

**\*\*Expected Result:\*\* Courses and modules are presented in a sequential order within the learning path.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TC-LP-02**

**\*\*Test Case Title:\*\* Verify that Curricula are aligned with specific learning objectives.**

**\*\*Test Case Description:\*\* This test case ensures that each learning path's curriculum is clearly linked to specific learning objectives, providing learners with a clear understanding of the skills and knowledge they will gain upon completion.**

**\*\*Test Suite:\*\* Learning Paths and Curriculum**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A learning path is created with defined learning objectives.

\* A user is logged in and has access to the learning path.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the learning path.
2. View the learning objectives for the path.
3. Review the curriculum content (courses and modules).
4. Verify that the curriculum content directly relates to and aligns with the stated learning objectives.

**\*\*Postconditions:\*\* None**

**\*\*Expected Result:\*\* The curriculum content aligns with and supports the specified learning objectives for the learning path.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* TC-LP-03**

**\*\*Test Case Title:\*\* Verify that learners can track their progress within a learning path.**

**\*\*Test Case Description:\*\* This test case verifies that learners can track their progress through a learning path, including completed courses, modules, and overall progress.**

**\*\*Test Suite:\*\* Learning Paths and Curriculum**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A learning path is created.

\* A user is logged in and enrolled in the learning path.

\* The user has completed some courses and modules within the learning path.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the learning path.
2. View the progress tracker for the learning path.
3. Verify that completed courses and modules are marked as completed.
4. Check the overall progress percentage displayed.

**\*\*Postconditions:\*\* None**

**\*\*Expected Result:\*\* The progress tracker accurately reflects the learner's progress through the learning path, showing completed courses, modules, and overall percentage completion.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* TC-LP-04**

**\*\*Test Case Title:\*\*** Verify that Instructors can customize learning paths for different needs.

**\*\*Test Case Description:\*\*** This test case ensures that instructors have the ability to customize learning paths based on different learning needs, such as skill levels, interests, or specific learning objectives.

**\*\*Test Suite:\*\*** Learning Paths and Curriculum

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* An instructor is logged in and has access to the learning path management.

\* A learning path is created.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. As an instructor, access the learning path management section.
2. Select a learning path for customization.
3. Edit the learning path by adding or removing courses, modules, or re-ordering content.
4. Verify that the modifications are saved and reflected in the learning path's content.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** Instructors can successfully customize the learning path content to meet specific learning needs.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC-LP-05**

**\*\*Test Case Title:\*\* Verify that Notifications are sent for upcoming courses or milestones.**

**\*\*Test Case Description:\*\* This test case ensures that learners receive notifications about upcoming courses or milestones within a learning path, facilitating timely engagement and progress.**

**\*\*Test Suite:\*\* Learning Paths and Curriculum**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* A learning path is created with upcoming courses or milestones.

\* A user is enrolled in the learning path and has notification settings enabled.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Enroll in the learning path.
2. Enable notification settings for upcoming courses and milestones.
3. Monitor the notification channel (email, in-app notification, etc.).
4. Verify that notifications are received for upcoming courses and milestones within the specified time frame.

**\*\*Postconditions:\*\* None**

**\*\*Expected Result:\*\* Learners receive timely notifications about upcoming courses and milestones within a learning path, reminding them of deadlines and encouraging progress.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

-----

**Data - 68**

**Project name -**



## Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Attendance Tracking

### **Feature description -**

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

### **Userstory title -**

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

### **Acceptance criteria -**

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

### **Generated Test Cases:**

## Test Cases for Attendance Tracking Feature

**\*\*Project:\*\*** Online Learning Platform

**\*\*User Story:\*\*** As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

**\*\*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
TC001	Login Functionality	User can login with valid credentials	Authentication	High	System is up and running	Valid username and password	1. Enter username 2. Enter password 3. Click Login	User is redirected to dashboard	Successful login	Critical	High	Manual	Standard

TC-ATT-01	Verify that attendance is automatically recorded during live sessions.	This test case validates that the system accurately captures attendance data for learners who join a live session.	Attendance Tracking	High	* A live session is scheduled and initiated. * Learners are logged in and join the session.	No test data needed	1. Start a live session. 2. Observe the attendance list during the session. 3. Verify that all learners who joined the session are recorded as "present" in the attendance list.	No postconditions needed	The attendance list should accurately reflect the learners who are present in the live session.	Major	High	Functional Testing	Positive
-----------	--	--	---------------------	------	---	---------------------	--	--------------------------	---	-------	------	--------------------	----------

TC-ATT-02   Verify that instructors can view attendance reports for each session.   This test case ensures that
instructors have access to attendance data for each session, allowing them to monitor participation.
Attendance Tracking   High   * A live session is completed. * Instructor is logged in.   No test data needed   1.
Go to the session details page. 2. Navigate to the "Attendance" tab. 3. Verify that the attendance report
displays a list of all learners who attended the session, along with their attendance status (present, absent, etc.).
No postconditions needed   The attendance report should display a complete and accurate list of learners who
attended the session, with their respective attendance status.   Major   High   Functional Testing   Positive

TC-ATT-03	Verify that learners can see their attendance records.	This test case ensures that learners can access their own attendance data for all sessions they have attended.	Attendance Tracking	Medium	* A live session is completed. * Learner is logged in.	No test data needed	1. Go to the "My Courses" page. 2. Select the course containing the attended session. 3. Navigate to the "Attendance" tab. 4. Verify that the attendance record displays a list of all sessions attended by the learner, along with their attendance status.	No postconditions needed	The attendance record should display a complete and accurate list of sessions attended by the learner, with their respective attendance status.	Minor	Medium	Functional Testing	Positive
-----------	--	--	---------------------	--------	---	---------------------	---	--------------------------	---	-------	--------	--------------------	----------

TC-ATT-04   Verify that attendance data is integrated with participation metrics.   This test case ensures that attendance data is linked with other participation metrics, providing a holistic view of learner engagement.	
--	--

Attendance Tracking | Medium | \* A live session is completed. \* Instructor is logged in. | No test data needed |

1. Go to the session details page. 2. Navigate to the "Analytics" tab. 3. Verify that the analytics dashboard displays attendance data alongside other engagement metrics like active participation (chat, questions, etc.). |

No postconditions needed | The analytics dashboard should display attendance data integrated with other relevant participation metrics, providing a comprehensive view of learner engagement. | Minor | Medium |

Functional Testing | Positive |

| TC-ATT-05 | Verify that notifications are sent for missed sessions. | This test case ensures that learners receive timely notifications about missed sessions, promoting accountability and encouraging participation. |

Attendance Tracking | High | \* A live session is scheduled. \* Learner is logged in. \* Learner misses the session. | No test data needed | 1. Schedule a live session. 2. Log in as a learner and miss the session. 3.

Check the learner's notification settings. 4. Verify that a notification is sent to the learner indicating the missed session. | No postconditions needed | The learner should receive a notification about the missed

session, providing details about the session and encouraging them to catch up. | Major | High | Functional Testing | Negative |

-----

**Data - 69**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Content Licensing and Rights Management

**Feature description -**

The platform implements robust content licensing and rights management features to protect intellectual property and ensure compliance with copyright laws. By managing content licenses, usage rights, and permissions, the platform safeguards the interests of content creators and providers, enabling secure and lawful distribution of educational materials.

#### **Userstory title -**

As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

#### **Acceptance criteria -**

['Content licenses are assigned to each resource.', 'Usage rights and permissions are clearly defined.', 'Content creators can set restrictions on usage and distribution.', 'Instructors and learners are informed of content licensing terms.', 'Notifications are sent for license expiration or renewal.']

#### **Generated Test Cases:**

## Test Cases for Content Licensing and Rights Management

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Content Licensing and Rights Management**

**\*\*User Story:\*\*** As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

**\*\*Acceptance Criteria:\*\***

1. Content licenses are assigned to each resource.
2. Usage rights and permissions are clearly defined.
3. Content creators can set restrictions on usage and distribution.
4. Instructors and learners are informed of content licensing terms.

5. Notifications are sent for license expiration or renewal.

**\*\*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_CLRM_01	Verify that content licenses are assigned to each resource	This test case verifies that the platform automatically assigns content licenses to each uploaded resource, such as videos, documents, and quizzes.	Content Licensing	High	A content creator account is created and logged in.	A video file is uploaded to the platform.	1. Upload a video file to the platform. 2. Navigate to the uploaded video's details page. 3. Verify that a content license is assigned to the video.	The uploaded video is successfully added to the platform.	A content license is correctly assigned to the uploaded video.	Major	High	Functional Testing	Positive
TC_CLRM_02	Verify that usage rights and permissions are clearly defined	This test case verifies that the platform clearly defines usage rights and permissions for each content license, including access, download, and sharing limitations.	Content Licensing	High	A content creator account is created and logged in.	A document is uploaded with a Creative Commons license.	1. Upload a document with a Creative Commons license. 2. View the document's details page. 3. Verify that the usage rights and permissions associated with the Creative Commons license are clearly displayed.	The document is uploaded successfully, and the assigned license is visible.	The document's details page clearly displays usage rights and permissions according to the selected Creative Commons license.	Major	High	Functional Testing	Positive
TC_CLRM_03	Verify that content creators can set restrictions on usage and distribution	This test case verifies that content creators can define restrictions on usage and distribution of their content, such as limiting downloads, embedding, or sharing.	Content Licensing	High	A content creator account is created and logged in.	A tutorial is uploaded with custom usage restrictions.	1. Upload a tutorial and access the licensing settings. 2. Configure restrictions to limit downloads and sharing. 3. Save the settings and attempt to						

download/share the tutorial. | The tutorial is successfully uploaded and settings saved. | The platform restricts downloads and sharing based on the configured settings, preventing unauthorized access. | Major | High | Functional Testing | Negative |

| TC\_CLRM\_04 | Verify that instructors and learners are informed of content licensing terms | This test case verifies that the platform clearly displays content licensing terms to instructors and learners before accessing licensed content. | Content Licensing | High | A content creator uploads a video with a restricted license. | A video file with a restricted license is uploaded. | 1. Upload a video with a restricted license. 2. Access the video from the instructor's perspective. 3. Access the video from the learner's perspective. 4. Verify that the license terms are prominently displayed before accessing the video. | The video is accessible to both instructors and learners. | The content licensing terms are clearly displayed to both instructors and learners before accessing the video. | Major | High | Functional Testing | Positive |

| TC\_CLRM\_05 | Verify that notifications are sent for license expiration or renewal | This test case verifies that the platform sends timely notifications to content creators when their content licenses are nearing expiration or require renewal. | Content Licensing | High | A content creator uploads a video with a limited-time license. | A video is uploaded with a 30-day license. | 1. Upload a video with a 30-day license. 2. Set the system date to 25 days after the license start date. 3. Verify that the content creator receives a notification regarding the upcoming license expiration. | The system date is updated successfully. | The platform sends a notification to the content creator informing them about the license expiration date and renewal options. | Major | High | System Testing | Positive |

**\*\*Objective:\*\***

These test cases comprehensively cover the acceptance criteria outlined for the Content Licensing and Rights Management feature, ensuring that the platform effectively protects intellectual property, manages usage rights, and provides clear communication to all users.

-----

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Content Versioning

**Feature description -**

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

**Userstory title -**

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

**Acceptance criteria -**

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.', 'Notifications are sent for significant content updates.']

**Generated Test Cases:**

## Test Cases for Content Versioning Feature

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Content Versioning**

**\*\*User Story:\*\*** As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

**\*\*Acceptance Criteria:\*\***

1. Instructors can create and save multiple versions of content.
2. Version history is maintained and accessible.
3. Instructors can compare different versions of content.
4. Previous versions can be restored if needed.
5. Notifications are sent for significant content updates.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC-CV-01**

**\*\*Test Case Title:\*\*** Verify that instructors can create and save multiple versions of content.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of creating and saving multiple versions of content. The instructor will create a new version of a course module, make changes to the content, and save the new version.

**\*\*Test Suite:\*\* Content Versioning**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* An instructor account is created and logged in.

\* A course module with existing content is available.



**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course module page.
2. Click on the "Create New Version" button.
3. Modify the existing content, for example, add a new paragraph, update a heading, or change the content of a video lecture.
4. Save the new version.

**\*\*Postconditions:\*\***

- \* A new version of the course module is created and saved.
- \* The new version is accessible for viewing and editing.

**\*\*Expected Result:\*\*** A new version of the course module is successfully created and saved. The version history should reflect the new version with a timestamp and a description of the changes.

**\*\*Severity:\*\*** Blocker

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CV-02

**\*\*Test Case Title:\*\*** Verify that version history is maintained and accessible.

**\*\*Test Case Description:\*\*** This test case verifies that the platform maintains and allows access to the version history of course content. The instructor will view the version history of a course module and verify that all previous versions are listed with their timestamps and change descriptions.

**\*\*Test Suite:\*\*** Content Versioning

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* An instructor account is created and logged in.

\* A course module with multiple saved versions is available.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course module page.
2. Access the "Version History" section.
3. Verify that all previous versions of the module are listed.
4. Check that each version has a timestamp and a description of the changes made.

**\*\*Postconditions:\*\***

\* The version history is displayed correctly.

\* All previous versions are accessible for viewing and restoring.

**\*\*Expected Result:\*\*** The version history displays all saved versions of the course module, including their timestamps and change descriptions.

**\*\*Severity:\*\*** Blocker

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CV-03

**\*\*Test Case Title:\*\*** Verify that instructors can compare different versions of content.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of comparing different versions of content. The instructor will select two versions of a course module and compare them to see the differences between the content.

**\*\*Test Suite:\*\*** Content Versioning

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* An instructor account is created and logged in.

\* A course module with at least two different versions is available.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course module page.
2. Select two different versions from the "Version History" section.
3. Use the "Compare Versions" functionality to highlight the differences between the selected versions.

**\*\*Postconditions:\*\***

\* The comparison between the two versions is displayed correctly.

\* The highlighted differences are clearly visible.

**\*\*Expected Result:\*\*** The platform provides a visual comparison of the selected versions, highlighting the differences in the content.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CV-04

**\*\*Test Case Title:\*\*** Verify that previous versions can be restored if needed.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of restoring previous versions of content.

The instructor will restore an older version of a course module and verify that the content is restored correctly.

**\*\*Test Suite:\*\*** Content Versioning

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* An instructor account is created and logged in.

\* A course module with at least two different versions is available.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course module page.
2. Select an older version from the "Version History" section.
3. Click on the "Restore Version" button.
4. Confirm the restoration process.

**\*\*Postconditions:\*\***

- \* The selected older version is restored.
- \* The current version of the course module reverts to the restored version.

**\*\*Expected Result:\*\*** The course module is successfully restored to the selected older version. The content of the module should match the content of the restored version.

**\*\*Severity:\*\*** Blocker

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CV-05

**\*\*Test Case Title:\*\*** Verify that notifications are sent for significant content updates.

**\*\*Test Case Description:\*\*** This test case verifies that the platform sends notifications to relevant users (instructors and students) when significant content updates are made. The instructor will make a significant change to a course module, and the notification should be sent to both the instructor and the enrolled students.

**\*\*Test Suite:\*\*** Content Versioning

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- \* An instructor account is created and logged in.
- \* A course module with enrolled students is available.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Make a significant change to the course module content, such as adding a new module, deleting an existing module, or making major edits to an existing module.
2. Save the new version.

**\*\*Postconditions:\*\***

- \* The notification is sent to the instructor.
- \* The notification is sent to all enrolled students.

**\*\*Expected Result:\*\*** The platform sends a notification to both the instructor and enrolled students informing them about the significant content update.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 71**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

API Integration

**Feature description -**

The platform supports API integration, allowing third-party applications to interact with the platform's data

and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

**Userstory title -**

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

**Acceptance criteria -**

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

**Generated Test Cases:**

## Test Cases for API Integration Feature

**\*\*Project:\*\*** Online Learning Platform

**\*\*User Story:\*\*** As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

**\*\*Acceptance Criteria:\*\***

1. APIs are available for key platform functionalities.
2. Comprehensive API documentation is provided.
3. Secure access protocols are implemented.
4. Third-party applications can authenticate and access APIs.
5. Rate limiting and monitoring features are in place.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\*** TC\_API\_[Number]

**\*\*Test Case Title:\*\*** Verify that...

**\*\*Test Case Description:\*\*** (Brief description of the test case)

**\*\*Test Suite:\*\*** API Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* The online learning platform is running and accessible.
- \* A test account with developer privileges is created.

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "No test data needed")

**\*\*Test Steps:\*\*** (List steps for executing the test case)

**\*\*Postconditions:\*\***

- \* No changes to the platform should be made during the test.

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Priority:\*\*** (High, Medium, Low)

**\*\*Type of Testing:\*\*** API Testing

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Cases:\*\***

**\*\*TC\_API\_01\*\***

**\*\*Test Case Title:\*\*** Verify that APIs are available for key platform functionalities.

**\*\*Test Case Description:\*\*** This test verifies the existence of APIs for core functionalities of the platform, such as retrieving course details, creating users, and managing quizzes.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Access the API documentation for the platform.
2. Verify the existence of APIs for the following functionalities:
  - \* Retrieving course details (by ID or name)
  - \* Creating new user accounts
  - \* Accessing and managing quizzes
  - \* Updating user profile information

**\*\*Expected Result:\*\*** All specified APIs are documented and available for use.



**\*\*Severity:\*\* Blocker**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* API Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*TC\_API\_02\*\***

**\*\*Test Case Title:\*\* Verify that comprehensive API documentation is provided.**

**\*\*Test Case Description:\*\* This test evaluates the quality and completeness of the provided API documentation.**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Access the API documentation for the platform.
2. Verify that the documentation provides clear and detailed information on:
  - \* API endpoints and methods
  - \* Request and response parameters
  - \* Data formats (JSON, XML, etc.)
  - \* Error codes and handling
  - \* Authentication and authorization mechanisms
  - \* Rate limiting and usage guidelines

**\*\*Expected Result:\*\*** The API documentation is comprehensive, well-organized, and readily understandable.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** API Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*TC\_API\_03\*\***

**\*\*Test Case Title:\*\*** Verify that secure access protocols are implemented for API endpoints.

**\*\*Test Case Description:\*\*** This test ensures that the platform utilizes secure access protocols like HTTPS to protect sensitive data exchanged through APIs.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Access any API endpoint through a browser or API testing tool.
2. Verify that the connection is established via HTTPS (Secure Sockets Layer).

**\*\*Expected Result:\*\*** All API endpoints are accessed through HTTPS, ensuring secure communication.

**\*\*Severity:\*\* Blocker**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* API Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*TC\_API\_04\*\***

**\*\*Test Case Title:\*\* Verify that third-party applications can authenticate and access APIs.**

**\*\*Test Case Description:\*\* This test simulates a third-party application attempting to authenticate and access platform APIs using provided credentials.**

**\*\*Test Data:\*\***

**\* Test application with authentication credentials.**

**\*\*Test Steps:\*\***

1. Using the test application, attempt to authenticate with the platform using provided credentials.
2. Once authenticated, make a request to a specific API endpoint (e.g., retrieve user details).

**\*\*Expected Result:\*\* The test application successfully authenticates and accesses the requested API endpoint, returning the expected data.**

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* API Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*TC\_API\_05\*\***

**\*\*Test Case Title:\*\* Verify that rate limiting and monitoring features are in place for API access.**

**\*\*Test Case Description:\*\* This test checks if rate limiting mechanisms are in place to prevent excessive API usage and monitors API calls to detect potential abuse.**

**\*\*Test Data:\*\***

**\* Test application for making repeated API calls.**

**\*\*Test Steps:\*\***

1. Use the test application to make a series of requests to an API endpoint within a short time frame.
2. Monitor the platform's response to these requests.

**\*\*Expected Result:\*\* The platform implements rate limiting mechanisms, blocking or slowing down excessive API calls. Additionally, monitoring features should track API usage patterns to detect potential abuse.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* API Testing**

**\*\*Test Case Approach:\*\* 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:

## Test Cases for Data Analytics and Insights Feature

\*\*Project:\*\* Online Learning Platform

\*\*User Story:\*\* As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

\*\*Test Case 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-001	Verify that data visualization tools are available for analysis	This test case verifies that the platform offers various data visualization tools to analyze learner performance data.	Data Analytics	High	Login as an administrator with access to analytics.	No test data needed	1. Navigate to the analytics section of the platform. 2. Identify and explore the available data visualization tools (e.g., charts, graphs, dashboards). 3. Verify the ability to select different data sets and filter results.	No postconditions needed	Data visualization tools are readily available, allowing the administrator to choose from various chart types and manipulate data for insights.	Major	High	Functional Testing	Positive
TC-002	Verify that reports can be generated for learner performance metrics	This test case checks the platform's ability to generate reports based on learner performance metrics.	Reports & Analytics	High	Login as an administrator with access to analytics.	Learner performance data (course completion rates, quiz scores, engagement levels, etc.)	1. Navigate to the reports section of the platform. 2. Select "Learner Performance" as the report type. 3. Choose specific metrics for the report, such as average score, time spent on						

courses, and completion rates. 4. Generate the report and verify its content and accuracy. | No postconditions needed | The platform successfully generates reports with selected performance metrics and data, displaying them in a clear and organized manner. | Major | High | Functional Testing | Positive |

| TC-003 | Verify real-time analytics provide insights into engagement and outcomes | This test case checks the platform's ability to provide real-time insights into learner engagement and course completion outcomes. | Real-Time Analytics | High | Login as an administrator with access to analytics. | No test data needed | 1. Access the real-time analytics dashboard. 2. Observe real-time updates on learner activity, such as active users, ongoing sessions, and course progress. 3. Verify the presence of visualizations and indicators reflecting engagement levels and course completion status. | No postconditions needed | Real-time analytics dashboard displays up-to-date data on engagement and outcomes, enabling informed decisions based on real-time insights. | Major | High | Functional Testing | Positive |

| TC-004 | Verify data can be exported for further analysis | This test case verifies the platform's capability to export data for external analysis. | Data Export | Medium | Login as an administrator with access to analytics. | Learner performance data (course completion rates, quiz scores, engagement levels, etc.) | 1. Navigate to the data analytics section. 2. Select a specific data set or report. 3. Identify and utilize the "Export Data" feature. 4. Verify the successful download of the data in the chosen format (e.g., CSV, Excel, PDF). | No postconditions needed | The platform successfully exports data in the selected format, allowing for further analysis outside of the platform. | Minor | Medium | Functional Testing | Positive |

| TC-005 | Verify notifications are sent for significant data trends or anomalies | This test case checks the platform's ability to send notifications for significant data trends or anomalies. | Notifications & Alerts | High | Login as an administrator with access to analytics. | No test data needed | 1. Configure notification settings for significant data trends or anomalies. 2. Simulate a scenario that triggers a notification (e.g., a sudden drop in engagement). 3. Verify the delivery of the notification to the administrator's account. | No postconditions needed | The platform sends timely notifications to the administrator when significant data trends or anomalies are detected, alerting them to potential issues and informing them of actionable insights. | Major | High | Functional Testing | Positive |

---

## **Data - 73**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

User Roles and Permissions

### **Feature description -**

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

### **Userstory title -**

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

### **Acceptance criteria -**

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

### **Generated Test Cases:**

## Test Cases for User Roles and Permissions Feature



**\*\*Project:\*\* Online Learning Platform**

**\*\*User Story:\*\*** As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

**\*\*Acceptance Criteria:\*\***

1. Administrators can create and assign user roles.
2. Permissions are customizable for each role.
3. Users are restricted to functionalities based on their roles.
4. Changes to roles and permissions are logged.
5. Users are notified of their assigned roles and permissions.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-URP-01

**\*\*Test Case Title:\*\*** Verify that administrators can create new user roles.

**\*\*Test Case Description:\*\*** This test case verifies that administrators have the ability to create new user roles with unique names and descriptions.

**\*\*Test Suite:\*\*** User Roles and Permissions

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* An administrator account is logged in.

**\*\*Test Data:\*\***

\* New role name: "Content Creator"

\* New role description: "Users with access to create and manage course content."

**\*\*Test Steps:\*\***

1. Navigate to the "User Roles" section in the platform administration panel.
2. Click on the "Create New Role" button.
3. Enter the new role name "Content Creator" in the designated field.
4. Enter the role description "Users with access to create and manage course content" in the corresponding field.
5. Click on the "Save" button to create the new role.

**\*\*Postconditions:\*\***

\* The new "Content Creator" role should be listed in the User Roles section.

**\*\*Expected Result:\*\*** The new role "Content Creator" is successfully created with the specified name and description.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-URP-02

**\*\*Test Case Title:\*\*** Verify that permissions can be customized for each role.

**\*\*Test Case Description:\*\*** This test case verifies that administrators can assign different permissions to different user roles, enabling fine-grained access control.

**\*\*Test Suite:\*\*** User Roles and Permissions

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* An administrator account is logged in.

\* A new user role "Content Creator" exists.

**\*\*Test Data:\*\***

\* Permissions: "Create Course", "Edit Course", "Delete Course"

**\*\*Test Steps:\*\***

1. Navigate to the "User Roles" section in the platform administration panel.
2. Select the "Content Creator" role.
3. Click on the "Permissions" tab.
4. Check the boxes corresponding to "Create Course", "Edit Course", and "Delete Course" permissions.
5. Click on the "Save" button to assign permissions to the role.

**\*\*Postconditions:\*\***

- \* The "Content Creator" role now has the specified permissions.

**\*\*Expected Result:\*\*** The selected permissions are successfully assigned to the "Content Creator" role.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-URP-03

**\*\*Test Case Title:\*\*** Verify that users are restricted to functionalities based on their assigned roles.

**\*\*Test Case Description:\*\*** This test case verifies that users can only access features and functionalities that are permitted by their assigned roles, preventing unauthorized access.

**\*\*Test Suite:\*\*** User Roles and Permissions

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* A "Content Creator" role with "Create Course", "Edit Course", and "Delete Course" permissions exists.
- \* A user is assigned the "Content Creator" role.
- \* A user is assigned the "Student" role with no access to course management functionalities.

**\*\*Test Data:\*\***

- \* No test data needed

**\*\*Test Steps:\*\***

1. Log in as a user with the "Content Creator" role.
2. Attempt to create, edit, and delete courses.
3. Log in as a user with the "Student" role.
4. Attempt to create, edit, and delete courses.

**\*\*Postconditions:\*\***

- \* No postconditions

**\*\*Expected Result:\*\***

- \* The "Content Creator" user should be able to successfully create, edit, and delete courses.
- \* The "Student" user should be prevented from accessing course management functionalities and receive an appropriate error message.

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Negative**

**\*\*Test Case ID:\*\* TC-URP-04**

**\*\*Test Case Title:\*\* Verify that changes to roles and permissions are logged.**

**\*\*Test Case Description:\*\*** This test case verifies that all changes made to user roles and permissions are accurately recorded in a log, providing an audit trail for security and accountability.

**\*\*Test Suite:\*\* User Roles and Permissions**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- \* An administrator account is logged in.
- \* A user role exists with specific permissions.

**\*\*Test Data:\*\***

\* No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the "User Roles" section in the platform administration panel.
2. Select an existing role and modify its permissions.
3. Navigate to the "Audit Logs" section.
4. Filter the logs for changes related to user roles and permissions.

**\*\*Postconditions:\*\***

\* No postconditions

**\*\*Expected Result:\*\*** The log should contain a record of the changes made to the role, including the date, time, administrator who made the changes, and the specific details of the modifications.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** System Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-URP-05

**\*\*Test Case Title:\*\*** Verify that users are notified of their assigned roles and permissions.

**\*\*Test Case Description:\*\*** This test case verifies that users are notified upon being assigned a role or receiving new permissions, ensuring transparency and awareness of their access rights.

**\*\*Test Suite:\*\*** User Roles and Permissions

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* A user account is created.

**\*\*Test Data:\*\***

\* No test data needed

**\*\*Test Steps:\*\***

1. Assign a specific role with permissions to the user account.
2. Log in as the assigned user.
3. Check the user's notification center or inbox for a message regarding their assigned role and permissions.

**\*\*Postconditions:\*\***

\* No postconditions

**\*\*Expected Result:\*\*** The user receives a clear and informative notification regarding their assigned role and permissions, including a summary of the granted access rights.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** System 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:**

## Test Cases for User Feedback and Rating System

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* User Feedback and Rating System**

**\*\*User Story 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:\*\***

1. Learners can rate courses and instructors on a scale.
2. Feedback forms are available for detailed comments.
3. Instructors can view and respond to feedback.
4. Aggregated ratings are displayed publicly.
5. Anonymous feedback options are available.

**\*\*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-01 | Verify that learners can rate courses and instructors on a scale. | This test case verifies that learners can submit ratings for courses and instructors using a defined scale (e.g., 1 to 5 stars). | Feedback and Rating System | High | User is logged in and enrolled in a course. | Course ID: "Introduction to Programming", Instructor ID: "Dr. Smith" | 1. Navigate to the course page. 2. Locate the rating section for the course. 3. Select a rating from the provided scale (e.g., 4 stars). 4. Submit the rating. | Rating is saved and displayed on the course page. | The user's rating is successfully saved and displayed on the course page. | Major | High | Functional Testing | Positive |

| TC-02 | Verify that feedback forms are available for detailed comments. | This test case verifies that a feedback form is available for learners to provide detailed comments on courses, instructors, and learning materials. | Feedback and Rating System | High | User is logged in and enrolled in a course. | No test data needed | 1. Navigate to the course page. 2. Locate the feedback section. 3. Click on the "Leave Feedback" button. 4. Verify that a feedback form appears with fields for detailed comments. | The feedback form is displayed with appropriate fields for providing detailed comments. | Major | High | Functional Testing | Positive |

| TC-03 | Verify that instructors can view and respond to feedback. | This test case verifies that instructors can access and view feedback provided by learners, and they have the ability to respond to the feedback. | Feedback and Rating System | High | Instructor is logged in. | Course ID: "Introduction to Programming", Instructor ID: "Dr. Smith", Feedback: "The course was well-structured and engaging." | 1. Navigate to the instructor dashboard. 2. Access the feedback section for the "Introduction to Programming" course. 3. Verify that learner feedback is displayed. 4. Compose a response to the feedback. 5. Submit the response. | The instructor can view the feedback, compose and submit a response. | Major | High | Functional Testing | Positive |

| TC-04 | Verify that aggregated ratings are displayed publicly. | This test case verifies that the platform



displays the aggregated average rating for courses and instructors publicly. | Feedback and Rating System | High | User is logged in and browsing courses. | No test data needed | 1. Browse courses on the platform. 2. Verify that each course displays an aggregated average rating (e.g., 4.5 stars). 3. Navigate to the instructor profile. 4. Verify that the instructor profile displays an aggregated average rating. | The average rating for each course and instructor is displayed publicly. | Major | High | Functional Testing | Positive |

| TC-05 | Verify that anonymous feedback options are available. | This test case verifies that learners can choose to provide anonymous feedback, ensuring their identity is not linked to the feedback. | Feedback and Rating System | High | User is logged in and enrolled in a course. | No test data needed | 1. Navigate to the course page. 2. Locate the feedback section. 3. Verify that an option exists to submit feedback anonymously. 4. Select the anonymous option and submit feedback. | The feedback is submitted without disclosing the user's identity. | Major | High | Functional Testing | Positive |

**\*\*Note:\*\*** These test cases cover the basic functionalities of the User Feedback and Rating System. Additional test cases can be created for edge cases, negative scenarios, and different user roles (e.g., administrator, student, instructor).

-----

## **Data - 75**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Social Media Integration

## **Feature description -**

The platform supports social media integration, allowing learners and instructors to share their achievements, courses, and updates on social media platforms. This feature promotes engagement and community building, enhancing the visibility of the platform's offerings.

## **Userstory title -**

As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

## **Acceptance criteria -**

['Learners can connect their social media accounts to the platform.', 'Achievements and course completions can be shared on social media.', 'Privacy settings control the visibility of shared content.', 'Instructors can promote courses and updates through social media.', 'Analytics track the reach and engagement of shared content.']

## **Generated Test Cases:**

## Test Cases for Social Media Integration Feature

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Social Media Integration**

**\*\*User Story Title:\*\* As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.**

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC-001**

**\*\*Test Case Title:\*\* Verify that learners can connect their social media accounts to the platform.**

**\*\*Test Case Description:\*\* This test case validates the functionality of connecting social media accounts (Facebook, Twitter, LinkedIn) to the online learning platform. It ensures a smooth connection process, data**

privacy, and secure authorization.

**\*\*Test Suite:\*\*** Social Media Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* The learner has an active social media account.

- \* The learner has registered on the online learning platform.

- \* The online learning platform supports social media integration with the chosen platform.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Login to the online learning platform as a learner.

2. Navigate to the user profile settings or social media integration section.

3. Select the desired social media platform (Facebook, Twitter, LinkedIn) for connection.

4. Follow the platform-specific authorization prompts (login, grant access).

5. After successful authorization, verify if the account is connected to the platform.

**\*\*Postconditions:\*\***

- \* The learner's social media account is successfully connected to the platform.

- \* The learner should be able to see the connected account in their profile settings.

- \* The learner should have the option to disconnect the account later.

**\*\*Expected Result:\*\*** The social media account is successfully connected to the platform with appropriate permissions.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-002

**\*\*Test Case Title:\*\*** Verify that achievements and course completions can be shared on social media.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of sharing achievements and course completions on social media. It ensures that learners can share their accomplishments with their network and promote their learning progress.

**\*\*Test Suite:\*\*** Social Media Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* The learner has a connected social media account.
- \* The learner has completed a course or achieved a milestone within the platform.
- \* The social media account has appropriate privacy settings for sharing.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Login to the online learning platform as a learner.
2. Navigate to the completed course or achievement details page.
3. Locate the option to share the achievement/completion on social media.
4. Select the desired social media platform.
5. Compose and share the post, including a relevant message and link.
6. Verify the shared post on the chosen social media platform.

**\*\*Postconditions:\*\***

- \* The post is successfully shared on the chosen social media platform.
- \* The shared post includes the course name, achievement description, or any relevant information.
- \* The shared post is visible according to the learner's social media privacy settings.

**\*\*Expected Result:\*\*** The achievement/completion is successfully shared on the chosen social media platform with the appropriate content.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\* TC-003**

**\*\*Test Case Title:\*\* Verify that privacy settings control the visibility of shared content.**

**\*\*Test Case Description:\*\* This test case ensures that learners have control over the privacy settings of their shared achievements and course completions on social media. It ensures the platform's compliance with data privacy regulations and user preferences.**

**\*\*Test Suite:\*\* Social Media Integration**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- \* The learner has a connected social media account.
- \* The learner has shared an achievement or course completion on social media.
- \* The social media account has different privacy settings available.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Login to the online learning platform as a learner.
2. Navigate to the social media integration settings or profile settings.
3. Modify the privacy settings for shared content.
4. Share a new achievement or course completion on the platform.
5. Verify the visibility of the shared content on the chosen social media platform based on the selected privacy settings.

**\*\*Postconditions:\*\***

- \* The shared content's visibility on the chosen social media platform reflects the selected privacy settings.
- \* The learner can access and modify privacy settings anytime.

**\*\*Expected Result:\*\* Shared content visibility on social media matches the selected privacy settings.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\*** Positive and Negative

**\*\*Test Case ID:\*\*** TC-004

**\*\*Test Case Title:\*\*** Verify that instructors can promote courses and updates through social media.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of instructors promoting courses and updates on social media. It ensures instructors can effectively reach potential learners and engage with the platform's community.

**\*\*Test Suite:\*\*** Social Media Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* The instructor has a connected social media account.
- \* The instructor has created a course or updated existing course content on the platform.
- \* The social media account has appropriate permissions for posting.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Login to the online learning platform as an instructor.
2. Navigate to the course details page or the platform's newsfeed/announcement section.
3. Locate the option to share the course/update on social media.
4. Select the desired social media platform.
5. Compose and share the post, including relevant information, links, and visuals.
6. Verify the shared post on the chosen social media platform.

**\*\*Postconditions:\*\***

- \* The course/update is successfully shared on the chosen social media platform.
- \* The shared post includes relevant information like course title, description, link, or key updates.
- \* The post is visible according to the instructor's social media privacy settings.

**\*\*Expected Result:\*\*** The course/update is successfully shared on the chosen social media platform with the appropriate content.

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-005**

**\*\*Test Case Title:\*\* Verify that analytics track the reach and engagement of shared content.**

**\*\*Test Case Description:\*\* This test case verifies the tracking and reporting of social media engagement for shared content. It ensures the platform provides valuable insights into the effectiveness of social media promotions and audience reach.**

**\*\*Test Suite:\*\* Social Media Integration**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- \* The learner/instructor has shared content on social media through the platform.
- \* The platform has analytics functionality for social media sharing.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Login to the online learning platform as a learner or instructor.
2. Navigate to the relevant section for analytics or social media insights.
3. View the data related to shared content, including reach, impressions, likes, shares, comments, or other relevant metrics.
4. Analyze the data and verify its accuracy and usefulness.

**\*\*Postconditions:\*\***

- \* The platform provides detailed analytics on shared content engagement.
- \* The analytics data is accurate and comprehensive, providing actionable insights.

**\*\*Expected Result:\*\* The platform accurately tracks and reports on the reach and engagement of shared content on social media.**

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing & Non-Functional Testing (Analytics)

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 76**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Activity Tracking

### **Feature description -**

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity levels and progress towards fitness goals.

### **Userstory title -**

As a user, I want to track my activities so that I can monitor my daily physical activity levels.

### **Acceptance criteria -**

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']



**Generated Test Cases:**

**## Test Cases for Fitness Tracker App - Activity Tracking**

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*User Story:\*\* As a user, I want to track my activities so that I can monitor my daily physical activity levels.**

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\* Activity Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the app.
- App has permission to access location data (if applicable).
- Activity tracking is enabled.

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_AT\_01

**\*\*Test Case Title:\*\*** Verify that the app records data on distance, duration, pace, and calories burned.

**\*\*Test Case Description:\*\*** This test case verifies that the app accurately captures and displays distance, duration, pace, and calories burned during an activity.

**\*\*Test Data:\*\***

- Activity Type: Running

- Distance: 5 km

- Duration: 30 minutes

- Pace: 6 km/h

**\*\*Test Steps:\*\***

1. Start an activity tracking session for "Running".
2. Run for 30 minutes covering a distance of 5 km.
3. End the activity tracking session.
4. View the activity summary.

**\*\*Postconditions:\*\***

- Activity tracking session is stopped.
- Activity summary is displayed.

**\*\*Expected Result:\*\***

- The activity summary should display the following data:
  - Distance: 5 km
  - Duration: 30 minutes
  - Pace: 6 km/h
  - Calories Burned: (Accurate calculation based on user profile and activity intensity)

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_AT\_02

**\*\*Test Case Title:\*\*** Verify that users can view activity summaries.

**\*\*Test Case Description:\*\*** This test case validates that users can access and view detailed information about their past activities.

**\*\*Test Data:\*\***

- Activity Type: Walking
- Distance: 3 km
- Duration: 45 minutes
- Pace: 4 km/h

**\*\*Test Steps:\*\***

1. Start an activity tracking session for "Walking".
2. Walk for 45 minutes covering a distance of 3 km.
3. End the activity tracking session.
4. Navigate to the "Activity History" section.
5. Select the completed walking activity.

**\*\*Postconditions:\*\***

- Activity summary is displayed.

**\*\*Expected Result:\*\***

- The activity summary should display all recorded data (distance, duration, pace, calories burned, date, time, etc.).

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC\_AT\_03**

**\*\*Test Case Title:\*\* Verify that data is stored for historical analysis.**

**\*\*Test Case Description:\*\* This test case ensures that the app stores activity data for future reference and analysis.**

**\*\*Test Data:\*\***

- Activity Type: Cycling
- Distance: 10 km
- Duration: 1 hour
- Pace: 10 km/h

**\*\*Test Steps:\*\***

1. Start an activity tracking session for "Cycling".
2. Cycle for 1 hour covering a distance of 10 km.
3. End the activity tracking session.
4. Log out of the app.
5. Log back into the app.
6. Navigate to the "Activity History" section.

**\*\*Postconditions:\*\***

- Activity summary is displayed.

**\*\*Expected Result:\*\***

- The previously recorded cycling activity should be available for viewing in the "Activity History" section.

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_AT\_04

**\*\*Test Case Title:\*\*** Verify that activity tracking can be paused and resumed.

**\*\*Test Case Description:\*\*** This test case ensures that users can pause an ongoing activity tracking session and resume it later.

**\*\*Test Data:\*\***

- Activity Type: Swimming

- Distance: 1 km

- Duration: 30 minutes

- Pace: 2 km/h

**\*\*Test Steps:\*\***

1. Start an activity tracking session for "Swimming".
2. Swim for 15 minutes covering a distance of 500 meters.
3. Pause the activity tracking session.
4. Resume the activity tracking session.
5. Swim for another 15 minutes covering a distance of 500 meters.
6. End the activity tracking session.

**\*\*Postconditions:\*\***

- Activity tracking session is stopped.

- Activity summary is displayed.

**\*\*Expected Result:\*\***

- The activity summary should display the total distance (1 km) and duration (30 minutes).

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_AT\_05

**\*\*Test Case Title:\*\*** Verify that users receive notifications for milestones.

**\*\*Test Case Description:\*\*** This test case ensures that users receive notifications when they achieve predefined milestones (e.g., reaching a certain distance or duration).

**\*\*Test Data:\*\***

- Activity Type: Running
- Distance: 10 km
- Duration: 1 hour
- Pace: 10 km/h
- Milestone: Running 10 km

**\*\*Test Steps:\*\***

1. Set a milestone notification for running 10 km.
2. Start an activity tracking session for "Running".
3. Run for 1 hour covering a distance of 10 km.

**\*\*Postconditions:\*\***

- Activity tracking session is stopped.
- Activity summary is displayed.

**\*\*Expected Result:\*\***

- A notification should be displayed in the app and/or on the user's device indicating that they have achieved the "10 km running" milestone.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

-----

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

GPS Tracking

**Feature description -**

Integrated GPS tracking allows users to accurately track outdoor activities and routes, providing detailed maps, elevation profiles, and location-based metrics. Users can visualize their workout routes, analyze performance data, and explore new routes for added variety and challenge.

**Userstory title -**

As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

**Acceptance criteria -**

['GPS tracking accurately records route data.', 'Users can view detailed maps and elevation profiles.', 'Performance data includes location-based metrics.', 'Routes can be saved and revisited.', 'Notifications are provided for deviations from planned routes.']

**Generated Test Cases:**

## Fitness Tracker App - GPS Tracking Test Cases

**\*\*Project Context:\*\***

**\* \*\*Project Name:\*\*** Fitness Tracker App

**\* \*\*Project Description:\*\*** A mobile application allowing users to track their physical activity, exercise

routines, and health metrics.

\* \*\*Feature Name:\*\* GPS Tracking

\* \*\*Feature Description:\*\* Integrated GPS tracking allowing users to accurately track outdoor activities and routes, providing detailed maps, elevation profiles, and location-based metrics.

\* \*\*User Story Title:\*\* As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

\*\*Acceptance Criteria:\*\*

- 1. \*\*GPS tracking accurately records route data.\*\*
- 2. \*\*Users can view detailed maps and elevation profiles.\*\*
- 3. \*\*Performance data includes location-based metrics.\*\*
- 4. \*\*Routes can be saved and revisited.\*\*
- 5. \*\*Notifications are provided for deviations from planned routes.\*\*

\*\*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-GPS-01	Verify that GPS tracking accurately records route data.	This test case verifies that the GPS tracking feature accurately records the user's route data during an outdoor activity, capturing start and end points, distance, and time spent.	GPS Tracking	High	User is logged in and GPS is enabled.	A known route of a specific distance and duration.	1. Start a new outdoor activity using the GPS tracking feature. 2. Perform the activity along the predefined route. 3. Stop the activity and review the recorded route data.	Route data is displayed accurately in the app.	Start and end points, distance, and time spent match the actual route.	Major	High	Functional Testing	Positive
-----------	---	--	--------------	------	---------------------------------------	--	--	--	--	-------	------	--------------------	----------



| TC-GPS-02 | Verify that users can view detailed maps and elevation profiles. | This test case verifies that users can view detailed maps and elevation profiles of their recorded routes. | GPS Tracking | Medium | A route has been recorded using GPS tracking. | No test data needed. | 1. Select a recorded route from the activity history. 2. View the map of the recorded route. 3. Verify the presence of elevation profile information. | The map displays the correct route and elevation profile information is available. | The map displays the route accurately, including all major turns and points of interest. Elevation profile displays accurate elevation changes. | Minor | Medium | Functional Testing | Positive |

| TC-GPS-03 | Verify that performance data includes location-based metrics. | This test case verifies that the recorded performance data includes location-based metrics such as pace, speed, and altitude. | GPS Tracking | Medium | A route has been recorded using GPS tracking. | No test data needed. | 1. Select a recorded route from the activity history. 2. View the performance data for the route. 3. Verify the presence of location-based metrics. | The performance data includes location-based metrics. | Pace, speed, and altitude information is displayed accurately. | Minor | Medium | Functional Testing | Positive |

| TC-GPS-04 | Verify that routes can be saved and revisited. | This test case verifies that users can save their completed routes and revisit them for future reference or use as a planned route. | GPS Tracking | Medium | User is logged in and GPS is enabled. | A previously recorded route. | 1. Select a recorded route from the activity history. 2. Save the route. 3. Access the saved routes list. 4. Select the saved route and verify its details. | The route is saved successfully and can be revisited from the saved routes list. | The saved route is accessible, and its details (map, performance data, etc.) are displayed correctly. | Minor | Medium | Functional Testing | Positive |

| TC-GPS-05 | Verify that notifications are provided for deviations from planned routes. | This test case verifies that the app provides notifications to users when they deviate from a planned route during an activity. | GPS Tracking | High | A planned route has been selected and an activity using GPS tracking is started. | A pre-defined planned route with a clear deviation point. | 1. Select a planned route for an activity. 2. Start the activity using GPS tracking. 3. Deviate from the planned route at a designated point. | A notification is triggered when deviating from the planned route. | The notification informs the user about the deviation and guides them back to the planned route. | Major | High | Functional Testing | Negative |

---

## **Data - 78**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Heart Rate Monitoring

### **Feature description -**

The app monitors heart rate in real-time using built-in sensors or compatible wearable devices, providing insights into exercise intensity, recovery, and cardiovascular health. Heart rate data helps users optimize their workouts, maintain target heart rate zones, and track improvements in fitness over time.

### **Userstory title -**

As a user, I want to monitor my heart rate so that I can optimize my workouts and track my cardiovascular health.

### **Acceptance criteria -**

['Heart rate data is recorded in real-time.', 'Users can set and monitor target heart rate zones.', 'Historical heart rate data is stored and viewable.', 'Alerts are provided for abnormal heart rate readings.', 'Data is compatible with various wearable devices.']

### **Generated Test Cases:**

## Test Cases for Heart Rate Monitoring Feature:

**\*\*Test Case ID:\*\* TC\_HR\_01**

**\*\*Test Case Title:\*\* Verify that heart rate data is recorded in real-time.**

**\*\*Test Case Description:\*\* This test case aims to validate that the app accurately captures and displays real-time heart rate data during an exercise session.**

**\*\*Test Suite:\*\* Heart Rate Monitoring**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged into the app.

\* User has granted permission for the app to access heart rate data from the device or wearable.

\* User has started an exercise session.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Start an exercise session in the app.

2. Observe the heart rate display in the app.

3. Monitor the heart rate display for 1 minute.

4. Compare the displayed heart rate values with the actual heart rate, measured using a separate heart rate monitor (e.g., smartwatch or chest strap).

**\*\*Postconditions:\*\***

\* End the exercise session in the app.

**\*\*Expected Result:\*\* The displayed heart rate in the app should closely match the actual heart rate readings from the separate heart rate monitor.**

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing, Integration Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_HR\_02**

**\*\*Test Case Title:\*\*** Verify that users can set and monitor target heart rate zones.

**\*\*Test Case Description:\*\*** This test case verifies that users can define their desired heart rate zones and monitor their progress within those zones during an exercise session.

**\*\*Test Suite:\*\*** Heart Rate Monitoring

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged into the app.

\* User has completed a previous exercise session with recorded heart rate data.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Access the heart rate settings within the app.
2. Set target heart rate zones for different activity levels (e.g., Fat Burning, Cardio, Peak Performance).
3. Start an exercise session.
4. Observe the displayed heart rate and its location within the set target zones during the session.

**\*\*Postconditions:\*\***

\* End the exercise session in the app.

**\*\*Expected Result:\*\*** The app accurately displays the user's current heart rate in relation to the set target zones, providing visual feedback on exercise intensity.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing, User Interface Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_HR\_03

**\*\*Test Case Title:\*\*** Verify that historical heart rate data is stored and viewable.

**\*\*Test Case Description:\*\*** This test case ensures that the app stores past heart rate data from exercise sessions and allows users to access and analyze this data.

**\*\*Test Suite:\*\* Heart Rate Monitoring**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* User is logged into the app.

\* User has completed at least 3 exercise sessions with recorded heart rate data.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Access the history or log section within the app.
2. Select "Heart Rate" or a similar option to view past heart rate data.
3. Browse through the recorded heart rate data from past exercise sessions.
4. Check the availability of details like date, time, duration, average heart rate, and heart rate graph.

**\*\*Postconditions:\*\***

\* No postconditions.

**\*\*Expected Result:\*\*** The app displays a clear and organized history of heart rate data, including relevant information about each exercise session.

**\*\*Severity:\*\* Minor**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing, User Interface Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_HR\_04**

**\*\*Test Case Title:\*\* Verify that alerts are provided for abnormal heart rate readings.**

**\*\*Test Case Description:\*\*** This test case tests whether the app alerts the user if their heart rate exceeds a predefined threshold or falls below a healthy range during an exercise session.

**\*\*Test Suite:\*\* Heart Rate Monitoring**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged into the app.

\* User has started an exercise session.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. During an exercise session, manually increase the heart rate beyond the pre-defined threshold (e.g., 200 bpm).

2. Monitor the app for any alerts or notifications.

3. Repeat step 1 but manually decrease the heart rate below a healthy range (e.g., 40 bpm).

4. Monitor the app for any alerts or notifications.

**\*\*Postconditions:\*\***

\* End the exercise session in the app.

**\*\*Expected Result:\*\*** The app should display a clear and informative alert when the heart rate exceeds the upper threshold or drops below the lower threshold.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing, User Interface Testing

**\*\*Test Case Approach:\*\*** Negative

**\*\*Test Case ID:\*\*** TC\_HR\_05

**\*\*Test Case Title:\*\*** Verify that data is compatible with various wearable devices.

**\*\*Test Case Description:\*\*** This test case evaluates the app's ability to connect and receive heart rate data from different compatible wearable devices.

**\*\*Test Suite:\*\*** Heart Rate Monitoring

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged into the app.

\* User has at least two different compatible wearable devices (e.g., smartwatch, chest strap).

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Connect the first wearable device to the app.
2. Start an exercise session and monitor the heart rate data received from the device.
3. Disconnect the first device and connect the second wearable device to the app.
4. Start another exercise session and monitor the heart rate data received from the second device.

**\*\*Postconditions:\*\***

\* Disconnect both wearable devices from the app.

**\*\*Expected Result:\*\*** The app successfully connects to both wearable devices, accurately receives heart rate data, and displays it correctly during exercise sessions.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Compatibility Testing, Integration Testing

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 79**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Calorie Tracking

## **Feature description -**

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

## **Userstory title -**

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

## **Acceptance criteria -**

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

## **Generated Test Cases:**

## Test Cases for Calorie Tracking Feature - Fitness Tracker App

**\*\*Project:\*\*** Fitness Tracker App

**\*\*Feature:\*\*** Calorie Tracking

**\*\*User Story:\*\*** 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:\*\***

1. Users can log daily food intake and calorie consumption.
2. The app calculates and displays total calories consumed and burned.
3. Users can set daily calorie goals and track progress.
4. Food items and calorie information are searchable in the database.
5. Users receive alerts for exceeding calorie goals.



**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Calorie Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User has an active account in the Fitness Tracker App.

\* The user has logged in to the app.

\* The app has access to the user's daily activity data.

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_CT\_01

**\*\*Test Case Title:\*\*** Verify that users can log daily food intake and calorie consumption.

**\*\*Test Case Description:\*\*** This test case checks the user's ability to log food items and their corresponding calorie values.

**\*\*Test Data:\*\***

**\* Sample Food Items:**

- \* Apple (100 calories)
- \* Chicken Breast (200 calories)
- \* Salad (150 calories)

**\*\*Test Steps:\*\***

1. Open the Calorie Tracker section of the app.
2. Click on the "Add Food" button.
3. Search for the food item "Apple" and select it.
4. Enter the quantity consumed (e.g., 1).
5. Click on "Add to Diary".
6. Repeat steps 2-5 for "Chicken Breast" and "Salad".
7. Verify that the food items and their calorie values are correctly added to the daily intake log.

**\*\*Postconditions:\*\***

- \* The daily food log should be updated with the added items and their calorie values.

**\*\*Expected Result:\*\*** The app should successfully record the user's food intake and calculate the total calories consumed.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_CT\_02

**\*\*Test Case Title:\*\*** Verify that the app calculates and displays total calories consumed and burned.

**\*\*Test Case Description:\*\*** This test case verifies that the app accurately calculates and displays the total calories consumed and burned based on the user's food log and activity data.

**\*\*Test Data:\*\***

\* Calorie Intake: 500 calories (from test case 1)

\* Calorie Burned: 300 calories (from user activity data)

**\*\*Test Steps:\*\***

1. Open the Calorie Tracker section of the app.
2. Navigate to the "Daily Summary" view.
3. Verify that the "Total Calories Consumed" value is 500.
4. Verify that the "Total Calories Burned" value is 300.
5. Verify that the "Calorie Balance" value is 200 (consumed - burned).

**\*\*Postconditions:\*\***

\* The daily calorie summary should be displayed correctly based on the user's data.

**\*\*Expected Result:\*\*** The app should accurately display the total calories consumed, burned, and the calorie balance for the day.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_CT\_03

**\*\*Test Case Title:\*\*** Verify that users can set daily calorie goals and track progress.

**\*\*Test Case Description:\*\*** This test case verifies the ability of the user to set daily calorie goals and monitor their progress toward achieving those goals.

**\*\*Test Data:\*\***

\* User's current calorie intake: 500 calories

\* Desired calorie goal: 700 calories

**\*\*Test Steps:\*\***

1. Open the Calorie Tracker section of the app.
2. Navigate to the "Goal Setting" section.
3. Enter the desired calorie goal (700 calories).
4. Save the goal setting.
5. Verify that the calorie goal is displayed in the daily summary view.
6. Verify that the progress bar accurately reflects the user's progress towards the goal (500/700).

**\*\*Postconditions:\*\***

\* The daily calorie goal should be set and saved.

\* The progress bar should accurately reflect the user's progress toward the goal.

**\*\*Expected Result:\*\*** The app should allow users to set daily calorie goals and track their progress towards achieving them.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_CT\_04

**\*\*Test Case Title:\*\*** Verify that food items and calorie information are searchable in the database.

**\*\*Test Case Description:\*\*** This test case ensures that users can easily search for food items and retrieve their calorie information.

**\*\*Test Data:\*\***

\* Search term: "Chicken"

\* Expected results: Various chicken dishes with their corresponding calorie values.

**\*\*Test Steps:\*\***

1. Open the Calorie Tracker section of the app.
2. Click on the "Search Food" option.
3. Enter the search term "Chicken".
4. Verify that the app displays a list of relevant food items containing "Chicken".
5. Verify that the calorie information is displayed alongside each search result.

**\*\*Postconditions:\*\***

\* The app should display a relevant list of food items matching the search term.

**\*\*Expected Result:\*\*** The app should provide an efficient and accurate search functionality to help users find food items and their calorie information.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_CT\_05

**\*\*Test Case Title:\*\*** Verify that users receive alerts for exceeding calorie goals.

**\*\*Test Case Description:\*\*** This test case checks if the app alerts the user when their calorie intake exceeds their daily goal.

**\*\*Test Data:\*\***

\* User's daily calorie goal: 700 calories

\* Current calorie intake: 800 calories

**\*\*Test Steps:\*\***

1. Open the Calorie Tracker section of the app.
2. Manually add food items to exceed the user's daily goal of 700 calories.
3. Verify that the app displays an alert or notification indicating that the user has exceeded their calorie goal.
4. Verify that the alert provides relevant information like the current calorie intake and the remaining calories allowed.

**\*\*Postconditions:\*\***

- \* The user should be notified of exceeding the calorie goal.
- \* The notification should be displayed in a user-friendly manner.

**\*\*Expected Result:\*\*** The app should provide timely alerts to users when they exceed their set calorie goals.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Negative

-----

**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:

## Test Cases for Fitness Tracker App - Sleep Tracking Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*User Story Title:\*\* As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.**

**\*\*Acceptance Criteria:\*\***

1. Sleep data is recorded including duration and quality.
2. Users can view detailed sleep stages and patterns.
3. The app provides insights into sleep disturbances.
4. Users receive recommendations for improving sleep quality.

5. Sleep data can be viewed in daily, weekly, and monthly summaries.

### \*\*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_SLEEP_01	Verify that sleep duration and quality are recorded accurately.	This test case verifies that the app correctly records sleep duration and quality based on data from the accelerometer and heart rate sensors. The test involves simulating a sleep session with known duration and quality, and then comparing the recorded data to the actual values.	Sleep Tracking	High	User has enabled sleep tracking	Simulated sleep session data (duration, heart rate patterns)	<ol style="list-style-type: none"> <li>1. Start a simulated sleep session with specific duration and heart rate patterns.</li> <li>2. End the simulated sleep session.</li> <li>3. Navigate to the sleep tracking section of the app.</li> <li>4. Verify that the recorded sleep duration and quality match the simulated data.</li> </ol>	Sleep session data is cleared after test execution	Recorded sleep duration and quality match the simulated data within an acceptable tolerance.	Major	High	Functional Testing	Positive
-------------	---	---	----------------	------	---------------------------------	--	--	--	--	-------	------	--------------------	----------

TC_SLEEP_02	Verify that users can view detailed sleep stages and patterns.	This test case verifies that the app displays detailed information about sleep stages (e.g., light sleep, deep sleep, REM sleep) and patterns, allowing users to understand their sleep cycles. The test involves examining the sleep data visualization and confirming that it presents accurate and comprehensive information.	Sleep Tracking	High	User has completed a sleep session.	Sleep data from a completed session	<ol style="list-style-type: none"> <li>1. Navigate to the sleep tracking section of the app.</li> <li>2. Select a specific sleep session from the history.</li> <li>3. Verify that the sleep data visualization displays detailed information on sleep stages and patterns, including the time spent in each stage and the overall sleep cycle.</li> </ol>	No specific postconditions	The sleep data visualization accurately presents sleep stages and patterns for the selected sleep session.	Major	High	Functional Testing	Positive
-------------	--	--	----------------	------	-------------------------------------	-------------------------------------	--	----------------------------	--	-------	------	--------------------	----------

TC_SLEEP_03	Verify that the app provides insights into sleep disturbances.	This test case verifies that the app identifies and reports any sleep disturbances during a session, such as wake-ups, movement, or changes in
-------------	--	--



heart rate. The test involves simulating a sleep session with known disturbances and then checking if the app identifies and presents this information to the user. | Sleep Tracking | High | User has completed a sleep session with simulated disturbances | Simulated sleep data with known disturbances (e.g., wake-ups, movement) | 1. Start a simulated sleep session with specific disturbances. 2. End the simulated sleep session. 3. Navigate to the sleep tracking section and view the selected sleep session. 4. Verify that the app identifies and presents information about the simulated disturbances. | Sleep session data is cleared after test execution | The app accurately identifies and displays the simulated sleep disturbances within the selected sleep session. | Major | High | Functional Testing | Positive |

| TC\_SLEEP\_04 | Verify that the app provides recommendations for improving sleep quality. | This test case verifies that the app offers personalized recommendations based on the user's sleep data and patterns. The test involves analyzing the recommendations provided by the app for various sleep scenarios and ensuring they are appropriate and relevant to the user's situation. | Sleep Tracking | Medium | User has completed a sleep session. | Sleep data from a completed session | 1. Navigate to the sleep tracking section of the app. 2. Select a specific sleep session from the history. 3. Check if the app provides recommendations for improving sleep quality based on the sleep data. 4. Analyze the recommendations to ensure they are relevant to the sleep patterns and disturbances observed in the session. | No specific postconditions | The app provides personalized recommendations for improving sleep quality based on the sleep data, covering aspects like bedtime routines, sleep environment, and lifestyle choices. | Minor | Medium | Functional Testing | Positive |

| TC\_SLEEP\_05 | Verify that sleep data can be viewed in daily, weekly, and monthly summaries. | This test case verifies that users can view their sleep data aggregated in different timeframes: daily, weekly, and monthly. The test involves navigating to different timeframes within the sleep tracking section and confirming that the data displayed is accurate and appropriately summarized. | Sleep Tracking | High | User has completed multiple sleep sessions | Multiple completed sleep sessions | 1. Navigate to the sleep tracking section of the app. 2. Select the "Daily" view and verify that the sleep data is presented for the current day. 3. Select the "Weekly" view and verify that the data is summarized for the current week. 4. Select the "Monthly" view and verify that the data is summarized for the current month. | No specific postconditions | The sleep data is correctly presented in daily, weekly, and monthly summaries, providing clear and accurate overviews of the

**Data - 81**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Workout Logging

**Feature description -**

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

**Userstory title -**

As a user, I want to log my workouts so that I can track my progress and improve my performance.

**Acceptance criteria -**

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

**Generated Test Cases:**

## Fitness Tracker App - Workout Logging Test Cases

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Workout Logging**

**\*\*User Story Title:\*\* As a user, I want to log my workouts so that I can track my progress and improve my performance.**

**\*\*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-WL-01 | Verify that users can log various workout types with details | This test case checks the ability to log different workout types, including specifying the exercise type, duration, intensity, and additional notes. |  
Workout Logging | High | User is logged in to the app. | Workout type: Running, Duration: 30 minutes, Intensity: Moderate, Notes: "Ran on the treadmill." | 1. Navigate to the "Log Workout" section. <br> 2. Select "Running" as the workout type. <br> 3. Enter 30 minutes as the duration. <br> 4. Select "Moderate" as the intensity level. <br> 5. Add a note "Ran on the treadmill". <br> 6. Save the workout entry. | Workout entry is saved successfully. | Workout entry displays the correct information: "Running", 30 minutes, Moderate, "Ran on the treadmill." | Major | High | Functional Testing | Positive |

| TC-WL-02 | Verify that workout logs include duration, intensity, and notes | This test case ensures that the workout logs display the correct duration, intensity level, and any user-defined notes. | Workout Logging |  
High | A workout entry has been logged. | No test data needed | 1. Navigate to the "Workout History" section. <br> 2. View the recently logged workout. <br> 3. Verify that the workout log displays the duration, intensity level, and notes. | Workout log is displayed with the correct information. | Major | High | Functional Testing | Positive |

| TC-WL-03 | Verify that users can edit and delete workout entries | This test case checks the functionality to edit or delete an existing workout entry. | Workout Logging | Medium | A workout entry has been logged. | No

test data needed | 1. Navigate to the "Workout History" section. <br> 2. Select a workout entry to edit. <br> 3. Modify the duration, intensity, or notes. <br> 4. Save the changes. <br> 5. Verify the changes are reflected in the workout log. <br> 6. Select a different workout entry to delete. <br> 7. Confirm the deletion. | Workout entry is updated successfully. Workout entry is deleted successfully. | Major | Medium | Functional Testing | Positive |

| TC-WL-04 | Verify that data is stored and viewable for tracking progress | This test case checks the storage and retrieval of workout data for tracking progress over time. | Workout Logging | High | Multiple workout entries have been logged over a period of time. | No test data needed | 1. Navigate to the "Workout History" section. <br> 2. Verify that all previously logged workout entries are displayed chronologically. <br> 3. View the workout logs for different time periods (e.g., last week, last month). | Workout data is stored correctly and retrieved successfully. | Critical | High | Functional Testing | Positive |

| TC-WL-05 | Verify that users can filter logs by date and workout type | This test case ensures that users can filter workout logs by date and workout type for easy analysis and review. | Workout Logging | Medium | Multiple workout entries have been logged with different workout types and dates. | No test data needed | 1. Navigate to the "Workout History" section. <br> 2. Use the "Filter" option to filter workouts by a specific date range. <br> 3. Use the "Filter" option to filter workouts by a specific workout type. <br> 4. Verify that the filtered results only display relevant entries. | Filters apply correctly and display only relevant workout entries. | Major | Medium | Functional Testing | Positive |

-----

**Data - 82**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Goal Setting

**Feature description -**

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

**Userstory title -**

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

**Acceptance criteria -**

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

**Generated Test Cases:**

## Fitness Tracker App - Goal Setting Test Cases

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*User Story:\*\*** As a user, I want to set fitness goals so that I can stay motivated and track my progress.

**\*\*Acceptance Criteria:\*\***

1. Users can set goals for various fitness metrics.
2. Progress tracking is available for each goal.
3. Users receive motivational feedback and reminders.

4. Goals can be adjusted or updated as needed.
5. The app provides a summary of goal attainment.

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_GS\_01

**\*\*Test Case Title:\*\*** Verify that users can set goals for various fitness metrics.

**\*\*Test Case Description:\*\*** This test case verifies that users can set goals for different fitness metrics, including steps, calories burned, distance traveled, and weight loss.

**\*\*Test Suite:\*\*** Goal Setting

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged into the app.

\* The app has access to the user's fitness data (if applicable).

**\*\*Test Data:\*\***

\* Goal type: Steps

\* Target value: 10,000 steps per day

\* Goal type: Calories burned

\* Target value: 2,000 calories per day

\* Goal type: Distance traveled

\* Target value: 5 miles per day

\* Goal type: Weight loss

\* Target value: 1 pound per week

**\*\*Test Steps:\*\***

1. Navigate to the Goal Setting section of the app.
2. Select "Add Goal".

3. Choose a fitness metric (e.g., steps, calories burned, distance traveled, weight loss).

4. Set a target value for the chosen metric.

5. Save the goal.

**\*\*Postconditions:\*\***

\* The goal is successfully saved and displayed in the Goal list.

**\*\*Expected Result:\*\*** The app successfully saves the goal with the selected metric and target value.

**\*\*Severity:\*\*** Blocker

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_GS\_02

**\*\*Test Case Title:\*\*** Verify that progress tracking is available for each goal.

**\*\*Test Case Description:\*\*** This test case verifies that the app tracks progress towards each set goal and displays the progress visually.

**\*\*Test Suite:\*\*** Goal Setting

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User has at least one goal set.

\* The app has access to the user's fitness data (if applicable).

**\*\*Test Data:\*\***

\* Goal type: Steps

\* Target value: 10,000 steps per day

**\*\*Test Steps:\*\***

1. Navigate to the Goal Setting section of the app.
2. Select a goal to view its progress.
3. Verify that the progress bar or graph accurately reflects the user's progress towards the goal.

**\*\*Postconditions:\*\***

\* The progress display updates dynamically as the user accumulates progress.

**\*\*Expected Result:\*\*** The app displays a visual representation of the user's progress towards their goal, updated in real-time.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_GS\_03

**\*\*Test Case Title:\*\*** Verify that users receive motivational feedback and reminders.

**\*\*Test Case Description:\*\*** This test case verifies that the app provides motivational feedback and timely reminders to help users stay on track with their goals.

**\*\*Test Suite:\*\*** Goal Setting

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User has at least one goal set.

**\*\*Test Data:\*\***

\* Goal type: Steps

\* Target value: 10,000 steps per day

**\*\*Test Steps:\*\***

1. Navigate to the Goal Setting section of the app.



2. Select a goal to view its progress.

3. Observe if the app provides motivational feedback based on the user's progress (e.g., encouraging messages, progress summaries).

4. Verify that the app sends reminders for the selected goal at the appropriate time (e.g., daily, weekly).

**\*\*Postconditions:\*\***

\* The motivational feedback and reminders should be user-friendly and relevant to the goal.

**\*\*Expected Result:\*\*** The app provides motivational feedback and reminders to encourage users to stay engaged with their goals.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_GS\_04

**\*\*Test Case Title:\*\*** Verify that goals can be adjusted or updated as needed.

**\*\*Test Case Description:\*\*** This test case verifies that users can adjust or update their goals based on changes in their fitness level or goals.

**\*\*Test Suite:\*\*** Goal Setting

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User has at least one goal set.

**\*\*Test Data:\*\***

\* Goal type: Steps

\* Target value: 10,000 steps per day

**\*\*Test Steps:\*\***

1. Navigate to the Goal Setting section of the app.
2. Select a goal to modify.
3. Change the target value of the selected goal.
4. Save the updated goal.

**\*\*Postconditions:\*\***

\* The goal is successfully updated.

**\*\*Expected Result:\*\*** The app allows users to update existing goals by changing the target value or other goal parameters.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_GS\_05

**\*\*Test Case Title:\*\*** Verify that the app provides a summary of goal attainment.

**\*\*Test Case Description:\*\*** This test case verifies that the app provides a summary of the user's goal attainment, including successful completions, missed targets, and overall progress.

**\*\*Test Suite:\*\*** Goal Setting

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User has at least one goal set.

**\*\*Test Data:\*\***

\* Goal type: Steps

\* Target value: 10,000 steps per day

**\*\*Test Steps:\*\***

1. Navigate to the Goal Setting section of the app.
  2. Access the goal summary view (if available).
  3. Verify that the app displays a summary of goal attainment for each goal, including:
    - \* Number of goals successfully completed.
    - \* Number of goals missed.
    - \* Overall progress towards goals.
- \*\*Postconditions:\*\***
- \* The goal summary should be clear, concise, and easy to understand.
- \*\*Expected Result:\*\*** The app provides a summary of goal attainment for all user goals, including their progress and success rate.
- \*\*Severity:\*\*** Minor
- \*\*Priority:\*\*** Medium
- \*\*Type of Testing:\*\*** Functional Testing
- \*\*Test Case Approach:\*\*** Positive
- 

## **Data - 83**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Workout Plans

## **Feature description -**

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

## **Userstory title -**

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

## **Acceptance criteria -**

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

## **Generated Test Cases:**

## Fitness Tracker App - Workout Plans Test Cases

**\*\*Project:\*\*** Fitness Tracker App

**\*\*Feature:\*\*** Workout Plans

**\*\*User Story:\*\*** As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

**\*\*Acceptance Criteria:\*\***

1. **\*\*Users can select workout plans based on fitness levels.\*\***
2. **\*\*Plans include detailed exercise routines and schedules.\*\***
3. **\*\*Users can track adherence and progress within the plan.\*\***

- 4. \*\*The app provides notifications and reminders for workouts.\*\*
- 5. \*\*Users can adjust plans based on progress and preferences.\*\*

**\*\*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-WP-01 | Verify that users can select workout plans based on fitness levels. | This test case aims to verify that users can successfully select workout plans based on their declared fitness levels. The user will choose a fitness level (Beginner, Intermediate, Advanced) and the app should display corresponding workout plans. |  
Workout Plans | High | User is logged into the app. | User's fitness level: Beginner, Intermediate, Advanced. |  
1. Access the "Workout Plans" section. <br> 2. Select the "Fitness Level" filter. <br> 3. Choose a fitness level (e.g., Beginner). <br> 4. Observe the displayed workout plans. | Workout plans are displayed based on the chosen fitness level. | Workout plans are filtered based on fitness level. | Minor | High | Functional Testing | Positive |

| TC-WP-02 | Verify that plans include detailed exercise routines and schedules. | This test case ensures that selected workout plans include detailed exercise routines, specifying exercises, sets, reps, and rest times, as well as a clear schedule for each workout session. | Workout Plans | High | User has selected a workout plan. |  
Workout plan details: Exercise name, sets, reps, rest time, duration, frequency. | 1. Select a workout plan. <br> 2. Access the plan details. <br> 3. Verify the presence of exercise routines with specific exercises, sets, reps, rest times, and workout durations. <br> 4. Verify the existence of a clear workout schedule for each week/day. | Workout plan details are displayed correctly and comprehensively. | Workout plan details are missing or incomplete. | Major | High | Functional Testing | Positive |

| TC-WP-03 | Verify that users can track adherence and progress within the plan. | This test case verifies that users can monitor their adherence to the plan by marking completed workouts and track their progress based on completed sessions and achieved goals. | Workout Plans | High | User is following a workout plan. |

Completed workouts, progress metrics (e.g., weight loss, muscle gain), goals. | 1. Select a workout plan. <br> 2. Mark workouts as completed. <br> 3. Observe the progress tracker within the plan. <br> 4. Verify that completed workouts are reflected in the progress tracker. | Adherence and progress are accurately tracked and displayed. | Progress tracking is not accurate or missing. | Major | High | Functional Testing | Positive |

| TC-WP-04 | Verify that the app provides notifications and reminders for workouts. | This test case aims to ensure that the app sends timely notifications and reminders to users about upcoming workouts, encouraging adherence to the plan. | Workout Plans | High | User has set up notifications and reminders. | Notification settings, reminder frequency, workout schedule. | 1. Configure notification settings for workouts. <br> 2. Schedule a workout within the plan. <br> 3. Observe the notification delivery at the scheduled time. | Notifications and reminders are delivered as scheduled. | Notifications and reminders are not delivered or delivered at incorrect times. | Major | High | Non-Functional Testing | Positive |

| TC-WP-05 | Verify that users can adjust plans based on progress and preferences. | This test case validates that users can modify the plan, including adjusting exercises, adjusting workout frequencies, and adding/removing workouts based on their progress or changing preferences. | Workout Plans | Medium | User is following a workout plan. | Modified plan details: Exercise changes, frequency adjustments, added/removed workouts. | 1. Access the selected workout plan. <br> 2. Select the "Adjust Plan" option. <br> 3. Modify exercises, frequency, or add/remove workouts. <br> 4. Save the changes. | The plan is adjusted based on user input. | Plan adjustments are not saved or not reflected in the plan. | Minor | Medium | Functional Testing | 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:**

## Test Cases for Fitness Tracker App - Community Support Feature

**Project:** Fitness Tracker App

**Feature:** Community Support

**User Story:** As a user, I want to connect with a community so that I can share my progress and stay motivated.

**Acceptance Criteria:**

1. Users can join and participate in community groups.
2. Users can share progress updates and achievements.
3. The app facilitates participation in community challenges.
4. Users can offer and receive support and encouragement.
5. Community activity and engagement are tracked.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-CS-01

**\*\*Test Case Title:\*\*** Verify that users can join and participate in community groups.

**\*\*Test Case Description:\*\*** This test case verifies that a user can successfully join an existing community group and view group content.

**\*\*Test Suite:\*\*** Community Support

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User account is created and logged in.

\* Community groups are available within the app.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Community" section of the app.
2. Select a community group from the list.
3. Click on the "Join Group" button.
4. Verify that the user is added to the group and can access group content, including posts, comments, and member profiles.

**\*\*Postconditions:\*\***

\* User remains logged in.



\* The user is a member of the selected community group.

**\*\*Expected Result:\*\*** The user successfully joins the selected community group and can access its content.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CS-02

**\*\*Test Case Title:\*\*** Verify that users can share progress updates and achievements.

**\*\*Test Case Description:\*\*** This test case verifies that a user can successfully share their progress updates and achievements within a community group.

**\*\*Test Suite:\*\*** Community Support

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User account is created and logged in.

\* User is a member of at least one community group.

\* User has recent progress or achievement data within the app (e.g., completed workouts, achieved goals).

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to a community group the user is a member of.
2. Click on the "Share Progress" button.
3. Select a recent progress update or achievement from the available options.
4. Add a brief message or comment if desired.
5. Click on the "Share" button.
6. Verify that the update or achievement is visible on the group feed.

**\*\*Postconditions:\*\***

- \* User remains logged in.
- \* The shared progress update or achievement is visible in the group feed.

**\*\*Expected Result:\*\*** The user successfully shares a progress update or achievement within the group.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CS-03

**\*\*Test Case Title:\*\*** Verify that the app facilitates participation in community challenges.

**\*\*Test Case Description:\*\*** This test case verifies that the app enables users to join, track progress, and compete in community challenges.

**\*\*Test Suite:\*\*** Community Support

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* User account is created and logged in.
- \* User is a member of at least one community group.
- \* Active community challenges are available within the app.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to a community group the user is a member of.
2. Locate and select an active community challenge.
3. Click on the "Join Challenge" button.
4. Track the challenge progress within the app, such as steps taken, calories burned, or time spent exercising.

5. Verify that the challenge progress is displayed correctly and that the user can view their ranking amongst other participants.

**\*\*Postconditions:\*\***

\* User remains logged in.

\* Challenge progress is tracked and displayed correctly.

**\*\*Expected Result:\*\*** The user can successfully join, track progress, and view their ranking in a community challenge.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CS-04

**\*\*Test Case Title:\*\*** Verify that users can offer and receive support and encouragement.

**\*\*Test Case Description:\*\*** This test case verifies that users can provide and receive encouragement through comments and reactions on progress updates and achievements shared within a community group.

**\*\*Test Suite:\*\*** Community Support

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User account is created and logged in.

\* User is a member of at least one community group.

\* Other users have shared progress updates or achievements within the group.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to a community group the user is a member of.

2. View a progress update or achievement shared by another user.
3. Leave a comment offering encouragement or support.
4. Use the available reaction options (e.g., likes, thumbs up) to express appreciation.
5. Verify that the comment and reactions are displayed on the update or achievement post.
6. Repeat steps 2-5 to receive encouragement and support from other users on the user's own progress updates or achievements.

**\*\*Postconditions:\*\***

- \* User remains logged in.
- \* Comments and reactions are visible on the shared updates or achievements.

**\*\*Expected Result:\*\*** The user can successfully provide and receive encouragement through comments and reactions within the community group.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-CS-05

**\*\*Test Case Title:\*\*** Verify that community activity and engagement are tracked.

**\*\*Test Case Description:\*\*** This test case verifies that the app tracks community activity, such as the number of posts, comments, reactions, and active users within a group.

**\*\*Test Suite:\*\*** Community Support

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- \* User account is created and logged in.
- \* User is a member of at least one community group.
- \* Other users have been active in the community group, sharing updates, commenting, and reacting.

**\*\*Test Data:\*\***

\* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to a community group the user is a member of.
2. Locate the "Activity" or "Engagement" section of the group.
3. Verify that the app displays metrics such as:
  - \* Total number of posts
  - \* Total number of comments
  - \* Total number of reactions
  - \* Number of active users
4. Observe how the metrics change as users interact with the group (e.g., share new updates, comment, react).

**\*\*Postconditions:\*\***

- \* User remains logged in.
- \* Community activity metrics are updated and displayed correctly.

**\*\*Expected Result:\*\*** The app accurately tracks community activity and engagement metrics within the group, providing insight into user behavior and interaction.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 85**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Integration with Wearable Devices

**Feature description -**

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

**Userstory title -**

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

**Acceptance criteria -**

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

**Generated Test Cases:**

## Test Cases for Fitness Tracker App - Wearable Integration

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Integration with Wearable Devices**

**\*\*User Story Title:\*\* As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.**

**\*\*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_WI_01	Verify that the app supports integration with multiple wearable devices	This test case verifies the app's compatibility with various popular wearable devices. It ensures that the user can successfully connect their chosen wearable device to the app.	Wearable Integration	High	User has a compatible wearable device	Wearable device name, model number, and type	1. Open the app and navigate to the settings section. 2. Select the "Wearable Devices" option and choose the appropriate device type. 3. Follow the instructions provided to pair the wearable device with the app. 4. Verify that the device is successfully connected and listed within the app's settings.	The app successfully identifies and connects to the selected wearable device.	Major	High	Integration Testing	Positive
----------	---	---	----------------------	------	---------------------------------------	--	--	---	-------	------	---------------------	----------

TC_WI_02	Verify that users can sync activity data from their wearables	This test case ensures that activity data from the wearable device is transferred and displayed in the app after successful connection.	Wearable Integration	High	User has a compatible wearable device with recorded activity data	Step count, distance travelled, calories burned, heart rate data (if available)	1. Initiate a workout session using the wearable device. 2. Sync the wearable device with the app after the workout. 3. Verify that the activity data from the wearable is reflected in the app's activity log.	Activity data from the wearable is successfully synced and displayed in the app.	Major	High	Functional Testing	Positive
----------	---	---	----------------------	------	---	---	---	--	-------	------	--------------------	----------

<p>  TC_WI_03  </p>	<p>Verify that the app provides real-time data updates from wearables   This test case validates whether the app receives real-time updates of data from the wearable device during an active session.  </p>	<p>Wearable Integration   High   User has a compatible wearable device with live data streaming capabilities  </p>
<p>Live heart rate data, steps taken during a workout  </p>	<p>1. Start an exercise session using the wearable device.   2. Open the app and navigate to the live data view.  </p>	<p>3. Verify that the app displays live updates of relevant metrics (e.g., heart rate, step count) from the wearable device in real-time.  </p>
<p>of data from the wearable device during an active session.  </p>	<p>Major   High   Functional Testing   Positive  </p>	

TC_WI_04	Verify that wearable data includes steps, heart rate, and calories burned	This test case checks if the app successfully pulls the expected data points from the wearable device, including step count, heart rate, and calories burned.
----------	---	---

and calories burned. | Wearable Integration | Medium | User has a compatible wearable device with recorded activity data | Step count, distance travelled, calories burned, heart rate data (if available) | 1. Start a workout session using the wearable device. | 2. Sync the wearable device with the app after the workout. | 3. Verify that the app successfully displays step count, heart rate, and calories burned data collected by the wearable device. | The app displays step count, heart rate, and calories burned data collected by the wearable device. | Minor | Medium | Functional Testing | Positive |

| TC\_WI\_05 | Verify that users can troubleshoot connectivity issues with wearables | This test case checks the app's ability to assist users in resolving connectivity problems with their wearable device. | Wearable Integration | Low | User has a compatible wearable device with known connectivity issues | No test data needed | 1. Deliberately disconnect the wearable device from the app. | 2. Attempt to reconnect the wearable device to the app. | 3. Verify that the app provides clear instructions or troubleshooting steps to address the connectivity issues. | The app provides clear and helpful instructions to troubleshoot connectivity issues with the wearable device. | Minor | Low | Usability Testing | Negative |

-----

**Data - 86**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Water Intake Tracking

**Feature description -**



Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

**Userstory title -**

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

**Acceptance criteria -**

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

**Generated Test Cases:**

## Fitness Tracker App - Water Intake Tracking Test Cases

**Project Name:** Fitness Tracker App

**Feature Name:** Water Intake Tracking

**User Story:** As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

**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-001	Verify that users can log their daily water intake	This test case verifies the functionality of logging water intake. The user will log a specific amount of water consumed at a specific time.	Water Intake Tracking	High	User is logged in to the app	Water intake value (e.g., 250ml) and time (e.g., 10:00 AM)	1. Navigate to the Water Intake section. 2. Enter the amount of water consumed (e.g., 250ml) in the input field.						

3. Select the time of consumption (e.g., 10:00 AM). 4. Tap the "Log" or "Add" button. | Water intake is logged in the app, and the total water intake for the day is updated. | Water intake is successfully logged with the correct amount and time. | Minor | High | Functional Testing | Positive |

| TC-002 | Verify that the app calculates total water consumption for the day | This test case verifies that the app accurately calculates the total water consumed throughout the day by adding up individual entries. | Water Intake Tracking | High | User has logged multiple water intake entries | Multiple water intake values and timestamps (e.g., 250ml at 10:00 AM, 300ml at 12:00 PM, 200ml at 2:00 PM) | 1. Log multiple water intake entries throughout the day. 2. Navigate to the Water Intake section. 3. Observe the total water consumed for the day. | The total water consumption for the day is displayed correctly based on the sum of all logged entries. | Major | High | Functional Testing | Positive |

| TC-003 | Verify that users can set daily hydration goals | This test case checks if users can define a specific target water intake amount for the day. | Water Intake Tracking | High | User is logged in to the app | Hydration goal value (e.g., 2 Liters) | 1. Navigate to the Water Intake section. 2. Tap on the "Set Goal" option. 3. Enter the desired daily hydration goal (e.g., 2 Liters) in the input field. 4. Tap on "Save" or "Confirm." | The hydration goal is saved and displayed prominently in the Water Intake section. | The hydration goal is saved correctly and can be viewed by the user. | Minor | High | Functional Testing | Positive |

| TC-004 | Verify that reminders are provided to encourage regular water intake | This test case checks if the app sends timely reminders to the user to drink water based on the set goal and their current water intake progress. | Water Intake Tracking | High | User has set a hydration goal and enabled reminders | Hydration goal value (e.g., 2 Liters), reminder frequency (e.g., every 2 hours) | 1. Set a hydration goal (e.g., 2 Liters). 2. Enable reminder notifications within the app settings. 3. Observe the reminder notification at the predefined intervals. | The user receives timely reminders to drink water based on the set frequency and their current progress towards their hydration goal. | Major | High | Functional Testing | Positive |

| TC-005 | Verify that users can view hydration progress and trends | This test case verifies the ability to view their current progress towards the set hydration goal and observe past water intake patterns. | Water Intake Tracking | High | User has logged water intake data for multiple days | No test data needed | 1. Navigate to the Water Intake section. 2. Observe the progress bar indicating the current water intake against the set goal. 3.

Switch to the "Trends" view (if available). | The app displays a clear visualization of the user's daily water intake progress against the set goal. Trends view provides a graphical representation of water intake patterns over a specified period (e.g., weekly, monthly). | Minor | High | Functional Testing | Positive |

---

**Data - 87**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Nutrition Tracking

**Feature description -**

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

**Userstory title -**

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

**Acceptance criteria -**

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

**Generated Test Cases:**

## ## Test Cases for Fitness Tracker App - Nutrition Tracking

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Nutrition Tracking**

**\*\*User Story Title:\*\* As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.**

**\*\*Acceptance Criteria:\*\***

1. Users can log their daily meals and snacks.
2. The app calculates macronutrient intake (carbs, fats, proteins).
3. Users can set nutritional goals and track progress.
4. The app provides nutritional information and suggestions.
5. Users can view their nutritional history and trends.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\* TC-NUTR-01**

**\*\*Test Case Title:\*\* Verify that users can log their daily meals and snacks.**

**\*\*Test Case Description:\*\* This test case verifies the functionality of logging meals and snacks within the app. It checks if the user can input different food items, specify quantities, and save the entries.**

**\*\*Test Suite:\*\* Nutrition Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- \* User account is created and logged in.
- \* The app has access to a food database or allows manual entry.

**\*\*Test Data:\*\***

- \* Sample meal: Breakfast (1 cup oatmeal, 1/2 cup berries)

\* Sample snack: Afternoon snack (1 apple, 10 almonds)

**\*\*Test Steps:\*\***

1. Open the "Nutrition" or "Food Diary" section of the app.
2. Select "Log Meal" or "Add Food".
3. Search for or manually enter the breakfast meal items (oatmeal, berries) and specify their quantities.
4. Save the meal entry.
5. Repeat steps 2-4 for the afternoon snack (apple, almonds).

**\*\*Postconditions:\*\***

\* The meal and snack entries should be saved in the user's food diary.

**\*\*Expected Result:\*\*** The app successfully saves the entered meals and snacks with their respective food items, quantities, and times.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-NUTR-02

**\*\*Test Case Title:\*\*** Verify that the app calculates macronutrient intake (carbs, fats, proteins).

**\*\*Test Case Description:\*\*** This test case checks if the app correctly calculates the total macronutrients consumed based on the logged meals and snacks.

**\*\*Test Suite:\*\*** Nutrition Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User account is created and logged in.

\* Meals and snacks are logged as per TC-NUTR-01.

**\*\*Test Data:\*\***

\* Macronutrient values for oatmeal, berries, apple, and almonds.

**\*\*Test Steps:\*\***

1. Access the "Nutrition Summary" or "Macros" section of the app.
2. Verify that the app displays the total macronutrient intake (carbs, fats, proteins) for the day.
3. Compare the displayed macronutrient values with the calculated values based on the food database or manual input.

**\*\*Postconditions:\*\***

\* None.

**\*\*Expected Result:\*\*** The app accurately calculates and displays the total macronutrient intake for the day, matching the calculated values from the food database or manual input.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-NUTR-03

**\*\*Test Case Title:\*\*** Verify that users can set nutritional goals and track progress.

**\*\*Test Case Description:\*\*** This test case checks if the user can define their desired daily macronutrient targets and track their progress towards achieving these goals.

**\*\*Test Suite:\*\*** Nutrition Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User account is created and logged in.

**\*\*Test Data:\*\***

\* Sample nutritional goals: Carbs - 150g, Fats - 60g, Proteins - 100g

**\*\*Test Steps:\*\***

1. Navigate to the "Nutrition Goals" or "Target Macros" section of the app.
2. Enter the desired daily targets for carbs, fats, and proteins.

3. Save the nutritional goals.
4. Access the "Progress" or "Nutrition Tracking" section.
5. Verify that the app displays the progress bar for each macronutrient, indicating the user's intake vs. their set goals.

**\*\*Postconditions:\*\***

\* None.

**\*\*Expected Result:\*\*** The app allows users to set nutritional goals and provides a visual representation of their progress towards achieving those goals.

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-NUTR-04

**\*\*Test Case Title:\*\*** Verify that the app provides nutritional information and suggestions.

**\*\*Test Case Description:\*\*** This test case checks if the app provides insights into the user's nutritional intake and offers personalized recommendations based on their dietary habits.

**\*\*Test Suite:\*\*** Nutrition Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User account is created and logged in.

\* Meals and snacks are logged.

\* Nutritional goals are set.

**\*\*Test Data:\*\***

\* None.

**\*\*Test Steps:\*\***

1. Access the "Insights" or "Nutrition Report" section of the app.

2. Verify that the app displays information about the user's macronutrient intake, including graphs or charts.
3. Check for personalized recommendations based on the user's nutritional intake, goals, and dietary habits.
4. Observe if the app suggests food choices, meal planning tips, or strategies for achieving better nutritional balance.

**\*\*Postconditions:\*\***

\* None.

**\*\*Expected Result:\*\*** The app provides informative insights into the user's nutritional intake, along with personalized recommendations and suggestions to improve their diet.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-NUTR-05

**\*\*Test Case Title:\*\*** Verify that users can view their nutritional history and trends.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of viewing historical nutritional data and analyzing trends over time.

**\*\*Test Suite:\*\*** Nutrition Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User account is created and logged in.

\* Meals and snacks are logged over a period of time (e.g., a week).

**\*\*Test Data:\*\***

\* None.

**\*\*Test Steps:\*\***

1. Open the "Nutrition History" or "Food Diary Trends" section of the app.
2. Select a specific timeframe (e.g., last week, last month) to view historical nutritional data.



3. Verify that the app displays a timeline or graphical representation of the user's macronutrient intake over the selected timeframe.
4. Check if the app identifies any trends or patterns in the user's dietary habits, such as fluctuations in macronutrient intake or recurring food choices.

**\*\*Postconditions:\*\***

\* None.

**\*\*Expected Result:\*\*** The app allows users to view their nutritional history and analyze trends over time, providing insights into their dietary patterns and potential areas for improvement.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

These test cases are designed to ensure the functionality and usability of the nutrition tracking feature within the fitness tracker app. They cover all the specified acceptance criteria and provide a comprehensive assessment of the feature's quality.

-----

**Data - 88**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

## **Feature name -**

Progress Reports

## **Feature description -**

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

## **Userstory title -**

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

## **Acceptance criteria -**

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

## **Generated Test Cases:**

## Test Cases for Fitness Tracker App - Progress Reports Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Progress Reports**

**\*\*User Story Title:\*\* As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.**

**\*\*Acceptance Criteria:\*\***

1. Users receive weekly and monthly progress reports.
2. Reports include activity levels, workout performance, and goal attainment.
3. Users can view detailed breakdowns of achievements and trends.

4. The app provides recommendations based on report findings.

5. Users can customize the frequency and content of reports.

**\*\*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_PR_01	Verify that users receive weekly progress reports	This test case verifies that users receive weekly progress reports as scheduled. The user will be logged into the app and the system time will be advanced to the next scheduled report delivery.	Progress Reports	High	User account created and logged in.	No test data needed.	1. Advance system time to the next scheduled weekly report delivery. 2. Check the app notification center for a new progress report. 3. Verify that the notification indicates a weekly progress report is available.	Notification should be present and indicate a weekly progress report is available.	Major	High	Functional Testing	Positive
----------	---	---	------------------	------	-------------------------------------	----------------------	---	--	-------	------	--------------------	----------

TC_PR_02	Verify that users receive monthly progress reports	This test case verifies that users receive monthly progress reports as scheduled. The user will be logged into the app and the system time will be advanced to the next scheduled report delivery.	Progress Reports	High	User account created and logged in.	No test data needed.	1. Advance system time to the next scheduled monthly report delivery. 2. Check the app notification center for a new progress report. 3. Verify that the notification indicates a monthly progress report is available.	Notification should be present and indicate a monthly progress report is available.	Major	High	Functional Testing	Positive
----------	--	--	------------------	------	-------------------------------------	----------------------	---	---	-------	------	--------------------	----------

TC_PR_03	Verify that progress reports include activity levels, workout performance, and goal attainment	This test case verifies that progress reports contain information about user activity levels, workout performance, and goal attainment. A user's activity levels, workouts, and goals will be simulated and the progress report will be checked for corresponding data.	Progress Reports	High	User account created and logged in, user activity and goal data is populated.	User activity and goal data (steps taken, calories burned,
----------	--	---	------------------	------	---	--

workouts completed, goal progress, etc.). | 1. Access the progress report. | 2. Verify that the report displays the user's activity levels (e.g., steps taken, calories burned, distance traveled). | 3. Verify that the report displays the user's workout performance (e.g., workout durations, types, intensity). | 4. Verify that the report displays the user's goal progress (e.g., progress towards weight loss, steps goal, etc.). | All data should be present and accurately reflect the simulated user data. | Major | High | Functional Testing | Positive |

| TC\_PR\_04 | Verify that users can view detailed breakdowns of achievements and trends | This test case verifies that users can access detailed breakdowns of their achievements and trends within the progress report.

| Progress Reports | High | User account created and logged in, user activity and goal data is populated. | No test data needed. | 1. Access the progress report. | 2. Navigate through the report to locate the section displaying achievements and trends. | 3. Verify that the section provides detailed information on user achievements, such as streaks, personal bests, and progress towards goals. | 4. Verify that the section provides visual representations of trends, such as line graphs or charts, depicting user progress over time. | Detailed breakdowns of achievements and trends should be present and accessible within the report. | Major | High | Functional Testing | Positive |

| TC\_PR\_05 | Verify that users can customize the frequency and content of reports | This test case verifies that users can adjust the frequency and content of progress reports according to their preferences. | Progress Reports | Medium | User account created and logged in. | No test data needed. | 1. Navigate to the settings section of the app. | 2. Locate the Progress Reports settings. | 3. Verify that options exist to modify the report frequency (e.g., weekly, bi-weekly, monthly, custom). | 4. Verify that options exist to customize the report content (e.g., include/exclude specific metrics, focus on specific goals). | 5. Make changes to the frequency and content settings. | 6. Verify that the app successfully saves the changes. | Report frequency and content should be customizable and changes should be reflected in subsequent reports. | Minor | Medium | Functional Testing | Positive |

**\*\*Notes:\*\***

\* These test cases cover the core functionality of the progress reports feature.

- \* It is essential to add further test cases for edge cases, negative scenarios, and non-functional aspects of the feature (e.g., performance, usability, security).
  - \* This template can be adapted and expanded to suit the specific needs of your project.
- 

**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:**

## Fitness Tracker App - In-App Challenges Test Cases

\*\*Project Context:\*\* Fitness Tracker App

\*\*Feature Name:\*\* In-App Challenges

\*\*User Story 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:\*\*

- 1. Users can join various in-app challenges.
- 2. Challenges have clear objectives and rules.
- 3. Users can track their progress within the challenge.
- 4. Rewards are provided for achieving challenge milestones.
- 5. Users can compete with friends and view leaderboards.

\*\*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_01	Verify that users can join various in-app challenges	This test case verifies that users can successfully join different types of challenges available in the app, including those based on different activities, durations, and difficulty levels.	In-App Challenges	High	User account created and logged in	"Step Challenge", "Calorie Burn Challenge", "Distance Challenge"	1. Navigate to the "Challenges" section of the app. 2. Select a challenge from the list of available challenges. 3. Verify that the user is successfully enrolled in the challenge.	User is successfully enrolled in the chosen challenge.	Major	High	Functional Testing	Positive	

| TC\_02 | Verify that challenges have clear objectives and rules | This test case checks whether each challenge has clearly defined objectives, rules, and criteria for success. | In-App Challenges | High | User is enrolled in a challenge | "Step Challenge" (Objective: Take 10,000 steps daily for 7 days. Rule: Use the app's step tracker to record steps.) | 1. Navigate to the "Challenge Details" page for a selected challenge. | 2. Verify that the challenge objective, rules, and duration are clearly displayed. | 3. Verify that all necessary information for participation is provided. | Challenge details, including objective, rules, and duration, are clearly visible and understandable. | Major | High | Functional Testing | Positive |

| TC\_03 | Verify that users can track their progress within a challenge | This test case checks if the app allows users to monitor their progress in real-time and see their current position compared to the challenge objective. | In-App Challenges | High | User is enrolled in a challenge | "Step Challenge" | 1. Navigate to the "Challenge Progress" section for a selected challenge. | 2. Verify that the progress bar is accurately reflecting the user's progress towards the challenge goal. | 3. Verify that the user's current progress is displayed in a clear and understandable manner. | The user can track their progress towards the challenge goal in real-time, with a visually clear display of their progress. | Major | High | Functional Testing | Positive |

| TC\_04 | Verify that rewards are provided for achieving challenge milestones | This test case checks if users receive rewards, such as badges or points, upon reaching specified milestones within a challenge. | In-App Challenges | High | User is enrolled in a challenge | "Step Challenge" (Milestone: Complete 5 consecutive days of the challenge) | 1. Achieve a challenge milestone (e.g., completing 5 consecutive days). | 2. Verify that the user receives a corresponding reward (e.g., a badge). | 3. Verify that the reward is displayed in the user's profile or within the challenge section. | The user receives a reward for achieving a milestone within the challenge. | Major | High | Functional Testing | Positive |

| TC\_05 | Verify that users can compete with friends and view leaderboards | This test case checks if the app allows users to compete with friends, view a leaderboard, and see their ranking compared to other participants. | In-App Challenges | High | User is enrolled in a challenge | "Step Challenge" | 1. Navigate to the "Leaderboard" section of the challenge. | 2. Verify that the leaderboard displays all participants in the challenge, including the user's friend. | 3. Verify that the user can see their ranking and the ranking of their friend. | The user can compete with friends and view their ranking on the challenge leaderboard. | Major | High

**Data - 90**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Customizable Dashboards

**Feature description -**

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

**Userstory title -**

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

**Acceptance criteria -**

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

**Generated Test Cases:**



## ## Test Cases for Customizable Dashboards Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Customizable Dashboards**

**\*\*User Story 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:\*\***

1. Users can select which metrics to display on their dashboard.
2. The app allows customization of dashboard layout and theme.
3. Users can save and update their dashboard settings.
4. Dashboard displays are dynamic and update in real-time.
5. Users can reset the dashboard to default settings.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* TC\_Dashboard\_01**

**\*\*Test Case Title:\*\* Verify that users can select which metrics to display on their dashboard.**

**\*\*Test Case Description:\*\* This test case verifies that users can choose which fitness metrics (e.g., steps, calories burned, heart rate, distance) they want to display on their dashboard.**

**\*\*Test Suite:\*\* Dashboard Functionality**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged into the app.
- The dashboard is accessible.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the dashboard.
2. Locate the "Customize Dashboard" or similar option.
3. Select or deselect various fitness metrics from the available options.
4. Observe the changes on the dashboard in real-time.

**\*\*Postconditions:\*\***

- The selected metrics are displayed on the dashboard.
- Deselected metrics are removed from the dashboard.

**\*\*Expected Result:\*\*** The dashboard dynamically updates based on the selected metrics.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_Dashboard\_02

**\*\*Test Case Title:\*\*** Verify that the app allows customization of dashboard layout and theme.

**\*\*Test Case Description:\*\*** This test case checks if users can modify the arrangement of elements (e.g., widgets, sections) and choose different color schemes or themes for their dashboard.

**\*\*Test Suite:\*\*** Dashboard Customization

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged into the app.
- The dashboard is accessible.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the dashboard.
2. Locate the "Layout" or "Theme" customization options.
3. Experiment with different layout options (e.g., grid, list, compact).
4. Select various color themes from the available options.
5. Observe the changes in dashboard appearance.

**\*\*Postconditions:\*\***

- The dashboard layout and theme are changed as per user selections.

**\*\*Expected Result:\*\*** The dashboard's appearance reflects the selected layout and theme.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_Dashboard\_03

**\*\*Test Case Title:\*\*** Verify that users can save and update their dashboard settings.

**\*\*Test Case Description:\*\*** This test case ensures that users can store their customized dashboard settings and modify them later.

**\*\*Test Suite:\*\*** Dashboard Settings

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged into the app.
- The dashboard is accessible.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the dashboard.

2. Customize the dashboard (select metrics, change layout, choose theme).

3. Save the changes to the dashboard settings.

4. Exit the app and re-login.

5. Access the dashboard again.

6. Verify that the saved settings are applied to the dashboard.

7. Modify the dashboard settings again.

8. Save the updated settings.

9. Observe the changes reflected on the dashboard.

**\*\*Postconditions:\*\***

- The dashboard settings are saved successfully.

- Updated settings override the previously saved settings.

**\*\*Expected Result:\*\*** The dashboard displays the last saved settings upon access.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_Dashboard\_04

**\*\*Test Case Title:\*\*** Verify that dashboard displays are dynamic and update in real-time.

**\*\*Test Case Description:\*\*** This test case checks if the dashboard reflects changes to fitness data (e.g., steps, calories burned) in real-time without requiring manual refresh.

**\*\*Test Suite:\*\*** Dashboard Real-Time Updates

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged into the app.

- The dashboard is accessible.

- Fitness tracking data is being collected (e.g., using wearable device or manual input).

**\*\*Test Data:\*\***

- User's fitness data (steps, calories burned, heart rate).

**\*\*Test Steps:\*\***

1. Navigate to the dashboard.
2. Select a specific metric (e.g., steps) to display.
3. Initiate physical activity to generate new fitness data.
4. Observe the dashboard in real-time for changes in the selected metric.

**\*\*Postconditions:\*\***

- The dashboard automatically updates with the latest fitness data.

**\*\*Expected Result:\*\*** The displayed metric value reflects the real-time changes in fitness data.

**\*\*Severity:\*\*** Major

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Integration Testing, Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_Dashboard\_05

**\*\*Test Case Title:\*\*** Verify that users can reset the dashboard to default settings.

**\*\*Test Case Description:\*\*** This test case confirms that users can revert their customized dashboard to its initial default state, removing any personal settings applied.

**\*\*Test Suite:\*\*** Dashboard Reset

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged into the app.
- The dashboard is accessible.
- The dashboard is customized with different settings.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the dashboard.
2. Locate the "Reset Dashboard" or similar option.
3. Confirm the action of resetting the dashboard to default settings.
4. Observe the dashboard after the reset.

**\*\*Postconditions:\*\***

- The dashboard reverts to its default layout, theme, and metrics.
- Any user-defined settings are removed.

**\*\*Expected Result:\*\*** The dashboard displays its original default state.

**\*\*Severity:\*\*** Minor

**\*\*Priority:\*\*** Medium

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Negative (resets to default state)

-----

**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**

**\*\*Project Context:\*\***

**\* \*\*Project Name:\*\* Fitness Tracker App**

**\* \*\*Feature Name:\*\* Social Sharing**

**\* \*\*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.**

**\*\*Acceptance Criteria:\*\***

1. Users can share workout summaries and achievements.
2. The app supports sharing on multiple social media platforms.
3. Users can customize the content and format of shared posts.
4. Sharing features are integrated seamlessly with social media.

5. Users can track the engagement of their shared posts.

**\*\*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-SS-01 | Verify that users can share workout summaries | This test case verifies that users can share summaries of completed workouts on social media platforms. | Social Sharing | High | User is logged in and has completed a workout. | Sample workout data (duration, distance, calories burned, etc.) | 1. Navigate to the workout summary page. 2. Select the "Share" option. 3. Choose a social media platform to share on. 4. Review the pre-populated share text and adjust it if needed. 5. Click "Share". | The workout summary is shared on the selected social media platform. | The user receives confirmation that the share was successful. | Minor | High | Functional Testing | Positive |

| TC-SS-02 | Verify that the app supports sharing on multiple social media platforms | This test case verifies that the app supports sharing workout summaries and achievements on different social media platforms. | Social Sharing | High | User is logged in and has completed a workout. | Sample workout data, user accounts on different platforms (e.g., Facebook, Twitter, Instagram). | 1. Navigate to the workout summary page. 2. Select the "Share" option. 3. Verify that the app offers various social media platforms to choose from (e.g., Facebook, Twitter, Instagram). 4. Select a different platform for each test iteration. 5. Click "Share". | The workout summary is shared successfully on each selected platform. | Minor | High | Functional Testing | Positive |

| TC-SS-03 | Verify that users can customize the content and format of shared posts | This test case verifies that users can customize the text, images, and hashtags used when sharing workout summaries and achievements. | Social Sharing | Medium | User is logged in and has completed a workout. | Sample workout data, pre-populated share text, custom images. | 1. Navigate to the workout summary page. 2. Select the "Share" option. 3. Edit the pre-populated share text to add custom details and hashtags. 4. Add an image from



the gallery to personalize the post. 5. Click "Share". | The workout summary is shared on the selected platform with the customized content and image. | Minor | Medium | Functional Testing | Positive |

| TC-SS-04 | Verify seamless integration with social media platforms | This test case verifies that the app seamlessly integrates with social media platforms to share workout data. | Social Sharing | High | User is logged in and has completed a workout, social media accounts are linked. | Sample workout data. | 1. Navigate to the workout summary page. 2. Select the "Share" option. 3. Choose a social media platform to share on. 4. Click "Share". 5. Verify that the user is automatically logged into the social media platform. | The sharing process is smooth and doesn't require additional login steps. | Minor | High | Integration Testing | Positive |

| TC-SS-05 | Verify that users can track the engagement of their shared posts | This test case verifies that users can view the likes, comments, and shares received on their shared workout summaries and achievements. | Social Sharing | Medium | User is logged in and has shared a workout summary. | Sample workout data, shared post with engagement metrics. | 1. Navigate to the "Social" section of the app. 2. View the shared workout summary. 3. Verify that the app displays the number of likes, comments, and shares for the post. | The engagement metrics for the shared post are displayed accurately. | Minor | Medium | Functional Testing | 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:

## ## Test Cases for Fitness Tracker App - Rest and Recovery Tracking

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Rest and Recovery Tracking**

**User Story Title:** As a user, I want to track my rest and recovery so that I can optimize my fitness performance and prevent overtraining.

### \*\*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-01\*\*** | Verify that users can log rest days | This test case verifies that users can manually log rest days in the app. | Rest and Recovery Tracking | High | - User is logged into the app. | No test data needed. | 1. Navigate to the Rest and Recovery section of the app. 2. Select "Log Rest Day." 3. Choose the date of the rest day. 4. Confirm the log entry. | - The rest day is logged in the app's calendar. | The rest day is successfully logged, and the app's calendar displays the logged rest day. | Minor | Medium | Functional Testing | Positive

**\*\*TC-02\*\*** | Verify that the app tracks resting heart rate | This test case verifies that the app can track and display resting heart rate data. | Rest and Recovery Tracking | High | - User is logged into the app. - User has a connected wearable device or has manually inputted resting heart rate data. | Resting heart rate data from a wearable device or manually entered data. | 1. Access the Rest and Recovery section of the app. 2. Navigate to the "Resting Heart Rate" section. 3. Verify that the app displays the latest resting heart rate data. | - The resting heart rate data is displayed in the app. | The app displays the correct resting heart rate data, sourced from the connected wearable device or manual input. | Major | High | Functional Testing | Positive

**\*\*TC-03\*\*** | Verify that users receive recovery recommendations based on their activities | This test case verifies that the app provides personalized recovery recommendations based on user's activity data. | Rest and Recovery Tracking | High | - User is logged into the app. - User has logged recent activity data (exercise, sleep, etc.). | Recent activity data, including exercise duration, intensity, sleep duration and quality. | 1. Access the Rest and Recovery section of the app. 2. View the "Recovery Recommendations" section. 3. Verify that the app provides personalized recommendations based on the logged activity data, such as recommended rest time, recovery activities, or hydration advice. | - The app displays personalized recovery recommendations based on the user's recent activity data. | Minor | Medium | Functional Testing | Positive

**\*\*TC-04\*\*** | Verify that the app provides alerts for insufficient recovery time | This test case verifies that the app alerts the user if their recovery time is insufficient based on their activity data. | Rest and Recovery Tracking | High | - User is logged into the app. - User has logged recent activity data (exercise, sleep, etc.) and has not logged sufficient recovery time based on the app's calculations. | Recent activity data (exercise, sleep, etc.) that indicates insufficient recovery time. | 1. Access the Rest and Recovery section of the app. 2. View the "Recovery Time" section. 3. Verify that the app displays a notification or alert indicating insufficient

recovery time. | - The app displays an alert or notification indicating insufficient recovery time. | Major | High |  
Functional Testing | Negative

**\*\*TC-05\*\*** | Verify that users can view trends and history of their recovery data | This test case verifies that users can view historical data and trends of their recovery metrics, such as resting heart rate and sleep quality. |  
Rest and Recovery Tracking | Medium | - User is logged into the app. - User has logged historical recovery data (resting heart rate, sleep quality, etc.). | Historical recovery data, including resting heart rate and sleep quality data over a period of time. | 1. Access the Rest and Recovery section of the app. 2. Navigate to the "Recovery History" section. 3. Verify that the app displays historical data and trends for various recovery metrics (e.g., resting heart rate, sleep quality) over time. 4. Check if the app allows users to filter or sort the data by date range or metric. | - The app displays historical recovery data and trends in a visually appealing and easily understandable format. | Minor | Medium | Functional Testing | Positive

**\*\*Note:\*\*** This is a sample set of test cases. You may need to create additional test cases based on the specific features and functionalities of the Fitness Tracker App. It is crucial to cover all edge cases and scenarios to ensure the app's quality and functionality.

-----

**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:

#### ## Test Cases for Fitness Tracker App - Audio Coaching Feature

**\*\*Project Context:\*\*** Fitness Tracker App

**\*\*User Story:\*\*** As a user, I want to access audio coaching during workouts so that I can stay motivated and receive real-time guidance.

**\*\*Acceptance Criteria:\*\***

1. Users can access a variety of audio-guided workouts.
2. Audio coaching includes motivational messages and coaching cues.
3. Users can choose coaching based on workout type and intensity.
4. Audio coaching integrates seamlessly with other app features.

5. Users can control audio settings and preferences.

**\*\*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-AC1-1	Verify that users can access a variety of audio-guided workouts	This test case verifies that the app provides a selection of pre-recorded audio coaching sessions for different types of workouts.	Audio Coaching	High	User is logged in and has a workout selected.	Workout types: Running, Yoga, Strength Training	1. Navigate to the workout selection screen. 2. Choose a workout type. 3. Select an audio coaching option. 4. Verify that audio coaching is available for the selected workout type.	Audio coaching playback completes.	The app should display a list of available audio coaching sessions for the chosen workout type.	Major	High	Functional Testing	Positive
TC-AC2-1	Verify that audio coaching includes motivational messages and coaching cues	This test case ensures that the audio coaching provides encouragement and guidance during workouts.	Audio Coaching	High	User is logged in and has a workout selected with audio coaching enabled.	No test data needed.	1. Start the selected workout. 2. Listen to the audio coaching during the workout. 3. Pay attention to the audio content.	Audio coaching playback completes.	The audio coaching should include motivational messages, cueing for exercises, and guidance on workout intensity.	Major	High	Functional Testing	Positive
TC-AC3-1	Verify that users can choose coaching based on workout type and intensity	This test case validates the app's ability to offer customized audio coaching based on workout preferences.	Audio Coaching	High	User is logged in and has a workout selected.	Workout types: Running, Yoga, Strength Training. Intensities: Beginner, Intermediate, Advanced	1. Navigate to the workout selection screen. 2. Choose a workout type. 3. Select a desired intensity level for the workout. 4. Verify that the available audio coaching options match the selected workout type and intensity.	Audio coaching playback completes.	The app should display different audio coaching options based on the chosen workout type and intensity.	Major	High		

Functional Testing | Positive |

| TC-AC4-1 | Verify that audio coaching integrates seamlessly with other app features | This test case examines the interaction between audio coaching and other functionalities within the app. | Integration Testing | High | User is logged in and has a workout selected with audio coaching enabled. | No test data needed. | 1. Start a workout with audio coaching enabled. 2. Pause the workout. 3. Verify that the audio coaching also pauses. 4. Resume the workout and verify that the audio coaching resumes. | Audio coaching playback completes. | The audio coaching should seamlessly integrate with other features like workout pausing and resuming. | Major | High | Integration Testing | Positive |

| TC-AC5-1 | Verify that users can control audio settings and preferences | This test case ensures that users can customize the audio experience during workouts. | Audio Coaching | High | User is logged in and has a workout selected with audio coaching enabled. | No test data needed. | 1. Start the workout with audio coaching enabled. 2. Access audio settings within the app. 3. Modify volume level, playback speed, or choose a different voice option. 4. Verify that the changes are applied to the audio coaching. | Audio coaching playback completes with the modified settings. | Users should be able to adjust audio volume, playback speed, and other settings like voice selection to personalize the audio experience. | Major | High | Functional Testing | 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:**

## Test Cases for Fitness Tracker App - Integration with Health Apps

**\*\*Project:\*\*** Fitness Tracker App

**\*\*User Story:\*\*** 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:\*\***

1. The app integrates with popular health and wellness apps.
2. Users can sync data from connected health apps.
3. The app provides a consolidated view of health metrics.



- 4. Users receive insights based on combined data from multiple apps.
- 5. Integration settings are easy to manage and configure.

**\*\*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\_001 | Verify that the app integrates with popular health and wellness apps | This test case checks if the app supports integration with commonly used health and wellness applications such as MyFitnessPal, Google Fit, Apple Health, etc. | Integration Testing | High | The app is installed and running. The user is logged in. | List of popular health and wellness apps. | 1. Navigate to the app's integration settings. 2. Check for available options to connect with different health and wellness apps. 3. Select a popular health app from the list. | The integration settings should list the supported apps. | The app should successfully list popular health and wellness apps for integration. | Major | High | Integration Testing | Positive |

| TC\_002 | Verify that users can sync data from connected health apps | This test case validates whether users can sync data like steps, calories burned, heart rate, etc. from connected health apps. | Integration Testing | High | The app is installed and running. The user is logged in. A health app is connected to the fitness tracker app. | Data from a connected health app (e.g., steps taken, calories burned) | 1. Open the app and navigate to the settings. 2. Select the connected health app. 3. Initiate the data synchronization process. | The data from the connected health app should successfully sync with the fitness tracker app. | Major | High | Integration Testing | Positive |

| TC\_003 | Verify that the app provides a consolidated view of health metrics | This test case ensures the app displays data from different connected apps in a unified view, combining metrics like steps, calories burned, sleep duration, and more. | Functional Testing | High | The app is installed and running. The user is logged in. Data from multiple connected health apps is synced. | Synced data from different connected health apps (e.g.,

steps from fitness tracker, sleep data from a sleep tracking app). | 1. Navigate to the app's dashboard or health metrics section. 2. Verify if data from different connected health apps is displayed in a consolidated view. 3. Check if the displayed data is accurate and consistent with the synced data. | The app should display a comprehensive view of health metrics, combining data from different connected apps. | Major | High | Functional Testing | Positive |

| TC\_004 | Verify that users receive insights based on combined data from multiple apps | This test case checks if the app generates personalized insights based on the combined data from different health apps, providing a holistic view of the user's well-being. | Functional Testing | High | The app is installed and running. The user is logged in. Data from multiple connected health apps is synced. | Synced data from different connected health apps (e.g., steps from fitness tracker, sleep data from a sleep tracking app, mood data from a mental health app) | 1. Navigate to the app's insights section. 2. Observe if the app provides insights based on the combined data from different connected apps. 3. Verify if the insights are relevant, personalized, and actionable. | The app should provide users with insights based on the combined data, offering a holistic view of their well-being. | Major | High | Functional Testing | Positive |

| TC\_005 | Verify that integration settings are easy to manage and configure | This test case checks if users can easily manage and configure integration settings for connected health apps, such as enabling/disabling data syncing, choosing specific metrics to sync, etc. | Usability Testing | High | The app is installed and running. The user is logged in. | N/A | 1. Navigate to the app's integration settings. 2. Check the available options for managing and configuring connected health apps. 3. Attempt to enable/disable data syncing, select specific metrics to sync, and modify other settings. | Users should be able to easily manage and configure integration settings without any difficulties. | Minor | Medium | Usability Testing | Positive |

**\*\*Note:\*\*** These test cases provide a starting point for comprehensive testing of the fitness tracker app's integration with health apps. You may need to expand on these test cases, including negative scenarios and edge cases, depending on the specific features and functionality of the app.

-----

**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 for Body Composition Tracking Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Body Composition Tracking**

**\*\*User Story:\*\* 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:\*\***

- 1. Users can log body composition metrics such as weight, body fat percentage, and muscle mass.
- 2. The app calculates and displays BMI based on logged data.
- 3. Users can view trends and history of body composition changes.
- 4. The app provides insights and recommendations based on body composition data.
- 5. Users can set goals for body composition metrics.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\* TC-BodyComp-XXX**

**\*\*Test Case Title:\*\* Verify that...**

**\*\*Test Case Description:\*\* (Brief description of the test case)**

**\*\*Test Suite:\*\* Body Composition Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User has a registered account in the app.

\* User has logged at least one body composition metric.

**\*\*Test Data:\*\* (Specify data required for execution; if none, state "No test data needed")**

**\*\*Test Steps:\*\* (List steps for executing the test case)**

**\*\*Postconditions:\*\* (Any follow-up actions after test execution)**

**\*\*Expected Result:\*\* (Overall expected outcome of the test case)**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* (Positive, Negative, Destructive)**

**\*\*Test Cases:\*\***

**\*\*Test Case ID:\*\* TC-BodyComp-01**

**\*\*Test Case Title:\*\* Verify that users can log body composition metrics such as weight, body fat percentage, and muscle mass.**

**\*\*Test Case Description:\*\* This test case verifies that the user can successfully input their body composition data.**

**\*\*Test Data:\*\***

**\* Weight: 150 lbs**

**\* Body Fat Percentage: 20%**

**\* Muscle Mass: 120 lbs**

**\*\*Test Steps:\*\***

1. Log in as a registered user.
2. Navigate to the "Body Composition" section.
3. Enter the specified body composition metrics.
4. Tap the "Save" button.

**\*\*Postconditions:\*\***

**\* The logged data should be saved and accessible for viewing.**

**\*\*Expected Result:\*\* The app should successfully store the entered body composition data.**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-BodyComp-02**

**\*\*Test Case Title:\*\*** Verify that the app calculates and displays BMI based on logged data.

**\*\*Test Case Description:\*\*** This test case ensures that the app correctly calculates and displays the user's Body Mass Index (BMI) based on their logged weight and height.

**\*\*Test Data:\*\***

\* Weight: 150 lbs

\* Height: 5'10"

**\*\*Test Steps:\*\***

1. Log in as a registered user.
2. Navigate to the "Body Composition" section.
3. View the displayed BMI value.

**\*\*Postconditions:\*\***

\* The BMI value should be calculated correctly based on the user's logged weight and height.

**\*\*Expected Result:\*\*** The app should display the calculated BMI value.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-BodyComp-03

**\*\*Test Case Title:\*\*** Verify that users can view trends and history of body composition changes.

**\*\*Test Case Description:\*\*** This test case verifies that the app displays historical body composition data and trends for each metric.

**\*\*Test Data:\*\***

\* Multiple logged body composition data entries, with varying values over time.

**\*\*Test Steps:\*\***

1. Log in as a registered user.
2. Navigate to the "Body Composition" section.
3. Select the "History" or "Trends" tab.
4. View the historical data and identify trends for weight, body fat percentage, and muscle mass.

**\*\*Postconditions:\*\***

\* The app should display a visually informative graph or chart representing historical data and trends.

**\*\*Expected Result:\*\*** The app should display a history of logged body composition data and highlight any trends in changes.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-BodyComp-04

**\*\*Test Case Title:\*\*** Verify that the app provides insights and recommendations based on body composition data.

**\*\*Test Case Description:\*\*** This test case ensures the app offers relevant insights and recommendations based on user's body composition data.

**\*\*Test Data:\*\***

\* Logged body composition data that falls within a certain range (e.g., high body fat percentage, low muscle mass).

**\*\*Test Steps:\*\***

1. Log in as a registered user.
2. Navigate to the "Body Composition" section.
3. View any insights or recommendations offered by the app.

**\*\*Postconditions:\*\***

\* The insights and recommendations should be personalized based on the user's logged body composition data.

**\*\*Expected Result:\*\*** The app should display specific insights and recommendations relevant to the user's body composition data.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-BodyComp-05

**\*\*Test Case Title:\*\*** Verify that users can set goals for body composition metrics.

**\*\*Test Case Description:\*\*** This test case checks that users can set personalized goals for their body

composition metrics.

**\*\*Test Data:\*\***

\* Goal values for weight, body fat percentage, and muscle mass.

**\*\*Test Steps:\*\***

1. Log in as a registered user.
2. Navigate to the "Body Composition" section.
3. Access the "Goal Setting" feature.
4. Set goals for each body composition metric.

**\*\*Postconditions:\*\***

\* The set goals should be saved and displayed within the app.

**\*\*Expected Result:\*\*** The app should allow users to set personalized goals for each body composition metric.

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 96**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Workout Analysis

**Feature description -**

The app offers detailed workout analysis tools, including performance metrics, trends, and insights into



exercise intensity, duration, and calorie burn. Workout analysis helps users assess the effectiveness of their workouts, identify areas for improvement, and optimize training strategies for better results.

**Userstory title -**

As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

**Acceptance criteria -**

['Users can view detailed performance metrics for each workout.', 'The app provides trends and insights into workout intensity and duration.', 'Users can track calorie burn and other health metrics.', 'The app offers recommendations based on workout analysis.', 'Users can compare current workouts with past performance.']

**Generated Test Cases:**

## Fitness Tracker App - Workout Analysis Test Cases

**Project Name:** Fitness Tracker App

**Feature Name:** Workout Analysis

**User Story Title:** As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

**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-WA-01	Verify that users can view detailed performance metrics for each workout	This test case checks if the user can access and view detailed information about their individual workouts, including duration, distance, calories burned, and heart rate.	Workout Analysis	High	User has completed at least one workout	Sample workout data including duration, distance, calories burned, and heart rate	1. Navigate to the						

"Workouts" section of the app. 2. Select a specific workout from the list. 3. Verify that the workout details page displays the following metrics: duration, distance, calories burned, and heart rate. | The workout details page displays the metrics correctly and is accessible to the user. | No issues observed | Minor | Functional Testing | Positive |

| TC-WA-02 | Verify that the app provides trends and insights into workout intensity and duration | This test case verifies that the app analyzes the user's workout data and displays meaningful trends and insights, including average intensity and duration for different workout types. | Workout Analysis | High | User has completed at least 3 workouts of different types | Sample workout data with varying intensity and duration for different workout types | 1. Navigate to the "Workout Analysis" section of the app. 2. Check if the app displays trend graphs or charts related to workout intensity and duration. 3. Verify that the insights section provides information about average intensity and duration for different workout types. | The app displays trends and insights into workout intensity and duration, providing meaningful information to the user. | Major | High | Functional Testing | Positive |

| TC-WA-03 | Verify that users can track calorie burn and other health metrics | This test case ensures that the app accurately tracks and displays calorie burn and other health metrics like heart rate for each workout. | Workout Analysis | Medium | User has completed at least one workout | Sample workout data including calories burned and heart rate | 1. Navigate to the "Workouts" section of the app. 2. Select a specific workout from the list. 3. Verify that the workout details page displays the calorie burn and heart rate data. | The workout details page displays calorie burn and other health metrics correctly and is accessible to the user. | Minor | Medium | Functional Testing | Positive |

| TC-WA-04 | Verify that the app offers recommendations based on workout analysis | This test case checks if the app uses the analyzed workout data to provide personalized recommendations for improving workout strategies and achieving fitness goals. | Workout Analysis | Medium | User has completed at least 5 workouts | Sample workout data with varying intensity and duration for different workout types | 1. Navigate to the "Workout Analysis" section of the app. 2. Check if the app displays personalized recommendations based on analyzed workout data, such as suggestions for improving intensity or workout frequency. | The app provides personalized recommendations based on workout analysis, offering valuable insights for improving training

strategies. | Minor | Medium | Functional Testing | Positive |

| TC-WA-05 | Verify that users can compare current workouts with past performance | This test case ensures that the app allows users to compare their current workout performance with their historical data for a chosen workout type. | Workout Analysis | High | User has completed at least 3 workouts of the same type | Sample workout data with varying performance metrics for the same workout type (e.g., running) | 1. Navigate to the "Workouts" section of the app. 2. Select a specific workout from the list. 3. Verify that the app allows the user to compare the selected workout data with historical data for the same workout type. | The app allows users to compare current workouts with past performance for the same workout type, helping them track progress and identify areas for improvement. | Minor | High | Functional 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:

## Test Cases for Weather Integration Feature

\*\*Project Name:\*\* Fitness Tracker App

\*\*Feature Name:\*\* Weather Integration

\*\*User Story Title:\*\* As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

\*\*Acceptance Criteria:\*\*

- 1. The app provides current weather conditions and forecasts.
- 2. Users can view weather details for different locations.
- 3. Weather data is updated in real-time.
- 4. The app offers recommendations based on weather conditions.
- 5. Users can receive weather alerts for extreme conditions.

\*\*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\_Weather\_01 | Verify that the app provides current weather conditions. | This test case verifies that the app displays current weather conditions for the user's default location. | Weather Integration | High | - The app is installed and running. - The user is logged in. | No test data needed | 1. Open the Fitness Tracker app. 2. Navigate to the Weather section. 3. Observe the displayed weather information. | No postconditions | The app should display current weather conditions for the user's default location, including temperature, humidity, wind speed, and other relevant details. | Major | High | Functional Testing | Positive |

| TC\_Weather\_02 | Verify that users can view weather details for different locations. | This test case checks that users can access and view weather details for locations other than their default location. | Weather Integration | Medium | - The app is installed and running. - The user is logged in. | - Example city: London, UK. | 1. Open the Fitness Tracker app. 2. Navigate to the Weather section. 3. Use the location search feature to search for a different location (e.g., London, UK). 4. Observe the displayed weather information for the selected location. | No postconditions | The app should display weather details for the selected location, including temperature, humidity, wind speed, and other relevant details. | Major | Medium | Functional Testing | Positive |

| TC\_Weather\_03 | Verify that weather data is updated in real-time. | This test case ensures that weather information is updated frequently, reflecting current conditions. | Weather Integration | High | - The app is installed and running. - The user is logged in. - Weather conditions are known to change. | No test data needed | 1. Open the Fitness Tracker app. 2. Navigate to the Weather section. 3. Record the initial weather data displayed for a specific location. 4. Wait for a short period (e.g., 5 minutes). 5. Refresh the weather data or wait for the automatic update. 6. Observe the updated weather data. | No postconditions | The updated weather data should reflect changes in conditions since the initial recording, indicating real-time updates. | Major | High | Functional Testing | Positive |

| TC\_Weather\_04 | Verify that the app offers recommendations based on weather conditions. | This test case checks if the app provides recommendations tailored to weather conditions. | Weather Integration | Medium | -

The app is installed and running. - The user is logged in. - The weather conditions are known. | - Example condition: Rainy, with temperature below 10°C. | 1. Open the Fitness Tracker app. 2. Navigate to the Weather section. 3. Observe the displayed weather information and recommendations. | No postconditions | The app should provide relevant recommendations based on the weather conditions, such as suggesting indoor workouts or providing tips for staying warm during cold weather. | Major | Medium | Functional Testing | Positive |

| TC\_Weather\_05 | Verify that users can receive weather alerts for extreme conditions. | This test case checks if the app sends alerts for severe weather conditions. | Weather Integration | High | - The app is installed and running. - The user is logged in. - The user has enabled weather alerts. - Extreme weather conditions exist (e.g., heavy rain, thunderstorms). | No test data needed | 1. Open the Fitness Tracker app and enable weather alerts in the settings. 2. Simulate or wait for extreme weather conditions to occur. 3. Observe the app for weather alerts. | No postconditions | The app should send alerts to the user when severe weather conditions are detected in their location, informing them about the potential dangers and providing relevant recommendations. | Critical | High | Functional Testing, System Testing | Positive |

**\*\*Note:\*\*** These test cases are examples, and additional test cases may be required depending on the specific implementation and features of the weather integration within the Fitness Tracker app.

-----

**Data - 98**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

**Feature name -**

Mood Tracking

**Feature description -**

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

**Userstory title -**

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

**Acceptance criteria -**

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and energy levels.']

**Generated Test Cases:**

## Test Cases for Fitness Tracker App - Mood Tracking Feature

**\*\*Project Context:\*\*** Fitness Tracker App - A mobile application designed to track user's physical activity, exercise routines, and health metrics, including mood tracking.

**\*\*User Story:\*\*** As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

**\*\*Acceptance Criteria:\*\***

- 1. Users can log their mood, energy levels, and motivation.
- 2. The app provides insights into mood and energy trends over time.
- 3. Users receive recommendations based on their mood and energy data.
- 4. The app supports journaling for personal reflections on mood and workouts.
- 5. Users can set goals for improving mood and energy levels.

**\*\*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-Mood-01	Verify that users can log their mood, energy levels, and motivation	This test case verifies that users can successfully log their mood, energy level, and motivation before and after workouts using the app interface.	Mood Tracking	High	User is logged into the app and has completed a workout.	Mood: Happy, Energy Level: High, Motivation: Inspired	1. Navigate to the "Mood" section within the app.   2. Select a mood from the available options (e.g., Happy, Sad, Neutral).   3. Select an energy level from the available options (e.g., High, Medium, Low).   4. Enter a short motivation description (e.g., "Feeling energized and ready to go!").   5. Submit the mood log.	Mood log is successfully saved to the user's profile.	Mood log is successfully saved, and the user's mood, energy level, and motivation are displayed on the screen.	Minor	High	Functional Testing	Positive
TC-Mood-02	Verify that the app provides insights into mood and energy trends over time	This test case verifies that the app displays a visual representation of mood and energy levels over a specified period (e.g., daily, weekly, monthly).	Mood Tracking	High	User has logged multiple mood entries over a period of time.		1. Navigate to the "Trends" section within the app.   2. Select the desired time period for data visualization.   3. Observe the charts or graphs displaying mood and energy levels over time.	Charts or graphs display a clear visualization of mood and energy trends over the selected time period.		Minor	High	Functional Testing	Positive



| TC-Mood-03 | Verify that users receive recommendations based on their mood and energy data | This test case verifies that the app provides personalized recommendations based on the user's mood and energy data, suggesting activities or actions to improve mental well-being. | Mood Tracking | Medium | User has logged multiple mood entries and has chosen "Enable Recommendations" in their settings. | User has logged mood entries indicating consistent low energy levels. | 1. Navigate to the "Recommendations" section within the app. <br> 2. Observe the displayed recommendations based on the user's mood and energy data. | Recommendations are displayed, suggesting activities or actions to improve mood and energy levels (e.g., "Try a calming activity like yoga," "Consider taking a short break"). | Minor | Medium | Functional Testing | Positive |

| TC-Mood-04 | Verify that the app supports journaling for personal reflections on mood and workouts | This test case verifies that the app allows users to write journal entries associated with their mood and workouts, providing a platform for self-reflection and understanding the impact of exercise on mental well-being. | Mood Tracking | Medium | User has logged a mood entry after a workout. | User has a workout logged and a mood entry associated with it. | 1. Navigate to the "Journal" section within the app. <br> 2. Select the workout and mood entry to view. <br> 3. Access the journaling feature and write a reflection about the workout and how it impacted their mood. | Journal entry is successfully saved and associated with the selected workout and mood log. | Minor | Medium | Functional Testing | Positive |

| TC-Mood-05 | Verify that users can set goals for improving mood and energy levels | This test case verifies that users can set personalized goals related to their mood and energy levels, promoting proactive self-care and mental well-being through exercise. | Mood Tracking | Medium | User has accessed the "Goals" section within the app. | Goal: Increase positive mood days by 20%. | 1. Navigate to the "Goals" section within the app. <br> 2. Select the option to set a mood or energy-related goal. <br> 3. Choose the desired goal type (e.g., increase positive mood days, improve energy levels). <br> 4. Enter the specific goal details (e.g., target percentage of positive mood days). <br> 5. Save the goal. | Goal is successfully saved to the user's profile, and the app tracks progress towards achieving the set goal. | Minor | Medium | Functional Testing | Positive |

**\*\*Notes:\*\***

- \* These test cases are examples and should be adapted to cover the specific functionality and requirements of the fitness tracker app.
- \* The test data should be modified to include various user scenarios and edge cases.
- \* Additional test cases may be required to cover all aspects of the mood tracking feature and ensure thorough testing.

This set of test cases aims to contribute to a comprehensive quality assurance process for the Fitness Tracker App. By identifying potential defects early in the development cycle, these tests help to ensure the functionality, usability, and user satisfaction of the mood tracking feature.

-----

## **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

\*\*Project:\*\* Recipe Sharing Platform

\*\*User Story:\*\* As a user, I want to create and share my own recipes so that I can showcase my culinary  
creations.

\*\*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-RC-01	Verify that users can input and save ingredients and cooking instructions	This test case validates that users can successfully enter and save a list of ingredients and detailed cooking instructions for their recipes.	Recipe Creation	High	User is logged in and on the recipe creation page	Recipe Title: "Chocolate Chip Cookies", Ingredients: Flour, Sugar, Butter, Chocolate Chips. Instructions: Preheat oven to 350°F. Combine ingredients. Bake for 10 minutes.	1. Navigate to the "Create Recipe" page. 2. Enter the recipe title "Chocolate Chip Cookies". 3. Enter the ingredient list and cooking instructions as specified in Test Data. 4. Click on "Save Recipe".	Recipe is saved in the user's profile.	Recipe is saved with correct title, ingredients, and instructions.	Major	High	Functional Testing	Positive

| TC-RC-02 | Verify that users can upload photos for their recipes | This test case verifies that users can select and upload photos from their device to accompany their recipes. | Recipe Creation | High | User is logged in and on the recipe creation page, a photo is available on the device | Photo: "ChocolateChipCookies.jpg" (image of cookies) | 1. Navigate to the "Create Recipe" page. 2. Enter a recipe title and details. 3. Click on "Upload Photo". 4. Select the "ChocolateChipCookies.jpg" image. 5. Click on "Save Recipe". | Photo is associated with the recipe in the user's profile. | The uploaded photo is successfully associated with the recipe. | Major | High | Functional Testing | Positive |

| TC-RC-03 | Verify that recipes are saved in the user's profile | This test case validates that newly created recipes are correctly stored and accessible within the user's profile. | Recipe Creation | High | User is logged in and has created a recipe | Recipe Title: "Chicken Stir Fry" | 1. Create a recipe with the title "Chicken Stir Fry". 2. Navigate to the user's profile. 3. Search for the "Chicken Stir Fry" recipe. | Recipe is found in the user's profile with all the entered details. | The created recipe is successfully saved and displayed in the user's profile. | Major | High | Functional Testing | Positive |

| TC-RC-04 | Verify that recipes are accessible to other users on the platform | This test case verifies that created recipes are publicly viewable by other users of the platform. | Recipe Creation | High | User has created a recipe, another user is logged in | Recipe Title: "Spaghetti Bolognese" | 1. User A creates a recipe titled "Spaghetti Bolognese". 2. User B logs in and searches for "Spaghetti Bolognese". | User B can view the "Spaghetti Bolognese" recipe. | User B can view the recipe created by User A without any restrictions. | Major | High | Functional Testing | Positive |

| TC-RC-05 | Verify that users receive a confirmation upon successful recipe submission | This test case validates that upon successful recipe creation, the user receives a confirmation message or notification. | Recipe Creation | High | User is logged in and on the recipe creation page | No test data needed | 1. Navigate to the "Create Recipe" page. 2. Enter the recipe details. 3. Click on "Save Recipe". | User receives a confirmation message indicating successful recipe submission. | User receives a clear and visible confirmation message after submitting the recipe. | Major | High | Functional Testing | 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 Discovery Test Cases

**\*\*Project Name:\*\* Recipe Sharing Platform**

**\*\*User Story Title:\*\*** As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC\_RecipeDiscovery\_01

**\*\*Test Case Title:\*\*** Verify that users can search for recipes using keywords.

**\*\*Test Case Description:\*\*** This test case validates that the platform allows users to enter keywords in the search bar and retrieve relevant search results.

**\*\*Test Suite:\*\*** Recipe Search

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* The user is logged into the platform.

\* The search bar is visible on the platform.

**\*\*Test Data:\*\***

\* Keyword: "Chicken Curry"

**\*\*Test Steps:\*\***

1. Navigate to the recipe search page.

2. Enter the keyword "Chicken Curry" in the search bar.

3. Click the search button.

**\*\*Postconditions:\*\***

\* The search results page is displayed.

\* The search results page displays recipes relevant to "Chicken Curry".

**\*\*Expected Result:\*\*** The platform retrieves and displays recipes that contain the keyword "Chicken Curry".

**\*\*Severity:\*\*** Critical

**\*\*Priority:\*\*** High

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RecipeDiscovery\_02**

**\*\*Test Case Title:\*\* Verify that search results are relevant and accurately match the search criteria.**

**\*\*Test Case Description:\*\* This test case ensures that the search functionality retrieves recipes that match the entered keywords and ingredients.**

**\*\*Test Suite:\*\* Recipe Search**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* The user is logged into the platform.

\* The search bar and ingredient filters are visible on the platform.

**\*\*Test Data:\*\***

\* Keyword: "Pasta"

\* Ingredient: "Tomato"

**\*\*Test Steps:\*\***

1. Navigate to the recipe search page.

2. Enter the keyword "Pasta" in the search bar.

3. Select "Tomato" from the ingredient filter options.

4. Click the search button.

**\*\*Postconditions:\*\***

\* The search results page is displayed.

\* The search results page displays recipes containing "Pasta" and "Tomato".

**\*\*Expected Result:\*\* The search results display only recipes that contain both "Pasta" and "Tomato" as ingredients.**

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RecipeDiscovery\_03**

**\*\*Test Case Title:\*\* Verify that users can filter search results by cuisine type.**

**\*\*Test Case Description:\*\* This test case checks the functionality of the cuisine type filter, ensuring users can refine their search based on specific culinary styles.**

**\*\*Test Suite:\*\* Recipe Search**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* The user is logged into the platform.

\* The search bar and cuisine type filters are visible on the platform.

**\*\*Test Data:\*\***

\* Keyword: "Pizza"

\* Cuisine Type: "Italian"

**\*\*Test Steps:\*\***

1. Navigate to the recipe search page.

2. Enter the keyword "Pizza" in the search bar.

3. Select "Italian" from the cuisine type filter options.

4. Click the search button.

**\*\*Postconditions:\*\***

\* The search results page is displayed.

\* The search results page displays only Italian pizza recipes.

**\*\*Expected Result:\*\* The search results only include pizza recipes categorized as "Italian".**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**



**\*\*Test Case ID:\*\* TC\_RecipeDiscovery\_04**

**\*\*Test Case Title:\*\* Verify that recipes in the search results are accessible and detailed.**

**\*\*Test Case Description:\*\* This test case ensures that the displayed search results include essential recipe information like title, ingredients, instructions, and images, allowing users to easily access the details of each recipe.**

**\*\*Test Suite:\*\* Recipe Search**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* The user is logged into the platform.

\* The search bar is visible on the platform.

**\*\*Test Data:\*\***

\* Keyword: "Brownies"

**\*\*Test Steps:\*\***

1. Navigate to the recipe search page.
2. Enter the keyword "Brownies" in the search bar.
3. Click the search button.
4. Click on the first search result for a brownie recipe.

**\*\*Postconditions:\*\***

\* The selected recipe page is displayed.

\* The recipe page includes a clear title, list of ingredients, detailed instructions, and an image or video of the final dish.

**\*\*Expected Result:\*\* The selected recipe page displays all the essential information for users to understand and prepare the recipe.**

**\*\*Severity:\*\* Critical**

**\*\*Priority:\*\* High**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RecipeDiscovery\_05**

**\*\*Test Case Title:\*\* Verify that users can save search results for future reference.**

**\*\*Test Case Description:\*\* This test case checks the functionality of saving search results, enabling users to easily revisit relevant recipes at a later time.**

**\*\*Test Suite:\*\* Recipe Search**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* The user is logged into the platform.

\* The search bar and "Save Search" functionality are visible on the platform.

**\*\*Test Data:\*\***

\* Keyword: "Vegan Soup"

**\*\*Test Steps:\*\***

1. Navigate to the recipe search page.
2. Enter the keyword "Vegan Soup" in the search bar.
3. Click the search button.
4. Click on the "Save Search" button.

**\*\*Postconditions:\*\***

\* The user is redirected to their saved searches page.

\* The saved search "Vegan Soup" is displayed on their saved searches page.

**\*\*Expected Result:\*\* The user can successfully save the "Vegan Soup" search results and access them later from their saved searches page.**

**\*\*Severity:\*\* Major**

**\*\*Priority:\*\* Medium**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

-----