

# Generated Test Cases via gemini-1.5-flash LLM Model and Few-shot Prompting Technique

---

## Data - 1

### Project name -

Social Media Platform

### Project description -

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

### Feature name -

User Profiles

### Feature description -

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

### Userstory title -

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

### Acceptance criteria -

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

## Generated Test Cases:

## Test Cases for User Profile Feature

**\*\*Project Context:\*\***

\* Project Name: Social Media Platform

\* Project Description: 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.

\* User Story 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:

1. Users can add and edit their name, bio, profile picture, location, and interests.
2. The profile picture should support standard image formats.
3. Users can set privacy settings for each field.
4. The profile must be saved successfully.
5. Users should receive a confirmation upon saving their 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	Type of Testing	Test Case Approach	
TC-UP-01	Verify that users can add and edit their name	Ensure the system allows users to input and				

modify their name in the profile settings. | User Profiles | High | User is logged in | User name: "John Doe", Modified name: "Jane Doe" | 1. Go to "Profile Settings" 2. Click on "Edit Profile" 3. Modify the "Name" field to "Jane Doe" 4. Click "Save" | Profile updated with the new name | The system updates the user's name in the profile successfully and displays the updated name on the profile page. | Minor | Functional | Positive |

| TC-UP-02 | Verify that profile pictures support standard image formats | Ensure the system supports common image formats (e.g., JPG, PNG, GIF) for profile pictures. | User Profiles | High | User is logged in | Image 1: "profile\_pic.jpg" (JPG), Image 2: "profile\_pic.png" (PNG), Image 3: "profile\_pic.gif" (GIF) | 1. Go to "Profile Settings" 2. Click on "Edit Profile Picture" 3. Upload "profile\_pic.jpg" 4. Upload "profile\_pic.png" 5. Upload "profile\_pic.gif" | Profile picture updated | The system successfully uploads and displays the profile picture in all supported formats. | Major | Functional | Positive |

| TC-UP-03 | Verify that users can set privacy settings for each profile field | Ensure the system provides options to control the visibility of each profile field. | User Profiles | Medium | User is logged in | Profile field: "Location", Privacy settings: "Public", "Friends Only", "Private" | 1. Go to "Profile Settings" 2. Click on "Edit Profile" 3. Select "Location" field 4. Choose "Friends Only" privacy setting 5. Click "Save" | Privacy settings updated | The system updates the privacy settings for the selected field and restricts visibility accordingly. | Minor | Functional | Positive |

| TC-UP-04 | Verify that the profile is saved successfully | Ensure the system saves all changes made to the profile. | User Profiles | High | User is logged in | Modified profile details: Name, bio, profile picture, location, interests | 1. Go to "Profile Settings" 2. Click on "Edit Profile" 3. Modify various fields (Name, Bio, Profile Picture, Location, Interests) 4. Click "Save" | Profile data saved | The system saves all the changes made to the profile, and the updated data is reflected on the profile page. | Major | Functional | Positive |

| TC-UP-05 | Verify that users receive confirmation upon saving their profile | Ensure the system provides a confirmation message upon successful profile update. | User Profiles | Medium | User is logged in | No test data needed | 1. Go to "Profile Settings" 2. Click on "Edit Profile" 3. Modify any field 4. Click "Save" | Profile updated confirmation message displayed | The system displays a confirmation message indicating the profile has been saved successfully. | Minor | Functional | 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 Social Media Platform News Feed Feature

**\*\*Project Context:\*\***

\* 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.

\* 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.

\* User Story 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:\*\***

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 Creation Template:\*\*

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data
Test Steps	Postconditions	Expected Result	Severity	Type of Testing	Test Case Approach	
---	---	---	---	---	---	---

| TC\_NF\_01 | Verify that the news feed displays posts from followed accounts | This test case verifies that the news feed displays posts only from accounts that the user is following. It checks that posts from unfollowed accounts are not displayed. | News Feed | High | User is logged in, User has followed at least one account | No test data needed | 1. Navigate to the news feed. 2. Verify that posts are displayed from followed accounts. 3. Verify that posts from unfollowed accounts are not displayed. | No postconditions | The news feed only displays posts from followed accounts. | Major | Functional Testing | Positive |

| TC\_NF\_02 | Verify that content is displayed in chronological or algorithmic order | This test case verifies that the content is displayed in either chronological or algorithmic order, depending on the chosen setting. It checks that posts are displayed in the expected order based on the selected sorting method. | News Feed | High | User is logged in, User has followed at least two accounts | Two posts from different accounts with different timestamps | 1. Navigate to the news feed. 2. Verify that posts are displayed in chronological order (most recent first). 3. Change the sorting method to algorithmic. 4. Verify that posts are now displayed in the algorithm-determined order. | No postconditions | Content is displayed in the chosen order (chronological or algorithmic). | Major | Functional Testing | Positive |

| TC\_NF\_03 | Verify that users can interact with posts (like, comment, share) | This test case verifies that users can like, comment, and share posts displayed in the news feed. It checks that the interaction functionality is working correctly. | News Feed | High | User is logged in, A post is available in the news feed | No test data needed | 1. Navigate to the news feed. 2. Select a post. 3. Like the post. 4. Comment on the post. 5. Share the post. | Like, comment, and share actions are reflected on the post. | Major | Functional Testing | Positive |

| TC\_NF\_04 | Verify that the news feed updates in real-time | This test case verifies that the news feed updates automatically and displays new posts from followed accounts in real-time. It checks that the feed is responsive to new activity. | News Feed | High | User is logged in, A new post is made by a followed account | No test data needed | 1. Navigate to the news feed. 2. Observe the feed for existing posts. 3. Have a followed account post a new update. 4. Verify that the new post appears in the feed within a reasonable time frame. | No postconditions | The news feed updates in real-time with new posts from followed accounts. | Major | Functional Testing | Positive |

| TC\_NF\_05 | Verify that users can filter content based on type or topic | This test case verifies that users can

filter content in the news feed based on type or topic, such as filtering for photos, videos, articles, or specific interests. It checks that the filtering functionality is working correctly and only relevant content is displayed. | News Feed | High | User is logged in, The news feed contains content of different types and topics | No test data needed | 1. Navigate to the news feed. 2. Select the filter option. 3. Choose a specific content type (e.g., photos, videos, articles). 4. Verify that only posts of the selected type are displayed. 5. Repeat steps 2-4 with a specific topic filter. 6. Verify that only posts related to the chosen topic are displayed. | No postconditions | The news feed displays filtered content based on the chosen type or topic. | Major | Functional Testing | 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 Social Media Platform Messaging Feature

#### \*\*Project Context:\*\*

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.

User Story 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.

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



**\*\*Test Case ID:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module)

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

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

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

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

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

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

**\*\*Test Case Title:\*\*** Verify that users can send text messages in a private conversation.

**\*\*Test Case Description:\*\*** This test case checks if two users can successfully initiate a private chat and exchange text messages.

**\*\*Test Suite:\*\*** Messaging

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

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

- Two users are logged in
- Both users have added each other as friends

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

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

1. User 1 navigates to the messaging section.
2. User 1 selects User 2 from the list of friends.
3. User 1 types a text message and sends it to User 2.
4. User 2 receives the message and replies with a text message.

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

- The message is successfully sent and received between the two users.

**\*\*Expected Result:\*\*** User 2 receives the message and the chat history is updated with both messages.

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

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that users can send images in private messages.

**\*\*Test Case Description:\*\*** This test case ensures that users can successfully send images within a private chat.

**\*\*Test Suite:\*\*** Messaging

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

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

- Two users are logged in
- Both users have added each other as friends

**\*\*Test Data:\*\*** An image file.

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

1. User 1 navigates to the messaging section.
2. User 1 selects User 2 from the list of friends.
3. User 1 selects the image attachment option.
4. User 1 chooses the image file and sends it to User 2.

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

- The image is sent successfully to User 2.

**\*\*Expected Result:\*\*** User 2 receives the image in the chat and it is displayed correctly.

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

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that users can send videos in private messages.

**\*\*Test Case Description:\*\*** This test case verifies that users can send video files through the private messaging feature.

**\*\*Test Suite:\*\*** Messaging

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

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

- Two users are logged in
- Both users have added each other as friends

**\*\*Test Data:\*\*** A video file.

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

1. User 1 navigates to the messaging section.
2. User 1 selects User 2 from the list of friends.

3. User 1 selects the video attachment option.

4. User 1 chooses the video file and sends it to User 2.

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

- The video is sent successfully to User 2.

**\*\*Expected Result:\*\*** User 2 receives the video in the chat and can play it without errors.

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

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that messages show read receipts.

**\*\*Test Case Description:\*\*** This test case checks if the messaging system displays read receipts for messages sent in a private chat.

**\*\*Test Suite:\*\*** Messaging

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

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

- Two users are logged in

- Both users have added each other as friends

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

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

1. User 1 sends a text message to User 2.

2. User 2 reads the message.

3. User 1 checks the chat for read receipt indication.

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

- User 1 sees the read receipt indicator next to the message.

**\*\*Expected Result:\*\*** User 1 should observe a read receipt icon or similar indication next to the message, confirming that User 2 has read it.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that users receive a notification when they receive a new message in a private chat.

**\*\*Test Suite:\*\*** Messaging

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

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

- Two users are logged in
- Both users have added each other as friends
- User 2 is logged in and actively using the platform.

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

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

1. User 1 sends a message to User 2.
2. Observe User 2's notification center.

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

- User 2 receives a notification about the new message.

**\*\*Expected Result:\*\*** User 2's notification center displays a notification indicating a new message from User 1.

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

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

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

-----

## **Data - 4**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Content Sharing

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

## Test Cases for Content Sharing Feature on Social Media Platform

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

\*\*Feature Name:\*\* Content Sharing

\*\*User Story Title:\*\* 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	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Type of Testing	Test Case Approach
TC-CS-01	Verify that users can create and share text posts	This test case ensures that users can create and share text posts with their followers or publicly.	Content Sharing	High	- User is logged in	Text post content	1. Navigate to the post creation section. 2. Enter a text post message. 3. Select the audience (followers or public). 4. Click the "Share" button.	- The post is created and shared.	The system creates and shares the text post with the selected audience.	Major	Functional	Positive

| TC-CS-02 | Verify that users can create and share photos | This test case verifies that users can upload and share photos with their followers or publicly. | Content Sharing | High | - User is logged in | Photo file | 1. Navigate to the post creation section. 2. Select the "Photo" option. 3. Choose a photo from the device. 4. Select the audience (followers or public). 5. Click the "Share" button. | - The photo is uploaded and shared. | The system uploads and shares the selected photo with the selected audience. | Major | Functional | Positive |

| TC-CS-03 | Verify that users can create and share videos | This test case validates that users can upload and share videos with their followers or publicly. | Content Sharing | High | - User is logged in | Video file | 1. Navigate to the post creation section. 2. Select the "Video" option. 3. Choose a video from the device. 4. Select the audience (followers or public). 5. Click the "Share" button. | - The video is uploaded and shared. | The system uploads and shares the selected video with the selected audience. | Major | Functional | Positive |

| TC-CS-04 | Verify that users can edit or delete their posts | This test case ensures that users can edit or delete previously shared text posts, photos, or videos. | Content Sharing | Medium | - User is logged in | Previously shared post | 1. Navigate to the shared post. 2. Click the "Edit" or "Delete" option. 3. Perform the desired action (edit or delete). | - The post is updated or removed. | The system allows users to edit or delete their previously shared posts. | Minor | Functional | Positive |

| TC-CS-05 | Verify that posts support hashtags and tagging | This test case validates that users can add hashtags and tag other users in their posts. | Content Sharing | Medium | - User is logged in | Post content with hashtags and tags | 1. Navigate to the post creation section. 2. Enter a text post message with hashtags and tags. 3. Select the audience (followers or public). 4. Click the "Share" button. | - The post is created and displayed with hashtags and tags. | The system allows users to use hashtags and tag other users in their posts, making them easily searchable and discoverable. | Minor | Functional | 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:

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

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

**\*\*Test Case Description:\*\*** This test case checks if users can choose from the three available privacy settings (public, private, restricted) for their profiles.

**\*\*Test Suite:\*\*** Privacy Settings

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

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

- User is logged in to the social media platform.
- User has access to their profile settings.

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

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

1. Navigate to the user's profile settings.
2. Locate the privacy settings section.
3. Verify that the options "Public," "Private," and "Restricted" are available.
4. Select each privacy setting option one by one.
5. Verify that the chosen setting is applied to the user's profile.

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

- User's profile is set to the chosen privacy setting.

**\*\*Expected Result:\*\*** The system should successfully apply the chosen privacy setting to the user's profile.

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case checks if users can modify their privacy settings without restrictions.

**\*\*Test Suite:\*\*** Privacy Settings

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

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

- User is logged in to the social media platform.
- User's profile is set to a specific privacy setting (e.g., Private).

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

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

1. Navigate to the user's profile settings.
2. Locate the privacy settings section.
3. Change the user's privacy setting from "Private" to "Public."
4. Confirm the change by saving the settings.
5. Verify that the user's profile is now set to "Public."
6. Repeat steps 2-5 to change the privacy setting back to "Private" or "Restricted."

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

- User's profile privacy setting is successfully changed.

**\*\*Expected Result:\*\*** The system should allow users to change their privacy settings at any time without any limitations.

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that users can control visibility of individual posts.

**\*\*Test Case Description:\*\*** This test case checks if users can choose different privacy settings for individual posts, overriding their default profile privacy.

**\*\*Test Suite:\*\*** Privacy Settings

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

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

- User is logged in to the social media platform.
- User has created a post.

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

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

1. Create a new post on the platform.

2. Locate the privacy settings option within the post creation interface.
3. Select a privacy setting different from the user's default profile setting.
4. Post the content.
5. Verify that the post is only visible to the selected audience.

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

- The newly created post is visible according to the selected privacy setting.

**\*\*Expected Result:\*\*** The system should allow users to control the visibility of individual posts separately from their profile privacy settings.

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that the system respects privacy settings immediately.

**\*\*Test Case Description:\*\*** This test case checks if the system applies the selected privacy settings immediately without any delay.

**\*\*Test Suite:\*\*** Privacy Settings

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

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

- User is logged in to the social media platform.
- User's profile is set to "Private."

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

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

1. Navigate to the user's profile.
2. Attempt to view the user's profile as an un-connected user.
3. Verify that the profile is not accessible to the un-connected user.
4. Navigate to the user's profile settings and change the privacy setting to "Public."

5. Refresh the browser and attempt to view the user's profile as an un-connected user again.

6. Verify that the user's profile is now accessible.

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

- The privacy settings are applied immediately.

**\*\*Expected Result:\*\*** The system should immediately apply the chosen privacy setting to the user's profile and content.

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case checks if the system notifies users of changes made to their privacy settings, ensuring awareness.

**\*\*Test Suite:\*\*** Privacy Settings

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

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

- User is logged in to the social media platform.

- User's profile is set to "Restricted."

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

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

1. Navigate to the user's profile settings and change the privacy setting to "Private."

2. Verify that the user receives a notification informing them of the change in privacy settings.

3. Check if the notification contains details about the new privacy setting and its implications.

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

- The user receives a notification about the privacy setting change.

**\*\*Expected Result:\*\*** The system should notify the user of any changes made to their privacy settings,

providing clear information about the update.

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

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

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

-----

## **Data - 6**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Notifications

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

Generated Test Cases:

## Test Cases for Social Media Platform - Notifications

\*\*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.

\*\*Acceptance Criteria:\*\*

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

\*\*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	Type of Testing	Test Case Approach
TC-NOT-01	Verify that users receive notifications for likes, comments, and shares.	This test case verifies that users receive notifications when someone likes, comments on, or shares their post.	Notifications	High	User is logged in, User has created a post	A sample post with content	1. Log in as the user. 2. Create a new					

post. 3. Log in as another user. 4. Like, comment on, and share the user's post. 5. Observe the notifications received by the user. | User has received a notification for each action. | The user receives notifications for likes, comments, and shares on their post. | Major | Functional Testing | Positive |

| TC-NOT-02 | Verify that users are notified of new friend requests and mentions. | This test case verifies that users receive notifications when someone sends them a friend request or mentions them in a post. | Notifications | High | User is logged in | No test data needed | 1. Log in as the user. 2. Log in as another user. 3. Send a friend request to the user. 4. Mention the user in a post. 5. Observe the notifications received by the user. | User has received a notification for each action. | The user receives notifications for new friend requests and mentions. | Major | Functional Testing | Positive |

| TC-NOT-03 | Verify that notifications are displayed in real-time. | This test case verifies that notifications are displayed immediately after an event occurs. | Notifications | High | User is logged in, User has a new notification | No test data needed | 1. Log in as the user. 2. Ensure the user has no active notifications. 3. Have another user interact with the user's profile (e.g., like a post, send a friend request). 4. Immediately observe the notifications section for any new notifications. | User receives the notification in the notifications section within 10 seconds. | Major | Performance Testing | Positive |

| TC-NOT-04 | Verify that users can customize notification preferences. | This test case verifies that users can choose which types of notifications they want to receive. | Notifications | Medium | User is logged in | No test data needed | 1. Log in as the user. 2. Navigate to notification settings. 3. Customize notification preferences (e.g., disable notifications for likes, enable notifications for mentions). 4. Save changes. 5. Observe the notifications received by the user after interacting with their profile. | User's notification preferences are saved and applied to future notifications. | Minor | Functional Testing | Positive |

| TC-NOT-05 | Verify that notifications include a link to the relevant activity. | This test case verifies that notifications contain a clickable link that directs users to the relevant activity. | Notifications | Medium | User is logged in, User has a notification | No test data needed | 1. Log in as the user. 2. Ensure the user has a new notification. 3. Click on the notification. 4. Observe the page the user is redirected to. | User is redirected to the relevant page corresponding to the notification. | Minor | 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 Social Media Platform Search Functionality

\*\*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 1:\*\***

**\*\*Test Case ID:\*\* TC\_Search\_Keyword**

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

**\*\*Test Case Description:\*\* This test case aims to verify that users can search for content on the platform using keywords. It will involve searching for a specific keyword and checking if the search results display relevant posts, profiles, and topics related to the keyword.**

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

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

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

- User is logged in to the platform.

**\*\*Test Data:\*\* Keyword: "travel"**

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

1. Navigate to the search bar on the platform.
2. Enter the keyword "travel" in the search bar.
3. Click the search button or press Enter.

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

- Search results page is displayed.

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

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Search\_Hashtag

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

**\*\*Test Case Description:\*\*** This test case verifies that users can search for content using hashtags. It will involve searching for a specific hashtag and checking if the search results display relevant posts and profiles associated with that hashtag.

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

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

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

- User is logged in to the platform.

**\*\*Test Data:\*\*** Hashtag: #food

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

1. Navigate to the search bar on the platform.
2. Enter the hashtag #food in the search bar.
3. Click the search button or press Enter.

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

- Search results page is displayed.

**\*\*Expected Result:\*\*** The search results page displays relevant posts and profiles associated with the hashtag #food.

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

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

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

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

**\*\*Test Case ID:\*\* TC\_Search\_Results\_Order**

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

**\*\*Test Case Description:\*\* This test case ensures that search results are displayed in order of relevance. It will involve searching for a specific keyword and observing the order of search results based on their relevance to the query.**

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

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

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

- User is logged in to the platform.

**\*\*Test Data:\*\* Keyword: "photography"**

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

1. Navigate to the search bar on the platform.
2. Enter the keyword "photography" in the search bar.
3. Click the search button or press Enter.
4. Observe the order of search results based on their relevance to the keyword "photography".

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

- Search results page is displayed.

**\*\*Expected Result:\*\* The search results page displays the most relevant posts, profiles, and topics related to "photography" at the top, followed by less relevant results.**

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

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

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

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

**\*\*Test Case ID:\*\* TC\_Search\_Filter**

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

**\*\*Test Case Description:\*\*** This test case ensures users can filter search results based on the content type. It will involve searching for a specific keyword and then filtering the results by specific content types like posts, profiles, or topics.

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

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

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

- User is logged in to the platform.

**\*\*Test Data:\*\*** Keyword: "music"

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

1. Navigate to the search bar on the platform.
2. Enter the keyword "music" in the search bar.
3. Click the search button or press Enter.
4. Select the filter option for "Posts" to view only music related posts.
5. Select the filter option for "Profiles" to view only music-related profiles.

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

- Search results page is displayed with filtered content.

**\*\*Expected Result:\*\*** The search results page displays only the selected content type (posts or profiles) related to the keyword "music".

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_Search\_Suggestions

**\*\*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. It will involve typing a partial keyword in the search bar and observing if the system displays relevant suggestions.

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

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

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

- User is logged in to the platform.

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

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

1. Navigate to the search bar on the platform.
2. Start typing a partial keyword, for example, "photo".
3. Observe if the system provides suggestions related to "photo".

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

- Search bar displays suggestions.

**\*\*Expected Result:\*\*** The system displays relevant suggestions as the user types, for example, "photography", "photos", "photo editing", etc.

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

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

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

-----

## **Data - 8**

### **Project name -**

Social Media Platform

### **Project description -**

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

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

**Feature name -**

Hashtags and Tagging

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

## Test Cases for Hashtags and Tagging Feature

**Project Name:** Social Media Platform

**Feature Name:** Hashtags and Tagging

**Test Case 1:**

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

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

**Test Case Description:** This test case verifies that users can successfully add hashtags to their posts. The system should allow users to enter hashtags using the '#' symbol and display them correctly.

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

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

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

- User is logged in
- User has created a post

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

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

1. Go to the post creation area.
2. Type a message and include a hashtag using '#' (e.g., #travel).
3. Submit the post.

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

- The post is published with the hashtag.
- The hashtag is displayed correctly.

**\*\*Expected Result:\*\*** The system should successfully add the hashtag to the post and display it prominently.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that users can tag other profiles, brands, or locations in their posts using the '@' symbol. The system should display the tagged entities correctly and link them to their respective profiles.

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

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



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

- User is logged in
- User has created a post
- A profile, brand, or location is publicly available

**\*\*Test Data:\*\*** Publicly available profile/brand/location name

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

1. Go to the post creation area.
2. Type a message and tag a profile, brand, or location using '@' (e.g., @example\_brand).
3. Submit the post.

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

- The post is published with the tagged entity.
- The tagged entity is displayed correctly and linked to its respective profile.

**\*\*Expected Result:\*\*** The system should successfully tag the specified entity in the post and link it to its profile.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that hashtags and tags are clickable and lead to a relevant content feed or profile. Clicking a hashtag should display posts related to that hashtag. Clicking a tag should lead to the tagged profile, brand, or location.

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

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

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

- User is logged in
- Posts exist with hashtags and tags

**\*\*Test Data:\*\*** Hashtags and tags from existing posts

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

1. Go to a post containing a hashtag or tag.
2. Click on the hashtag or tag.

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

- A new page or section is displayed.

**\*\*Expected Result:\*\*** Clicking a hashtag should display a feed of relevant posts using that hashtag. Clicking a tag should redirect to the tagged profile, brand, or location.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that users can search for content using hashtags. The search function should allow users to input hashtags and display relevant posts.

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

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

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

- User is logged in
- Posts exist with hashtags

**\*\*Test Data:\*\*** Popular hashtags

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

1. Go to the search bar.
2. Enter a hashtag (e.g., #fashion).
3. Click on the search icon.

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

- A search results page is displayed.

**\*\*Expected Result:\*\*** The system should display posts containing the searched hashtag.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that the system suggests popular hashtags as users type them in the post creation area. The suggestions should be relevant and based on the user's typed text.

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

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

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

- User is logged in
- User is creating a post

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

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

1. Go to the post creation area.
2. Start typing a hashtag (e.g., #tra).

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

- A list of suggestions appears.

**\*\*Expected Result:\*\*** The system should display a list of popular and relevant hashtags based on the user's input.

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

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

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

-----

## **Data - 9**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Emojis and Reactions

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

['Users can react to posts with a variety of emojis.', 'Emojis can be used in comments and messages.', 'Users

can see a count of reactions on posts.', 'The platform should support a wide range of emojis.', 'Users can remove or change their reactions.']

## **Generated Test Cases:**

## Test Cases for Emojis and Reactions Feature:

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

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

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

**\*\*Test Case Description:\*\*** This test case ensures that the platform offers a diverse selection of emojis for users to react to posts, allowing for various emotional expressions.

**\*\*Test Suite:\*\*** Emojis and Reactions

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

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

- User is logged in
- A post is created

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

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

1. Navigate to a post.
2. Click on the reaction button (e.g., "Like" or "React").
3. Observe the displayed emoji options.
4. Select different emojis to react to the post.

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

- The user's selected emoji is displayed on the post.
- The emoji count updates accordingly.

**\*\*Expected Result:\*\*** The system successfully displays a variety of emojis for users to react to posts. The

user's selected emoji is displayed on the post, and the emoji count is updated accurately.

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case validates that users can use emojis in both comments and private messages, enhancing communication and expression.**

**\*\*Test Suite:\*\* Emojis and Reactions**

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

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

- User is logged in
- A post is created

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

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

1. Navigate to a post.
2. Enter a comment including one or more emojis.
3. Submit the comment.
4. Send a private message to another user including one or more emojis.

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

- The comment and message are successfully submitted.

**\*\*Expected Result:\*\* The system allows users to use emojis in comments and messages, displaying them correctly.**

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case verifies that the platform displays the total number of reactions received on a post, providing visibility into user engagement.**

**\*\*Test Suite:\*\* Emojis and Reactions**

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

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

- User is logged in
- A post is created
- At least one reaction has been added to the post.

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

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

1. Navigate to a post.
2. Observe the reaction count displayed near the reaction button.
3. Observe the number of individual emojis displayed.

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

- The reaction count is displayed accurately.

**\*\*Expected Result:\*\* The system displays a count of reactions received on a post, accurately reflecting the total number of reactions and the distribution of individual emojis.**

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case ensures that the platform supports a comprehensive selection of emojis, covering various categories and languages.**

**\*\*Test Suite:\*\* Emojis and Reactions**

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

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

- User is logged in
- A post is created

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

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

1. Navigate to a post.
2. Click on the reaction button.
3. Browse through the available emojis.
4. Observe the categories and languages represented in the emoji selection.

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

- The system displays a wide range of emojis covering different categories and languages.

**\*\*Expected Result:\*\* The platform supports a diverse selection of emojis encompassing various categories, including emotions, objects, animals, flags, and different languages.**

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

**\*\*Type of Testing:\*\* Functionality & Usability**

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

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



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

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

**\*\*Test Case Description:\*\* This test case ensures that users have the ability to change or remove their reactions to posts, providing flexibility and control over their interactions.**

**\*\*Test Suite:\*\* Emojis and Reactions**

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

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

- User is logged in
- A post is created
- The user has reacted to the post with an emoji.

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

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

1. Navigate to a post where the user has already reacted.
2. Click on the user's existing reaction emoji.
3. Select a different emoji or choose to remove the reaction.

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

- The user's reaction is updated or removed accordingly.

**\*\*Expected Result:\*\* The system allows users to change or remove their reactions to posts, updating the display and emoji count accordingly.**

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

**\*\*Type of Testing:\*\* Functionality & Usability**

**\*\*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 -**

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 Context:\*\***

\* Project Name: Social Media Platform

\* Project Description: A comprehensive social media platform with features like profiles, connections, posts,

live streaming, etc.

\* Feature Name: Live Streaming

\* Feature Description: Enables users to broadcast real-time video content to followers or publicly.

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

\* Acceptance Criteria:

1. Users can start and stop live streams.
2. Live streams can be public or limited to followers.
3. Viewers can interact with the streamer through comments and reactions.
4. The stream should display viewer count and live comments.
5. Users can save the live stream for later viewing.

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

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

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

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

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

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

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

\* User is logged in.

\* User has a valid internet connection.

\* User has a profile with followers (if applicable).

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

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

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

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

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case ensures the user can initiate and terminate live streams successfully.

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

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

**\*\*Expected Result:\*\*** The live stream starts broadcasting successfully and stops upon clicking the "End Live Stream" button.

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

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

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

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

**\*\*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 ensures that the user can control the visibility of their live stream, either public or limited to followers.

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

1. Navigate to the live stream creation section.
2. Select the "Public" visibility option for the live stream.
3. Start the live stream.
4. Verify that the live stream is visible to all users.
5. Stop the live stream.
6. Repeat steps 1-3 but select the "Followers Only" visibility option.
7. Verify that the live stream is only visible to the user's followers.

**\*\*Expected Result:\*\*** The live stream is visible to all users when set to "Public" and only visible to the user's followers when set to "Followers Only".

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

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

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

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

**\*\*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 ensures viewers can engage with the live streamer by commenting and reacting to the content.

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

1. Start a live stream.
2. As a viewer, access the live stream.
3. Type a comment and send it.
4. Verify that the comment appears on the live stream.
5. Use different reaction options (e.g., like, heart) and verify they are displayed on the live stream.

**\*\*Expected Result:\*\*** Viewer comments and reactions are displayed on the live stream in real-time.

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case ensures that the live stream displays the number of viewers and comments in real-time.**

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

1. Start a live stream.
2. Verify that the viewer count is displayed on the live stream.
3. Have multiple viewers join the live stream.
4. Verify that the viewer count updates accordingly.
5. Verify that comments are displayed on the live stream in real-time.

**\*\*Expected Result:\*\* The live stream displays the accurate viewer count and live comments, updating dynamically as viewers join and interact.**

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

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

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

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

**\*\*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 ensures that users can save their completed live stream for later**

access.

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

1. Start a live stream.
2. End the live stream.
3. Verify that the option to save the live stream is available.
4. Save the live stream.
5. Verify that the saved live stream is accessible from the user's profile or a dedicated "Saved Live Streams" section.

**\*\*Expected Result:\*\*** The user can save the live stream for later viewing, and the saved stream is accessible through the platform.

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

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

**\*\*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 Social Media Platform - Stories Feature

**Project Name:** Social Media Platform

**Feature Name:** Stories

**User Story Title:** 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:** Stories

**Test Priority:**

**Preconditions:**

**Test Data:**



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

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

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

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case ensures that users can create stories using different types of media and add text overlays to enhance their content.

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

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

- User is logged in.
- User has access to the Stories feature.

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

- Photo: A sample image.
- Video: A short video clip.
- Text Overlay: A sample text message.

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

1. Access the Stories creation feature.
2. Select the "Photo" option and upload the sample image.
3. Add a text overlay to the image.
4. Select the "Video" option and upload the video clip.
5. Add a text overlay to the video.

6. Publish both the photo story and the video story.

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

- Both the photo and video stories are published successfully.

**\*\*Expected Result:\*\*** The system should successfully create and publish both photo and video stories with the added text overlays.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that the system automatically removes stories after 24 hours, ensuring their temporary nature.

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

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

- User is logged in.
- User has created a story.

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

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

1. Create a story with any content.
2. Wait for more than 24 hours.
3. Check the user's profile and the Stories section.

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

- Story is no longer visible on the user's profile or Stories section.

**\*\*Expected Result:\*\*** The system should automatically remove the story after 24 hours.

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case confirms that users can see a list of viewers for their stories, providing insight into engagement.**

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

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

- User is logged in.
- User has created a story.
- Another user has viewed the story.

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

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

1. Create a story with any content.
2. Ensure another user views the story.
3. Access the story and check the "Viewers" list.

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

- The list of viewers for the story is displayed.

**\*\*Expected Result:\*\* The system should show a list of viewers who have seen the user's story.**

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case ensures that users can control the visibility of their stories, allowing them to share with specific audiences.

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

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

- User is logged in.
- User has created a story.

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

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

1. Create a story with any content.
2. Choose the "Public" sharing option for the story.
3. Publish the story.
4. Create another story and select the "Specific Followers" option.
5. Choose a specific set of followers and publish the story.

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

- Both stories are published with their respective sharing options.

**\*\*Expected Result:\*\*** The system should allow the user to publish stories with different sharing options: public or specific followers.

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case ensures that users can enhance their stories with stickers and filters, adding visual appeal and creativity.**

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

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

- User is logged in.
- User has created a story.

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

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

1. Create a story with any content.
2. Access the sticker library and add different stickers to the story.
3. Apply different filters to the story.
4. Publish the story with the added stickers and filters.

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

- The story is published with the added stickers and filters.

**\*\*Expected Result:\*\* The system should allow the user to add stickers and filters to their story before publishing.**

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

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

**\*\*Test Case Approach:\*\* 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 Social Media Platform - Analytics and Insights

**\*\*Project Context:\*\***

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

**\* \*\*Project Description:\*\*** A comprehensive social media platform with features like profiles, connections,

posts, messaging, privacy settings, and analytics.

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

\* \*\*Feature Description:\*\* Users can access data about their activity, audience engagement, and content performance.

\* \*\*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 Creation Template:\*\*

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Type of Testing	Test Case Approach
TC-001	Verify that users can view data on follower demographics	This test case ensures that users can access information about their followers' demographics, such as age, location, and gender.	Analytics and Insights	High	User is logged in and has published at least one post.	No test data needed	1. Navigate to the user's profile page. 2. Access the analytics section. 3. View the "Follower Demographics" tab. 4. Verify that the demographics data is displayed, including age, location, and gender.   User navigates back to the analytics section.   The system displays relevant data about the user's followers' demographics in a clear and accessible format.			Major	Functional	Positive

| TC-002 | Verify that post reach and engagement metrics are available | This test case verifies that users can access information about the reach and engagement of their posts. | Analytics and Insights | High | User is logged in and has published at least one post. | No test data needed | 1. Navigate to the user's profile page. 2. Access the analytics section. 3. View the "Post Performance" tab. 4. Verify that metrics such as post reach, impressions, likes, comments, and shares are displayed. | User navigates back to the analytics section. | The system displays relevant metrics about the user's post performance in a clear and accessible format. | Major | Functional | Positive |

| TC-003 | Verify that users can track audience growth trends | This test case ensures that users can see how their audience has grown over time. | Analytics and Insights | High | User is logged in and has published at least one post. | No test data needed | 1. Navigate to the user's profile page. 2. Access the analytics section. 3. View the "Audience Growth" tab. 4. Verify that the user can see a graph or chart displaying follower growth over a specified time period. | User navigates back to the analytics section. | The system displays a graph or chart representing follower growth trends over a chosen time period. | Major | Functional | Positive |

| TC-004 | Verify that analytics are displayed in a user-friendly format | This test case checks if the analytics interface is intuitive and easy to understand. | Analytics and Insights | High | User is logged in and has published at least one post. | No test data needed | 1. Navigate to the user's profile page. 2. Access the analytics section. 3. Observe the visual layout and organization of data within the analytics section. 4. Ensure that data is presented using clear labels, charts, graphs, and visualizations. | User navigates back to the analytics section. | The analytics section is easy to navigate, with clear labels, visuals, and logical data organization. | Major | Usability | Positive |

| TC-005 | Verify that data can be filtered by time range and content type | This test case ensures that users can filter analytics data by time range and content type to get specific insights. | Analytics and Insights | High | User is logged in and has published at least one post. | No test data needed | 1. Navigate to the user's profile page. 2. Access the analytics section. 3. Access the "Filters" section. 4. Choose a specific time range and content type (e.g., last 7 days, videos). 5. Verify that the displayed data reflects the selected filters. | User navigates back to the analytics section. | The system correctly filters the displayed data based on the chosen time range and content type. | Major | Functional | 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 Events and Groups Feature

\*\*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:\*\* TC\_EG\_01**

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

**\*\*Test Case Description:\*\* This test case ensures that a user can create an event, invite others, and join an existing event.**

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

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

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

- User is logged in.

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

- Event title: "Book Club Meeting"
- Event description: "Discussion about 'The Great Gatsby'"
- Event date: 2024-03-15
- Event time: 7:00 PM
- Event location: "Central Library"

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

1. Navigate to the "Events" section.
2. Click "Create Event".
3. Enter the event details (title, description, date, time, location).
4. Click "Create".
5. Search for an existing event and click "Join".

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

- The event is created and visible on the user's profile.
- The user is added to the list of attendees for the joined event.

**\*\*Expected Result:\*\*** The system successfully creates the event and allows the user to join another event.

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

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

**\*\*Test Case Behaviour:\*\* Positive**

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

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

**\*\*Test Case Description:\*\* This test case ensures that a user can search for and join groups based on their interests.**

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

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

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

- User is logged in.

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

- Group name: "Photography Enthusiasts"

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

1. Navigate to the "Groups" section.
2. Search for the group "Photography Enthusiasts".
3. Click "Join Group".

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

- The user is added to the "Photography Enthusiasts" group.

**\*\*Expected Result:\*\*** The system successfully allows the user to join the group.

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

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

**\*\*Test Case Behaviour:\*\*** Positive

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

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

**\*\*Test Case Description:\*\*** This test case checks if a user can create new discussions within a group and categorize them by specific topics.

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

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

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

- User is logged in and a member of a group.

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

- Group name: "Book Club"

- Discussion topic: "Character Analysis"

- Discussion content: "Thoughts on Jay Gatsby's character development?"

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

1. Navigate to the "Book Club" group.
2. Click "Start a Discussion".
3. Enter the discussion topic and content.
4. Select the category "Character Analysis".
5. Click "Post".

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

- The discussion post is visible in the group feed under the "Character Analysis" category.

**\*\*Expected Result:\*\*** The system successfully creates a new discussion post categorized by the selected topic.

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

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

**\*\*Test Case Behaviour:\*\*** Positive

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

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

**\*\*Test Case Description:\*\*** This test case ensures that all essential event details are displayed prominently for users to see.

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

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

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

- User is logged in.
- An event is created.

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

- Event details:
  - Date: 2024-03-22
  - Time: 6:00 PM
  - Location: "Downtown Park"
  - Description: "Outdoor Movie Night"

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

1. Navigate to the event page.
2. Verify that the event details (date, time, location, and description) are displayed on the event page.

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

- Event details are displayed correctly.

**\*\*Expected Result:\*\*** The system displays all the event details accurately.

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

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

**\*\*Test Case Behaviour:\*\* Positive**

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

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

**\*\*Test Case Description:\*\* This test case ensures that a user can RSVP to an event and view the list of other attendees.**

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

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

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

- User is logged in.
- An event is created.

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

- Event name: "Concert in the Park"

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

1. Navigate to the "Concert in the Park" event page.
2. Click "RSVP".
3. Select "Going".



4. Click "View Attendees".

5. Verify that the user's name is on the list of attendees.

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

- The user is marked as "Going" for the event.

- The user can view the list of attendees.

**\*\*Expected Result:\*\*** The system allows the user to RSVP and view the list of attendees for the event.

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

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

**\*\*Test Case Behaviour:\*\*** 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 Social Media Platform - Monetization Options

**\*\*Project Context:\*\***

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

\* **\*\*Project Description:\*\*** A comprehensive social media platform for users to create profiles, connect with friends, share content, and engage with communities.

\* **\*\*Feature Name:\*\*** Monetization Options

\* **\*\*Feature Description:\*\*** Users can earn money through various means, including advertising revenue, sponsored content, and more.

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

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

1. Users can apply for monetization options.
2. Monetization includes advertising revenue and sponsored content.
3. Users can receive payments through the platform.
4. Detailed earnings reports are available.
5. Monetization eligibility requirements are clearly stated.

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

**\*\*Test Case ID:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module)

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

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

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that the system allows users to apply for monetization options and that the application process is accessible and functional.

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

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

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

\* User is logged in and has a profile on the platform.

\* The user has a sufficient number of followers or content engagement for monetization eligibility (if applicable).

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

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

1. Navigate to the user's profile settings.
2. Locate and click on the "Monetization" or "Earnings" section.
3. Click on the "Apply for Monetization" or "Enable Monetization" button.
4. Review and accept the monetization terms and conditions.
5. Submit the application.

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

\* The system displays a confirmation message or updates the user's profile status indicating the application is submitted.

**\*\*Expected Result:\*\*** The system successfully processes the user's application for monetization options.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that the platform offers monetization options including advertising revenue and sponsored content, and that these features are accessible and functional.

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

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

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

\* User is logged in and has a monetized profile.

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

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

1. Navigate to the user's monetization settings or earnings dashboard.
2. Verify the presence of options related to advertising revenue and sponsored content.
3. Click on the "Advertising" or "Sponsored Content" section.
4. Explore the available settings and options related to each monetization feature.

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

\* The user can successfully access and interact with the settings related to advertising revenue and sponsored content.

**\*\*Expected Result:\*\*** The system allows users to configure settings for monetization through advertising revenue and sponsored content.

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case verifies that users can receive payments earned through monetization options directly through the platform, ensuring a seamless and secure payment process.**

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

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

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

\* User is logged in and has a monetized profile.

\* User has reached the minimum payout threshold for their chosen payment method.

\* User has provided valid payment information (e.g., bank account, PayPal, etc.).

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

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

1. Navigate to the user's earnings dashboard or payment settings.

2. Request a payout or withdraw earned funds.

3. Confirm the payment amount and select the preferred payment method.

4. Complete the payout request.

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

\* The system displays a confirmation message indicating the payout request was submitted.

\* The user receives a notification or email about the payment processing.

**\*\*Expected Result:\*\* The system successfully processes and delivers payments to the user through their chosen method.**

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case verifies that users can access comprehensive and detailed earnings reports, providing insights into their revenue generation and enabling effective financial management.**

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

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

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

\* User is logged in and has a monetized profile.

\* User has earned some revenue through monetization options.

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

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

1. Navigate to the user's earnings dashboard.
2. Access the "Earnings Reports" or "Transaction History" section.
3. Review the detailed reports showing earnings from different sources, dates, and amounts.
4. Check for filters or options to customize the report view.

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

\* The user can successfully download or view the earnings reports in various formats.

**\*\*Expected Result:\*\* The system provides accurate and detailed earnings reports for users to track their revenue and analyze performance.**

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case verifies that the platform clearly communicates the requirements and criteria for monetization eligibility, helping users understand what they need to meet to participate.

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

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

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

\* User is logged in.

\* User is not yet eligible for monetization.

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

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

1. Navigate to the user's monetization settings or earnings dashboard.

2. Locate and review the information about "Monetization Requirements" or "Eligibility Criteria."

3. Verify that the requirements are presented clearly and concisely, including factors like follower count, engagement metrics, content quality, and any other relevant criteria.

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

\* The user can easily understand the monetization eligibility requirements.

**\*\*Expected Result:\*\*** The system clearly communicates the requirements for monetization eligibility, ensuring transparency and user understanding.

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

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

**\*\*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 Social Media Platform - Explore Tab

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

**\*\*Feature Name:\*\* Explore or Discover Tab**

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

**\*\*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:\*\*** | **\*\*Type of Testing:\*\*** | **\*\*Test Case Approach:\*\***

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

**\*\*TC\_EXPLORE\_01\*\*** | Verify that the explore tab shows trending and popular content. | This test case verifies that the explore tab displays trending and popular content from across the platform, ensuring users can see the most engaging content. | Explore Tab | High | User is logged in. | No test data needed. | 1. Navigate to the Explore tab. 2. Observe the displayed content. 3. Verify that the content is relevant to trending topics, popular posts, and accounts. | No postconditions. | The explore tab displays trending and popular content, showcasing the most engaging content across the platform. | Major | Functional Testing | Positive

**\*\*TC\_EXPLORE\_02\*\*** | Verify that content recommendations are personalized based on user interests. | This test case checks if the explore tab offers personalized content recommendations based on user interests and engagement history, enhancing the user experience. | Explore Tab | High | User is logged in. User has previously interacted with specific content types or accounts. | User's interaction history, including liked posts, followed accounts, commented posts, etc. | 1. Navigate to the Explore tab. 2. Observe the displayed recommendations. 3. Verify that the recommendations are relevant to the user's previous interactions, showing content related to their interests. | No postconditions. | The explore tab displays personalized recommendations based on the user's engagement history and interests. | Major | Functional Testing | Positive

**\*\*TC\_EXPLORE\_03\*\*** | Verify that users can filter content by category or topic. | This test case verifies that users can filter the content displayed in the explore tab by category or topic, allowing them to refine their exploration and find relevant content more easily. | Explore Tab | Medium | User is logged in. The explore tab displays diverse content. | Various categories or topics relevant to the platform. | 1. Navigate to the Explore tab. 2. Locate the filter options for categories or topics. 3. Select a specific category or topic filter. 4. Observe the displayed content. | No postconditions. | The explore tab displays content filtered based on the selected category or topic, providing a more focused exploration experience. | Minor | Functional Testing | Positive

**\*\*TC\_EXPLORE\_04\*\*** | Verify that the explore tab updates regularly with new content. | This test case ensures the explore tab is updated regularly with new content, maintaining its relevance and providing users with fresh content to discover. | Explore Tab | Medium | User is logged in. The explore tab displays content. | No test data needed. | 1. Navigate to the Explore tab. 2. Note the timestamp or date of the displayed content. 3. Wait for a predetermined time interval (e.g., 30 minutes). 4. Refresh the Explore tab and observe the content. | No postconditions. | The explore tab displays new content, updated regularly to reflect trending topics and recent activity. | Minor | Performance Testing | Positive

**\*\*TC\_EXPLORE\_05\*\*** | Verify that users can save or follow content from the explore tab. | This test case verifies that users can save or follow content they discover in the explore tab, enabling them to revisit or interact with it later. | Explore Tab | Medium | User is logged in. The explore tab displays content. | User-specific content from the Explore tab. | 1. Navigate to the Explore tab. 2. Select a post, account, or topic. 3. Locate the "Save" or "Follow" options. 4. Click on "Save" or "Follow." 5. Verify that the content is saved or followed in the user's profile or saved items. | No postconditions. | The system allows users to save or follow content discovered in the Explore tab, facilitating engagement and access to the content later. | Minor | Functional Testing | 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 on Social Media Platform

**\*\*Project Context:\*\***

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.

User Story 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**\*\*Test Case ID:\*\*** AC-01

**\*\*Test Case Title:\*\*** Verify that the platform provides alt text for images.

**\*\*Test Case Description:\*\*** This test case verifies if the platform provides alternative text for images, allowing screen reader users to understand the content of the image.

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

- User is logged in.
- An image is uploaded in a post.

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

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

1. Navigate to a post containing an image.
2. Use a screen reader to read the content of the post.
3. Check if the screen reader reads the alt text associated with the image.

**\*\*Postconditions:\*\*** No postconditions.

**\*\*Expected Result:\*\*** The screen reader reads the alt text describing the image content.

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

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

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

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

**\*\*Test Case ID:\*\*** AC-02

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

**\*\*Test Case Description:\*\*** This test case ensures the platform's compatibility with common screen reader software, enabling users with visual impairments to navigate the platform.

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

- User is logged in.
- A screen reader software (e.g., JAWS, NVDA) is installed.

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

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

1. Use the screen reader to navigate through the platform's various sections, including the newsfeed, profile pages, and settings.

2. Verify that the screen reader accurately reads the content and interface elements.

3. Ensure the screen reader can navigate through interactive elements such as buttons, links, and menus.

**\*\*Postconditions:\*\*** No postconditions.

**\*\*Expected Result:\*\*** The screen reader functions correctly, providing audible feedback for all interface elements and content.

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

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

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

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

**\*\*Test Case ID:\*\*** AC-03

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

**\*\*Test Case Description:\*\*** This test case ensures users can navigate the platform using keyboard shortcuts, providing an alternative to mouse interaction for individuals with motor impairments.

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

- User is logged in.
- The keyboard is the primary input method.

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

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

1. Use keyboard shortcuts to navigate through different sections of the platform, including newsfeed, profile page, and settings.
2. Verify that each shortcut performs its intended action, such as opening a menu, selecting an option, or navigating to another page.

**\*\*Postconditions:\*\*** No postconditions.

**\*\*Expected Result:\*\*** The keyboard shortcuts function correctly, enabling navigation and interaction without relying on mouse input.

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

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

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

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

**\*\*Test Case ID:\*\* AC-04**

**\*\*Test Case Title:\*\* Verify that color contrast and font size options are available.**

**\*\*Test Case Description:\*\* This test case ensures the platform offers color contrast and font size options, enabling users with visual impairments to adjust the display to their preferences.**

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

- User is logged in.
- The platform's settings are accessible.

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

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

1. Navigate to the platform's settings.
2. Locate the color contrast and font size options.
3. Adjust the color contrast and font size settings to different levels.
4. Verify the platform's appearance changes accordingly.

**\*\*Postconditions:\*\* No postconditions.**

**\*\*Expected Result:\*\* The platform allows users to adjust color contrast and font size, enhancing readability and accessibility.**

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

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

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

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



**\*\*Test Case ID:\*\* AC-05**

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

**\*\*Test Case Description:\*\* This test case ensures that the platform's accessibility features are adequately documented, providing users with information on how to use them.**

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

- User is logged in.
- The platform's help or support documentation is accessible.

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

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

1. Access the platform's help or support documentation.
2. Search for information related to accessibility features.
3. Verify that the documentation provides clear instructions and explanations on using accessibility features.

**\*\*Postconditions:\*\* No postconditions.**

**\*\*Expected Result:\*\* The documentation includes a section on accessibility features, explaining their purpose, usage, and configuration options.**

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

**\*\*Type of Testing:\*\* Usability, Accessibility**

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

-----

**Data - 17**

**Project name -**

Social Media Platform

**Project description -**

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

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

#### **Feature name -**

Safety and Moderation Tools

#### **Feature description -**

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

#### **Userstory title -**

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

#### **Acceptance criteria -**

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

#### **Generated Test Cases:**

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

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

**\*\*Feature Name:\*\* Safety and Moderation Tools**

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

1. Content moderation algorithms filter inappropriate content.
2. Users can report harmful or inappropriate content.
3. Blocking capabilities are available to users.
4. Comment filters prevent offensive language.
5. Community guidelines are enforced consistently.

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

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

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

**\*\*Test Case Description:\*\*** This test case aims to ensure that the content moderation algorithms effectively filter out inappropriate content, such as hate speech, violence, and explicit materials.

**\*\*Test Suite:\*\*** Content Moderation

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

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

\* User is logged in.

\* A test account is created.

\* A post containing inappropriate content (hate speech, violence, etc.) is prepared.

**\*\*Test Data:\*\*** Inappropriate content for testing.

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

1. Login to the test account.
2. Create a post containing inappropriate content.
3. Submit the post.
4. Observe the outcome of the post.

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

\* The inappropriate content should be flagged and hidden from the public feed.

**\*\*Expected Result:\*\*** The content moderation algorithms should successfully identify and filter the inappropriate content, preventing it from appearing in the public feed.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case checks the functionality of the reporting mechanism. It ensures that users can easily report harmful or inappropriate content encountered on the platform.

**\*\*Test Suite:\*\*** Reporting

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

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

\* User is logged in.

\* A post with harmful or inappropriate content is visible on the platform.

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

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

1. Locate a post with inappropriate content.
2. Navigate to the reporting options.
3. Select the appropriate reason for reporting.
4. Submit the report.

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

\* The user should receive a confirmation message.

\* The platform should take appropriate action on the reported content.

**\*\*Expected Result:\*\*** The system should successfully process the report, sending it to the moderation team for review and action.

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

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

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

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

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

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

**\*\*Test Case Description:\*\*** This test case evaluates the functionality of the blocking feature, ensuring users can block other users and prevent them from interacting with their content.

**\*\*Test Suite:\*\*** Blocking

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

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

\* User A is logged in.

\* User B is logged in.

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

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

1. User A navigates to User B's profile.
2. User A initiates the blocking process.
3. User A confirms the blocking action.

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

\* User B should no longer be able to view User A's content.

\* User B should no longer be able to send messages to User A.

**\*\*Expected Result:\*\*** User B is successfully blocked, and their ability to interact with User A's content is restricted.

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

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that comment filters prevent offensive language.

**\*\*Test Case Description:\*\*** This test case checks the effectiveness of the comment filter in preventing offensive language from being posted on the platform.

**\*\*Test Suite:\*\*** Comment Filtering

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

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

\* User is logged in.

\* A post is created.

**\*\*Test Data:\*\*** Offensive language phrases for testing.

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

1. Go to a post and attempt to submit a comment containing offensive language.
2. Observe the comment submission process.

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

\* The comment should be blocked or flagged.

\* The user should receive an error message or warning.

**\*\*Expected Result:\*\*** The comment filter should successfully identify and block the offensive language, preventing it from being posted.

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

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

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

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

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

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

**\*\*Test Case Description:\*\* This test case aims to evaluate the platform's consistency in enforcing community guidelines, ensuring that all violations are treated equally and appropriately.**

**\*\*Test Suite:\*\* Community Guidelines Enforcement**

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

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

\* User is logged in.

\* A post violating community guidelines is prepared.

**\*\*Test Data:\*\* A post that violates the community guidelines.**

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

1. Create a post that violates the community guidelines.

2. Submit the post.

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

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

\* The violating post should be removed or hidden.

\* The user who posted the content should face appropriate consequences, such as a warning or temporary suspension.

**\*\*Expected Result:\*\* The platform consistently identifies and addresses violations of the community guidelines, ensuring a fair and safe environment for all users.**

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

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

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

-----

## **Data - 18**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Multi-Language Support

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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



Generated Test Cases:

## Test Cases for Multi-Language Support Feature

**Project Name:** Social Media Platform

**Feature Name:** Multi-Language Support

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

**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	Type of Testing	Test Case Approach
TL-001	Verify that users can select their preferred interface language	Ensure that the system allows users to choose their preferred language for the platform's user interface.	Multi-Language Support	High	- User is logged in.	- List of supported languages (e.g., English, Spanish, French, Chinese, etc.)	1. Navigate to the language settings section. 2. Select the desired language from the dropdown list. 3. Save the language preference.	- Language preference is saved in user profile.	The system successfully updates the interface language to the selected language, and all elements (buttons, menus, text) are displayed in the chosen language.	Major	Functional	Positive
TL-002	Verify that the platform offers content translation features	Ensure that the system provides the functionality to translate posts and messages into different languages.	Multi-Language Support	High	- User is logged in. - A post or message is created in one language.	- Post/message content in the source language. - List of supported languages for translation.	1. Navigate to a post or message. 2. Identify the translation option (button, icon). 3. Select the desired target language. 4. Initiate the translation process.	- Content is translated into the target language.	The system successfully translates the post/message content into the chosen target			

language, preserving the original content and meaning. | Major | Functional | Positive |

| TL-003 | Verify that posts and messages can be translated into different languages | Ensure that the system can translate posts and messages into multiple supported languages. | Multi-Language Support | Medium | - User is logged in. - A post or message is created in a source language. | - Post/message content in English. - Supported languages (e.g., Spanish, French, German, Japanese). | 1. Create a post or message in English. 2. Select the translation option. 3. Translate the post/message into Spanish, French, German, and Japanese. | - Content is translated into each target language. | The system successfully translates the post/message content into each of the chosen languages, ensuring accuracy and preserving the original meaning. | Minor | Functional | Positive |

| TL-004 | Verify that language preferences are saved and applied consistently | Ensure that the system remembers the user's preferred language and applies it to the interface and content consistently across different sessions. | Multi-Language Support | High | - User is logged in. - User has selected a preferred language. | - No test data needed. | 1. Log out of the platform. 2. Log back in as the same user. | - Interface and content are displayed in the previously selected language. | The system displays the interface and content in the user's preferred language after login, consistently applying the chosen setting across different sessions. | Major | Functional | Positive |

| TL-005 | Verify that the platform supports a wide range of languages | Ensure that the system offers support for a diverse selection of languages, catering to a global user base. | Multi-Language Support | Medium | - User is logged in. | - List of supported languages. | 1. Navigate to language settings. 2. Review the list of supported languages. | - The list includes a wide variety of languages, covering major language families and regional languages. | The system supports a range of languages, including major world languages and languages spoken by significant populations, ensuring accessibility for diverse user groups. | Minor | Functional | 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 Feature

**\*\*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.**

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

**\*\*Test Case ID:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module)

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

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

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

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

---

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

**\*\*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 a user's account with other social media platforms, ensuring that the integration process is successful and the user can choose which platforms to connect.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

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

- User is logged in to the social media platform.

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

- User account information
- Social media platform accounts (e.g., Facebook, Twitter, Instagram)

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

1. Navigate to the user's profile settings.
2. Select the "Connect Accounts" or "Linked Accounts" option.
3. Choose a social media platform to link.
4. Enter the login credentials for the chosen platform.
5. Authorize the connection request.

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

- User account is linked to the chosen social media platform.
- The user sees a confirmation message and the linked account details.

**\*\*Expected Result:\*\*** The user's account is successfully linked with the selected social media platform, and the connection is confirmed through a notification or visual indication.

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

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

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

---

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

**\*\*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 ability to cross-post content created on the social media platform to other linked social media platforms. It includes creating a post, selecting platforms for cross-posting, and ensuring the content is successfully published on the chosen platforms.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

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

- User account is linked to other social media platforms.
- User has created a post on the social media platform.

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

- User-generated post content
- Linked social media platform accounts

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

1. Create a new post on the social media platform.
2. Select the platforms where the content should be cross-posted.
3. Publish the post.

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

- The post is successfully published on the social media platform.
- The post appears on the linked social media platforms.

**\*\*Expected Result:\*\*** The post is successfully published on both the social media platform and the selected linked platforms, with appropriate formatting and content.

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

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

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

---

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

**\*\*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 single sign-on (SSO) for users who have linked their accounts with other social media platforms. It ensures that users can log in to the platform using their existing credentials from the linked platform.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

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

- User account is linked to other social media platforms.
- User is not logged in to the social media platform.

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

- User account information
- Linked social media platform accounts

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

1. Go to the login page of the social media platform.
2. Select the option to log in using a linked social media platform.
3. Choose the desired platform for SSO.
4. Authorize the login request.

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

- User is logged in to the social media platform.
- The user sees a confirmation message and their profile is accessible.

**\*\*Expected Result:\*\*** The user is successfully logged in to the social media platform using their linked social

media platform account credentials without having to enter their username or password separately.

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

**\*\*Type of Testing:\*\*** Security

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

---

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

**\*\*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 the integration of external media sources, such as YouTube or Vimeo, into the social media platform. It ensures that users can embed content from these sources into their posts and that the content is displayed correctly.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

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

- User is logged in to the social media platform.
- User has an account on an external media source (e.g., YouTube).

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

- External media content URL (e.g., YouTube video URL)

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

1. Create a new post on the social media platform.
2. Paste the URL of the external media content into the post editor.
3. Publish the post.

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

- The external media content is displayed in the post.



- The user can interact with the embedded content (play, pause, etc.).

**\*\*Expected Result:\*\*** The external media content (e.g., a YouTube video) is successfully embedded and displayed within the post, allowing users to interact with the content without leaving the platform.

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

**\*\*Type of Testing:\*\*** Integration

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

---

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

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

**\*\*Test Case Title:\*\*** Verify that content sharing across platforms is seamless.

**\*\*Test Case Description:\*\*** This test case verifies that sharing content from the social media platform to other linked platforms is seamless, without requiring manual intervention or duplicate content creation. It involves sharing a post on the platform and ensuring that it is automatically shared to the linked platforms.

**\*\*Test Suite:\*\*** Cross-Platform Integration

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

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

- User account is linked to other social media platforms.
- User has created a post on the social media platform.

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

- User-generated post content
- Linked social media platform accounts

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

1. Create a new post on the social media platform.
2. While creating the post, select the option to share it to linked platforms.

3. Publish the post.

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

- The post is successfully published on the social media platform.
- The post is automatically shared on the linked social media platforms.

**\*\*Expected Result:\*\*** The post is shared seamlessly to the linked social media platforms without requiring manual intervention or duplication of content. The user can view the shared content on the linked platforms with minimal to no additional effort.

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

**\*\*Type of Testing:\*\*** Integration

**\*\*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**

**\*\*User Story Title:\*\* As a user, I want gamification and rewards so that I can earn recognition and incentives for my activities.**

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

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

**\*\*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 | Type of Testing | Test Case Approach |

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

| TC-GR-01 | Verify that users can earn points and badges for their activities | This test case validates that the system correctly awards points and badges to users for performing specific actions on the platform. |

Gamification and Rewards | High | - User is logged in - User has performed eligible activities | User profile, activity log | 1. Perform activities like posting, commenting, sharing, liking, etc. 2. Check the user's points and badge count. 3. Verify that points and badges were awarded for the performed activities. | - Points and badges are updated in the user's profile. - Activity log reflects the earned points and badges. | The system should accurately track and display points and badges earned by the user for their activities. | Major | Functional | Positive |

| TC-GR-02 | Verify that leaderboards display top users based on points or achievements | This test case ensures that the platform accurately displays the top users based on points or achievements, allowing users to track their standing compared to others. | Gamification and Rewards | High | - User is logged in | No test data needed | 1. Navigate to the leaderboard section. 2. Observe the displayed rankings. 3. Verify that the top users are correctly ranked based on their points or achievements. | - Leaderboard displays the correct users based on points or achievements. - Leaderboard is sorted in descending order based on points or achievements. | Major | Functional | Positive |

| TC-GR-03 | Verify that challenges are available for users to complete and earn rewards | This test case verifies that the system offers challenges to users, allowing them to earn rewards by completing specific tasks within a given timeframe. | Gamification and Rewards | Medium | - User is logged in | Challenge details (description, requirements, rewards) | 1. Navigate to the challenges section. 2. Select a challenge and review its description, requirements, and rewards. 3. Complete the challenge by performing the required actions. | - User receives a notification upon challenge completion. - Challenge progress is updated in the user's profile. | The system should offer challenges with clear descriptions, requirements, and rewards that can be earned upon completion. | Major | Functional | Positive |

| TC-GR-04 | Verify that rewards can include virtual goods, discounts, or exclusive content | This test case verifies that the system provides a variety of reward options to users, including virtual goods, discounts, and exclusive content, providing an incentive to engage with the platform. | Gamification and Rewards | Medium | - User is logged in - User has earned a reward | Reward details (type, value, conditions) | 1. Review earned rewards in the user's profile. 2. Verify that the rewards include different types of items, such as virtual goods, discounts, or exclusive content. | - The system should offer diverse reward options for users based on their achievements. - Rewards should be clearly presented and understandable. | Minor | Functional | Positive |

| TC-GR-05 | Verify that users can view their progress and achievements in a dedicated section | This test case validates that the system provides a dedicated section for users to track their progress and achievements, offering a visual representation of their engagement and contributions. | Gamification and Rewards | High | - User is logged in | No test data needed | 1. Navigate to the user's profile section. 2. Look for a dedicated section displaying progress and achievements. 3. Verify that the section presents the user's points, badges, completed challenges, and earned rewards. | - The user's progress and achievements are displayed in a clear and organized manner. - The section allows users to track their progress and engage with the gamification features. | Major | Usability | Positive |

-----

## Data - 21

### Project name -

Task Management Tool

### Project description -

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

### Feature name -

Task Creation

## **Feature description -**

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

## **Userstory title -**

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

## **Acceptance criteria -**

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

## **Generated Test Cases:**

## Test Cases for Task Creation Feature

**Project Name:** Task Management Tool

**Feature Name:** Task Creation

**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:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module)

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

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

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

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

### **\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_001

**\*\*Test Case Title:\*\*** Verify that users can enter a task title, description, due date, and priority level.

**\*\*Test Case Description:\*\*** This test case verifies that the system allows users to input all required information for creating a task.

**\*\*Test Suite:\*\* Task Creation**

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

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

- User is logged in.

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

- Task Title: "Complete Project Proposal"
- Task Description: "Finalize the project proposal document and send to client."
- Due Date: 2024-03-15
- Priority Level: High

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

1. Navigate to the "Create Task" section.
2. Enter the specified task title, description, due date, and priority level.
3. Click on the "Create" button.

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

- The task is created and displayed on the task list.

**\*\*Expected Result:\*\*** The system successfully creates a task with the provided title, description, due date, and priority level.

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

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

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

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

**\*\*Test Case ID:\*\* TC\_002**

**\*\*Test Case Title:\*\* Verify that tasks can be assigned to specific team members.**

**\*\*Test Case Description:\*\* This test case ensures that users can assign tasks to other users within the system.**

**\*\*Test Suite:\*\* Task Creation**



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

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

- User is logged in.
- A task has been created.
- Team members are registered in the system.

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

- Task ID: "Task\_1"
- Team Member: "John Doe"

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

1. Navigate to the created task.
2. Select the "Assign" option.
3. Choose "John Doe" from the list of team members.
4. Save the changes.

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

- The task is assigned to "John Doe".

**\*\*Expected Result:\*\*** The task is successfully assigned to the selected team member.

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

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

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

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

**\*\*Test Case ID:\*\* TC\_003**

**\*\*Test Case Title:\*\* Verify that the system saves and displays created tasks.**

**\*\*Test Case Description:\*\* This test case ensures that the system persists and shows all created tasks.**

**\*\*Test Suite:\*\* Task Creation**

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

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

- User is logged in.
- A task has been created.

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

- Task ID: "Task\_1"

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

1. Log out and log back in.
2. Navigate to the "Tasks" section.

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

- The "Task\_1" should be visible in the task list.

**\*\*Expected Result:\*\*** The system successfully saves and displays the created task.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_004

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

**\*\*Test Case Description:\*\*** This test case verifies that users can modify task details after creation.

**\*\*Test Suite:\*\*** Task Creation

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

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

- User is logged in.
- A task has been created.

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

- Task ID: "Task\_1"

- New Due Date: 2024-03-20

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

1. Navigate to the created task.
2. Click on the "Edit" option.
3. Update the due date to "2024-03-20".
4. Save the changes.

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

- The task details are updated with the new due date.

**\*\*Expected Result:\*\*** The system allows the user to edit the task details and save the changes.

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

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

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

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

**\*\*Test Case ID:\*\*** TC\_005

**\*\*Test Case Title:\*\*** Verify that the system validates required fields (e.g., task title).

**\*\*Test Case Description:\*\*** This test case checks whether the system prevents task creation without a title.

**\*\*Test Suite:\*\*** Task Creation

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

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

- User is logged in.

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

- Task Title: (Empty)
- Task Description: "Finalize the project proposal document and send to client."
- Due Date: 2024-03-15
- Priority Level: High

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

1. Navigate to the "Create Task" section.
2. Enter the specified description, due date, and priority level, but leave the title empty.
3. Click on the "Create" button.

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

- The system displays an error message indicating the missing title.

**\*\*Expected Result:\*\*** The system prevents task creation without a title and displays an appropriate error message.

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

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

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

-----

**Data - 22**

**Project name -**

Task Management Tool

**Project description -**

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

**Feature name -**

Task Assignment

**Feature description -**

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

## **Userstory title -**

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

## **Acceptance criteria -**

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

## **Generated Test Cases:**

### **## Task Management Tool - Task Assignment Test Cases**

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

**\*\*Test Case ID:\*\*** TC-TM-TA-01

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

**\*\*Test Case Description:\*\*** This test case ensures that users can select team members from a list to assign tasks to them within the Task Management Tool.

**\*\*Test Suite:\*\*** Task Assignment

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

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

- User is logged in.
- A task is created.

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

- List of team members (at least 3 members).

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

1. Go to the task details page.
2. Click on the "Assign" button.

3. Select a team member from the dropdown list.

4. Click on the "Assign" button.

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

- The selected team member is assigned to the task.
- The assignee's name is displayed next to the task.

**\*\*Expected Result:\*\*** The task is assigned to the selected team member successfully.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-TM-TA-02

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

**\*\*Test Case Description:\*\*** This test case verifies that when a user assigns a task to a team member, the assignee receives a notification alerting them about the assigned task.

**\*\*Test Suite:\*\*** Task Assignment

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

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

- User is logged in.
- A task is created.
- A team member is assigned to the task.

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

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

1. Go to the task details page.
2. Assign the task to a team member.
3. Login as the assigned team member.

4. Check the notification center for a notification about the assigned task.

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

- The assigned team member receives a notification about the task.

**\*\*Expected Result:\*\*** The assigned team member receives a notification about the newly assigned task.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-TM-TA-03

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

**\*\*Test Case Description:\*\*** This test case checks if the assigned tasks are visible in the assignee's task list within the Task Management Tool.

**\*\*Test Suite:\*\*** Task Assignment

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

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

- User is logged in.
- A task is created.
- A team member is assigned to the task.

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

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

1. Login as the assigned team member.
2. Navigate to the "My Tasks" section.
3. Verify that the assigned task is displayed in the task list.

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

- The assigned task is displayed in the assignee's task list.

**\*\*Expected Result:\*\*** The assigned task appears in the assignee's task list.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-TM-TA-04

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

**\*\*Test Case Description:\*\*** This test case ensures that users can reassign tasks to other team members within the Task Management Tool, updating the task assignment and notification accordingly.

**\*\*Test Suite:\*\*** Task Assignment

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

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

- User is logged in.
- A task is created.
- A team member is assigned to the task.

**\*\*Test Data:\*\*** List of team members.

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

1. Go to the task details page.
2. Click on the "Reassign" button.
3. Select a different team member from the dropdown list.
4. Click on the "Reassign" button.

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

- The task is reassigned to the new team member.
- The previous assignee receives a notification about the reassignment.
- The new assignee receives a notification about the task.



**\*\*Expected Result:\*\*** The task is successfully reassigned to the new team member, and notifications are sent to both the previous and new assignees.

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

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

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

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

**\*\*Test Case ID:\*\*** TC-TM-TA-05

**\*\*Test Case Title:\*\*** Verify that task assignment history is tracked

**\*\*Test Case Description:\*\*** This test case ensures that the system keeps a record of all past task assignments, including the date and time of the assignment and the previous assignee.

**\*\*Test Suite:\*\*** Task Assignment

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

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

- User is logged in.
- A task is created.
- The task has been assigned and reassigned at least once.

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

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

1. Go to the task details page.
2. Click on the "History" tab or similar option to view the task assignment history.
3. Verify that the assignment history includes all past assignments, including dates, times, and previous assignees.

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

- Task assignment history is displayed.

**\*\*Expected Result:\*\*** The system displays a complete history of task assignments, including date and time of

each assignment and the name of the assignee.

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

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

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

-----

## **Data - 23**

### **Project name -**

Task Management Tool

### **Project description -**

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

### **Feature name -**

Task Deadline Reminders

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

## ## Test Cases for Task Deadline Reminders

### \*\*Project Context:\*\*

Project Name: Task Management Tool

Project Description: A task management tool that allows users to create, assign, prioritize, and track tasks and projects.

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.

User Story Title: 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 1\*\*

Test Case ID: TDR-001

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

Test Case Description: Ensure that users can set reminders for tasks based on deadlines and that the system

correctly records and displays the chosen reminder date and time.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged in
- Task is created with a deadline

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page.
2. Locate the 'Reminder' option.
3. Select a reminder date and time for the task.
4. Save the changes.

Postconditions:

- The chosen reminder date and time are saved for the task.
- The system displays the reminder date and time on the task details page.

Expected Result: The system successfully saves the reminder date and time for the task, ensuring that users can set reminders based on deadlines.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

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

Test Case ID: TDR-002

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

Test Case Description: Ensure that users can customize the urgency level of reminders (e.g., low, medium, high) and that the system displays the chosen urgency level correctly.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created with a deadline
- Reminder is set for the task

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page.
2. Locate the 'Reminder Urgency' option.
3. Choose a different urgency level for the reminder.
4. Save the changes.

Postconditions:

- The chosen urgency level is saved for the reminder.
- The system displays the updated urgency level on the task details page.

Expected Result: The system successfully saves the chosen urgency level for the reminder, enabling users to customize reminder urgency.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

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

Test Case ID: TDR-003

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

Test Case Description: Ensure that the system sends timely notifications for upcoming task deadlines, using the chosen notification methods (e.g., email, in-app).

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged in
- Task is created with a deadline
- Reminder is set for the task

Test Data: No test data needed

Test Steps:

1. Set the reminder time for the task to a future time.
2. Wait until the reminder time is reached.
3. Monitor the system for notifications (e.g., email inbox, in-app notifications).

Postconditions:

- The system sends notifications for the approaching deadline.

Expected Result: The system successfully sends notifications to the user regarding the approaching deadline, using the chosen notification methods.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

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

Test Case ID: TDR-004

Test Case Title: Verify that users can choose the delivery method for reminders.

Test Case Description: Ensure that users can select their preferred delivery method for task deadline reminders (e.g., email, in-app) and that the system sends notifications accordingly.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged in
- Task is created with a deadline
- Reminder is set for the task

Test Data: No test data needed

Test Steps:

1. Navigate to the user settings or preferences page.
2. Locate the 'Reminder Delivery Method' option.
3. Choose a different delivery method (e.g., email instead of in-app).
4. Save the changes.
5. Wait for the reminder time to be reached.
6. Monitor the system for notifications using the chosen delivery method.

Postconditions:

- The system saves the chosen delivery method.

Expected Result: The system successfully sends notifications for approaching deadlines using the delivery method selected by the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

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

Test Case ID: TDR-005

Test Case Title: Verify that reminder settings can be adjusted or disabled.

Test Case Description: Ensure that users can adjust or disable reminder settings for individual tasks or globally, and that the system updates the settings accordingly.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created with a deadline
- Reminder is set for the task

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page.
2. Adjust or disable the reminder settings for the task (e.g., change the reminder time, urgency level, or delivery method).
3. Save the changes.
4. Alternatively, navigate to the user settings or preferences page.
5. Adjust or disable global reminder settings (e.g., disable reminders entirely).
6. Save the changes.

Postconditions:

- The system updates the reminder settings accordingly.
- The user can confirm the updated settings.

Expected Result: The system successfully adjusts or disables reminder settings based on user preferences, either for individual tasks or globally, ensuring that reminders are managed effectively.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**Data - 24**

**Project name -**



## Task Management Tool

### **Project description -**

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

### **Feature name -**

Task Prioritization

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

## Test Cases for Task Management Tool - Task Prioritization

**\*\*Project Context:\*\***

\* Project Name: Task Management Tool

\* Project Description: Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects.

\* 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.

\* User Story Title: 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 | Type of Testing | Test Case Approach |

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

| TC-Prioritization-01 | Verify that users can set priority levels for tasks | This test case checks if the user interface allows selecting different priority levels (High, Medium, Low) for newly created tasks. | Task Prioritization | High | User is logged in and a task is created | N/A | 1. Navigate to the task creation screen. 2. Observe the priority level selection options (High, Medium, Low). 3. Select "High" as the priority level. 4. Save the task. | The task is saved with the selected "High" priority level. | The system allows users to select a priority level (High, Medium, Low) for a task. | Minor | Functional | Positive |

| TC-Prioritization-02 | Verify that tasks can be sorted based on priority levels | This test case verifies the functionality of sorting tasks based on their priority level. | Task Prioritization | High | User is logged in and multiple tasks with different priority levels exist | Multiple tasks with different priority levels (High, Medium,

Low) | 1. Access the task list view. 2. Observe the available sorting options. 3. Select "Priority" as the sorting criteria. 4. Verify that the tasks are displayed in descending order of priority (High, Medium, Low). | The task list is sorted based on priority levels, with High priority tasks appearing at the top. | Minor | Functional | Positive |

| TC-Prioritization-03 | Verify that users can change the priority level of a task | This test case ensures that users can modify the priority level of an existing task. | Task Prioritization | High | User is logged in and a task with a pre-defined priority level exists | A task with a pre-defined "Medium" priority level | 1. Select an existing task with a "Medium" priority level. 2. Access the task details. 3. Modify the priority level to "High". 4. Save the changes. | The task priority level is updated to "High". | Minor | Functional | Positive |

| TC-Prioritization-04 | Verify that the system visually distinguishes tasks based on priority | This test case assesses if the system visually differentiates tasks according to their priority level, making it easier for users to identify them. | Task Prioritization | High | User is logged in and multiple tasks with different priority levels exist | Multiple tasks with different priority levels (High, Medium, Low) | 1. Access the task list view. 2. Observe the visual representation of tasks with different priority levels. 3. Verify that tasks with "High" priority are visually distinct from those with "Medium" and "Low" priorities. | Tasks with different priority levels are visually distinguishable (e.g., through color coding, icons, or bold text). | Minor | Usability | Positive |

| TC-Prioritization-05 | Verify that tasks without a specified priority default to Medium | This test case validates that the system assigns a default "Medium" priority to newly created tasks that don't have a priority level specified by the user. | Task Prioritization | Medium | User is logged in and a task is created without specifying a priority level | N/A | 1. Navigate to the task creation screen. 2. Create a new task without selecting any priority level. 3. Save the task. | The task is saved with a default "Medium" priority level. | Minor | Functional | 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 -**

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**

**\*\*Feature Name:\*\* Subtasks**

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

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

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

**\*\*Test Case Description:\*\* This test case verifies the ability to create a subtask under an existing main task.**

**\*\*Test Suite:\*\* Subtask Creation**

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

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

- User is logged in
- A main task exists

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

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

1. Navigate to the main task.
2. Click on the "Add Subtask" button or icon.
3. Enter a name for the subtask.
4. Click on the "Create Subtask" button.

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

- A new subtask is created under the main task.

**\*\*Expected Result:\*\* The system successfully creates the subtask and displays it under the main task.**

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

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

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

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

**\*\*Test Case ID:\*\* TC\_SUBTASK\_DUE\_DATE\_01**

**\*\*Test Case Title:\*\* Verify that subtasks can have their own due dates.**

**\*\*Test Case Description:\*\* This test case validates the ability to set a due date for a subtask independent of the main task's due date.**

**\*\*Test Suite:\*\* Subtask Due Dates**

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

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

- User is logged in
- A subtask is created

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the subtask.
2. Click on the due date field.
3. Select a different due date than the main task.
4. Save the changes.

**\*\*Postconditions:\*\***

- The subtask's due date is updated.

**\*\*Expected Result:\*\* The subtask displays the newly assigned due date, different from the main task's due date.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC\_SUBTASK\_ASSIGNEE\_01**

**\*\*Test Case Title:\*\* Verify that subtasks can have their own assignees.**

**\*\*Test Case Description:\*\* This test case ensures that a subtask can be assigned to a different user than the main task.**

**\*\*Test Suite:\*\* Subtask Assignees**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- A subtask is created
- Multiple users are registered

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the subtask.
2. Click on the assignee field.
3. Select a different user from the main task's assignee.
4. Save the changes.

**\*\*Postconditions:\*\***

- The subtask's assignee is updated.

**\*\*Expected Result:\*\*** The subtask displays the new assignee, different from the main task's assignee.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_SUBTASK\_COMPLETION\_01

**\*\*Test Case Title:\*\*** Verify that subtasks can be marked as complete independently of the main task.

**\*\*Test Case Description:\*\*** This test case checks that a subtask can be marked as completed without impacting the main task's completion status.

**\*\*Test Suite:\*\*** Subtask Completion

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in

- A subtask is created

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the subtask.
2. Click on the checkbox to mark the subtask as completed.
3. Verify the main task's completion status remains unchanged.

**\*\*Postconditions:\*\***

- The subtask is marked as complete.

**\*\*Expected Result:\*\*** The subtask is marked as completed, while the main task's completion status remains unchanged.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_SUBTASK\_DEPENDENCY\_01

**\*\*Test Case Title:\*\*** Verify that dependencies between subtasks can be set.

**\*\*Test Case Description:\*\*** This test case validates the ability to establish dependencies between subtasks, where one subtask's completion is required before another can start.

**\*\*Test Suite:\*\*** Subtask Dependencies

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- Two subtasks are created under the same main task

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***



1. Navigate to subtask 1.
2. Click on the dependency setting option.
3. Select subtask 2 as a dependent subtask.
4. Save the changes.

**\*\*Postconditions:\*\***

- The dependency between the subtasks is established.

**\*\*Expected Result:\*\*** The system creates a dependency where subtask 2 cannot be started until subtask 1 is completed.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*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 Management Tool - Task Dependencies

**Project Name:** Task Management Tool

**Feature Name:** Task Dependencies

**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	Type of Testing	Test Case Approach
TC_TD_01	Verify that users can set dependencies between tasks	Ensure the system allows users to define dependencies between tasks, specifying the order of completion.	Task Dependencies	High	- User is logged in. - Two or more tasks are created.	Task names, task descriptions.	1. Navigate to a task. 2. Select "Add Dependency" option. 3. Choose the dependent task from the list. 4. Save the dependency.	- Dependency is successfully saved. - Dependent task appears under the parent task in the task list.	The system successfully creates a dependency between the two tasks, displaying the dependency in the task details.	Major	Functional	Positive

| TC\_TD\_02 | Verify that the system displays dependencies in the task details | Ensure the dependency is visually represented in the task details, clearly indicating the relationship between tasks. | Task Dependencies | High | - User is logged in. - A dependency is set between two tasks. | No test data needed. | 1. Navigate to the details of the dependent task. 2. Observe the task details. | - Task details display the dependency information. | The task details display the dependent task and the task it depends on, highlighting the relationship. | Major | Functional | Positive |

| TC\_TD\_03 | Verify that tasks cannot be marked as complete if dependent tasks are not completed | Ensure the system prevents users from marking tasks as complete if the dependent tasks are not yet completed, enforcing the dependency order. | Task Dependencies | High | - User is logged in. - A dependency is set between two tasks. - The dependent task is not completed. | Task status (completed/incomplete), dependency information. | 1. Navigate to the parent task. 2. Attempt to mark the task as complete. | - The system prevents the parent task from being marked as complete. - An error message or warning is displayed. | The system blocks the completion of the parent task, displaying an error message or warning until the dependent task is completed. | Blocker | Functional | Negative |

| TC\_TD\_04 | Verify that users can update or remove dependencies as needed | Ensure the system allows users to modify or remove dependencies after they are initially set, providing flexibility in managing task relationships. | Task Dependencies | Medium | - User is logged in. - A dependency is set between two tasks. | Task names, dependency information. | 1. Navigate to the parent task. 2. Select the "Edit Dependency" option. 3. Choose a different dependent task or remove the dependency. 4. Save the changes. | - Dependency is updated or removed successfully. | The system allows users to modify or remove dependencies, reflecting the changes in the task details. | Major | Functional | Positive |

| TC\_TD\_05 | Verify that the system provides alerts or warnings if a dependent task is not completed | Ensure the system notifies users if a dependent task is overdue or not completed, providing timely reminders to manage dependencies effectively. | Task Dependencies | Medium | - User is logged in. - A dependent task is overdue. | Task due date, notification settings. | 1. Navigate to the task list. 2. Check for notifications or warnings related to overdue dependent tasks. | - The system displays alerts or warnings about overdue dependent tasks. | The system effectively notifies users about overdue or incomplete dependent tasks,

prompting action and maintaining project workflow. | Major | Functional | Positive |

**Data - 27**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Comments and Collaboration

**Feature description -**

Users can collaborate on tasks by leaving comments, attaching files, and engaging in discussions directly within the task interface. This feature facilitates communication and teamwork, allowing team members to provide updates, share feedback, and ask questions related to the task.

**Userstory title -**

As a user, I want to comment on tasks so that I can collaborate with my team.

**Acceptance criteria -**

['Users can add comments to tasks.', 'Comments can include text, attachments, and links.', 'Users can reply to and edit their comments.', 'The system notifies users of new comments.', 'Comments are displayed in chronological order.']

**Generated Test Cases:**

## Test Cases for Task Management Tool - Task Comments and Collaboration

**\*\*Project Name:\*\* Task Management Tool**

**\*\*Feature Name:\*\* Task Comments and Collaboration**

**\*\*User Story Title:\*\* As a user, I want to comment on tasks so that I can collaborate with my team.**

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* (Provide a unique identifier for each test case)**

**\*\*Test Case Title:\*\* (Must start with "Verify that..." and clearly state the feature/function being tested)**

**\*\*Test Case Description:\*\* (A brief description of the test case, between 70-100 words)**

**\*\*Test Suite:\*\* Task Comments and Collaboration**

**\*\*Test Priority:\*\* (Priority level: High, Medium, Low)**

**\*\*Preconditions:\*\* (List any prerequisites before executing the test case, max 3 items, in unordered list)**

**\*\*Test Data:\*\* (Specify data required for execution; if none, state "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, in unordered list)**

**\*\*Expected Result:\*\* (Overall expected outcome of the test case)**

**\*\*Severity:\*\* (Blocker, Critical, Major, Minor, Trivial)**

**\*\*Type of Testing:\*\* (Functional Testing)**

**\*\*Test Case Approach:\*\* (Positive, Negative, Destructive)**

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* TC\_001**

**\*\*Test Case Title:\*\* Verify that users can add comments to tasks.**

**\*\*Test Case Description:\*\* This test case ensures that the system allows users to add comments to a task, including text content.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- Task is created

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Locate the comment section.
3. Enter a text comment.
4. Click the "Add Comment" button.

**\*\*Postconditions:\*\***

- The comment is displayed below the task.

**\*\*Expected Result:\*\*** The system successfully adds the comment to the task, and it is displayed below the task with the user's name and timestamp.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_002

**\*\*Test Case Title:\*\*** Verify that comments can include text, attachments, and links.

**\*\*Test Case Description:\*\*** This test case verifies that users can attach files and insert links within comments.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- Task is created

**\*\*Test Data:\*\***

- Sample image file (.jpg)

- Sample website link

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Locate the comment section.
3. Enter a text comment.
4. Attach the sample image file.
5. Paste the website link in the comment.
6. Click the "Add Comment" button.

**\*\*Postconditions:\*\***

- The comment is displayed below the task.

**\*\*Expected Result:\*\*** The system displays the comment, including the attached image file and the clickable website link.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_003

**\*\*Test Case Title:\*\*** Verify that users can reply to and edit their comments.

**\*\*Test Case Description:\*\*** This test case verifies that users can reply to existing comments and edit their own comments.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in
- Task is created
- Existing comment exists on the task.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Locate the existing comment.
3. Click the "Reply" button.
4. Enter a reply comment.
5. Click the "Add Comment" button.
6. Click the "Edit" button on the user's own comment.
7. Edit the comment text.
8. Click the "Save" button.

**\*\*Postconditions:\*\***

- The reply comment is displayed below the original comment.
- The edited comment is updated.

**\*\*Expected Result:\*\*** The system allows the user to successfully reply to an existing comment, and the reply is displayed under the original comment. The user can also edit their own comment, and the changes are saved and reflected in the task.

**\*\*Severity:\*\*** Minor

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_004

**\*\*Test Case Title:\*\*** Verify that the system notifies users of new comments.

**\*\*Test Case Description:\*\*** This test case verifies that the system sends notifications to users when a new comment is added to a task they are assigned to.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in



- Task is created
- User is assigned to the task

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Another user (not the primary user) adds a new comment to the task.
2. Verify the primary user's notification settings.
3. Observe the notifications received by the primary user.

**\*\*Postconditions:\*\***

- The primary user receives a notification about the new comment.

**\*\*Expected Result:\*\*** The system sends a notification to the primary user about the new comment. The notification includes the task name, commenter's name, and a preview of the comment.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_005

**\*\*Test Case Title:\*\*** Verify that comments are displayed in chronological order.

**\*\*Test Case Description:\*\*** This test case ensures that comments are displayed in the order they were added, from oldest to newest.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in
- Task is created
- Multiple comments exist on the task, added at different times.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Observe the order of the displayed comments.

**\*\*Postconditions:\*\***

- Comments are displayed on the task.

**\*\*Expected Result:\*\*** The comments are displayed in chronological order, starting with the oldest comment at the top and the newest comment at the bottom.

**\*\*Severity:\*\*** Minor

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 28**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

File Attachments

### **Feature description -**

Users can attach relevant files, documents, and resources to tasks, ensuring that all necessary information is accessible in one place. File attachments can include documents, spreadsheets, images, links, and other resources needed for task execution.

### **Userstory title -**

As a user, I want to attach files to tasks so that all relevant information is easily accessible.

### **Acceptance criteria -**

['Users can attach multiple files to a task.', 'Attachments can include various file types (documents, images, etc.).', 'Users can view and download attachments.', 'The system limits the size of attachments to prevent data overload.', 'Users can remove attachments if needed.']

## Generated Test Cases:

### ## Test Cases for File Attachments Feature

#### \*\*Project Context:\*\*

\* \*\*Project Name:\*\* Task Management Tool

\* \*\*Project Description:\*\* A task management tool for creating, assigning, prioritizing, and tracking tasks and projects.

\* \*\*Feature Name:\*\* File Attachments

\* \*\*Feature Description:\*\* Users can attach relevant files, documents, and resources to tasks.

\* \*\*User Story Title:\*\* 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 Creation Template:\*\*

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** File Attachments

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* User is logged in

- \* A task is created

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC-001

**\*\*Test Case Title:\*\*** Verify that users can attach multiple files to a task.

**\*\*Test Case Description:\*\*** This test case checks if the system allows users to attach more than one file to a task.

**\*\*Test Data:\*\***

- \* Two different files (e.g., a document and an image).

**\*\*Test Steps:\*\***

1. Navigate to the task.

2. Click on the "Attach File" button.

3. Select the first file from the local directory.
4. Click on the "Attach File" button again.
5. Select the second file from the local directory.
6. Verify that both files are successfully attached to the task.

**\*\*Postconditions:\*\***

- \* Both files are uploaded and visible on the task page.

**\*\*Expected Result:\*\*** The system successfully attaches both files to the task.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC-002

**\*\*Test Case Title:\*\*** Verify that attachments can include various file types.

**\*\*Test Case Description:\*\*** This test case verifies if the system accepts different file types as attachments.

**\*\*Test Data:\*\***

- \* A document file (e.g., .docx)
- \* An image file (e.g., .jpg)
- \* A spreadsheet file (e.g., .xlsx)
- \* A video file (e.g., .mp4)

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Click on the "Attach File" button.
3. Select the document file from the local directory.
4. Attach the image file.
5. Attach the spreadsheet file.

6. Attach the video file.

7. Verify that all files are successfully attached to the task.

**\*\*Postconditions:\*\***

- \* All files are uploaded and visible on the task page.

**\*\*Expected Result:\*\*** The system accepts and attaches all files, regardless of their file types.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC-003

**\*\*Test Case Title:\*\*** Verify that users can view and download attachments.

**\*\*Test Case Description:\*\*** This test case checks if users can view and download attached files.

**\*\*Test Data:\*\***

- \* A document file (e.g., .pdf)

**\*\*Test Steps:\*\***

1. Navigate to the task with an attached file.
2. Click on the attached file name to view it.
3. Verify that the file is displayed correctly (e.g., opens in a new tab or window).
4. Right-click on the file name and select "Download".
5. Verify that the file is downloaded to the local directory.

**\*\*Postconditions:\*\***

- \* The attached file is downloaded to the local directory.

**\*\*Expected Result:\*\*** Users can view and download the attached file.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC-004**

**\*\*Test Case Title:\*\* Verify that the system limits the size of attachments.**

**\*\*Test Case Description:\*\* This test case checks if the system has a file size limit for attachments to prevent data overload.**

**\*\*Test Data:\*\***

\* A large file exceeding the system's limit.

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Click on the "Attach File" button.
3. Select the large file that exceeds the system's file size limit.
4. Attempt to upload the file.

**\*\*Postconditions:\*\***

\* The file is not uploaded.

**\*\*Expected Result:\*\* The system prevents the upload of files that exceed the defined size limit and displays an error message.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Negative**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC-005**

**\*\*Test Case Title:\*\* Verify that users can remove attachments if needed.**

**\*\*Test Case Description:\*\*** This test case verifies if users can remove attached files from a task.

**\*\*Test Data:\*\***

\* A file attached to a task.

**\*\*Test Steps:\*\***

1. Navigate to the task with an attached file.
2. Hover over the attached file.
3. Click on the "Remove" or "Delete" icon (or button) associated with the file.
4. Confirm the removal action.
5. Verify that the file is no longer attached to the task.

**\*\*Postconditions:\*\***

\* The file is removed from the task.

**\*\*Expected Result:\*\*** Users can remove previously attached files from the task.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 29**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Recurring Tasks



## **Feature description -**

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

## **Userstory title -**

As a user, I want to set recurring tasks so that regular activities are automated.

## **Acceptance criteria -**

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

## **Generated Test Cases:**

## Test Cases for Recurring Tasks Feature

**\*\*Project Context:\*\***

\* **\*\*Project Name:\*\*** Task Management Tool

\* **\*\*Project Description:\*\*** A task management tool that allows users to create, assign, prioritize, and track tasks and projects.

\* **\*\*Feature Name:\*\*** Recurring Tasks

\* **\*\*Feature Description:\*\*** Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks.

\* **\*\*User Story Title:\*\*** As a user, I want to set recurring tasks so that regular activities are automated.

**\*\*Acceptance Criteria:\*\***

1. Users can set tasks to recur at specified intervals (daily, weekly, monthly).
2. The system generates instances of recurring tasks automatically.
3. Users can modify the recurrence pattern or cancel the recurrence.
4. The system notifies users of upcoming instances of recurring tasks.
5. Users can view a list of all instances of a recurring task.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case 1\*\***

**\* \*\*Test Case ID:\*\* TC\_Recurring\_01**

**\* \*\*Test Case Title:\*\* Verify that users can set tasks to recur at specified intervals.**

**\* \*\*Test Case Description:\*\* This test case verifies that users can select different recurrence intervals for their tasks, including daily, weekly, and monthly.**

**\* \*\*Test Suite:\*\* Recurring Tasks**

**\* \*\*Test Priority:\*\* High**

**\* \*\*Preconditions:\*\***

**\* User is logged in.**

**\* A task is created.**

**\* \*\*Test Data:\*\* No test data needed.**

**\* \*\*Test Steps:\*\***

1. Navigate to the task details page.
2. Select the "Recurrence" option.
3. Choose "Daily" from the recurrence interval dropdown.
4. Repeat steps 3 with "Weekly" and "Monthly" options.

**\* \*\*Postconditions:\*\***

**\* The task is set to recur with the selected interval.**

\* \*\*Expected Result:\*\* The system successfully sets the recurrence interval for the task to daily, weekly, or monthly as selected by the user.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Functional

\* \*\*Test Case Approach:\*\* Positive

**\*\*Test Case 2\*\***

\* \*\*Test Case ID:\*\* TC\_Recurring\_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 new task instances based on the specified recurrence interval.

\* \*\*Test Suite:\*\* Recurring Tasks

\* \*\*Test Priority:\*\* High

\* \*\*Preconditions:\*\*

- \* User is logged in.

- \* A task is set to recur daily.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. Create a task and set it to recur daily.
2. Wait for 24 hours.
3. Verify that a new instance of the task is generated.

\* \*\*Postconditions:\*\*

- \* The system generates a new task instance.

\* \*\*Expected Result:\*\* The system automatically creates a new instance of the recurring task after 24 hours, reflecting the daily recurrence interval.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Functional

\* \*\*Test Case Approach:\*\* Positive

\*\*Test Case 3\*\*

\* \*\*Test Case ID:\*\* TC\_Recurring\_03

\* \*\*Test Case Title:\*\* Verify that users can modify the recurrence pattern of a recurring task.

\* \*\*Test Case Description:\*\* This test case verifies that users can change the recurrence interval of an existing recurring task.

\* \*\*Test Suite:\*\* Recurring Tasks

\* \*\*Test Priority:\*\* Medium

\* \*\*Preconditions:\*\*

- \* User is logged in.

- \* A task is set to recur weekly.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. Navigate to the task details page of the weekly recurring task.

2. Select the "Recurrence" option.

3. Change the recurrence interval to "Monthly".

4. Save changes.

\* \*\*Postconditions:\*\*

- \* The recurrence pattern is updated.

\* \*\*Expected Result:\*\* The system successfully updates the recurrence interval of the task from weekly to monthly.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Functional

\* \*\*Test Case Approach:\*\* Positive

## **\*\*Test Case 4\*\***

\* \*\*Test Case ID:\*\* TC\_Recurring\_04

\* \*\*Test Case Title:\*\* Verify that users can cancel the recurrence of a recurring task.

\* \*\*Test Case Description:\*\* This test case verifies that users can stop a task from recurring after a specific instance.

\* \*\*Test Suite:\*\* Recurring Tasks

\* \*\*Test Priority:\*\* Medium

\* \*\*Preconditions:\*\*

- \* User is logged in.

- \* A task is set to recur monthly.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. Navigate to the task details page of the monthly recurring task.
2. Select the "Recurrence" option.
3. Choose "Do not repeat" from the recurrence interval dropdown.
4. Save changes.

\* \*\*Postconditions:\*\*

- \* The recurrence pattern is removed.

\* \*\*Expected Result:\*\* The system successfully stops the recurring task from generating new instances after the current instance.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Functional

\* \*\*Test Case Approach:\*\* Positive

## **\*\*Test Case 5\*\***

\* \*\*Test Case ID:\*\* TC\_Recurring\_05

\* \*\*Test Case Title:\*\* Verify that the system notifies users of upcoming instances of recurring tasks.

\* \*\*Test Case Description:\*\* This test case verifies that users receive notifications about upcoming instances of recurring tasks.

\* \*\*Test Suite:\*\* Recurring Tasks

\* \*\*Test Priority:\*\* High

\* \*\*Preconditions:\*\*

- \* User is logged in.

- \* A task is set to recur daily.

- \* User has notification settings enabled.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. Create a task and set it to recur daily.

2. Set notification preferences to receive reminders for upcoming tasks.

3. Wait for the next day.

4. Verify that the user receives a notification about the upcoming instance of the recurring task.

\* \*\*Postconditions:\*\*

- \* The user receives a notification.

\* \*\*Expected Result:\*\* The system sends a notification to the user about the upcoming instance of the recurring task based on their notification settings.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Functional

\* \*\*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 Tags and Labels

**Feature description -**

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

**Userstory title -**

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

**Acceptance criteria -**

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

**Generated Test Cases:**

## Test Cases for Task Management Tool - Task Tags and Labels

**\*\*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.

## **\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\*** TC\_001

**\*\*Test Case Title:\*\*** Verify that users can add tags and labels to tasks.

**\*\*Test Case Description:\*\*** This test case verifies that the system allows users to add tags and labels to tasks.

The test case will create a new task and add different tags to it, ensuring that the tags are correctly applied and displayed.

**\*\*Test Suite:\*\*** Task Management - Task Tags

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in.

\* A task is created.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Go to the task details page.
2. Click on the "Add Tag" or "Add Label" field.
3. Enter a tag/label name (e.g., "Project A", "High Priority").
4. Save the task.

**\*\*Postconditions:\*\***

\* The task is saved with the added tags.

\* Tags are displayed on the task list.

**\*\*Expected Result:\*\*** The system should successfully add the tags to the task and display them on the task list.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive



**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TC\_002**

**\*\*Test Case Title:\*\* Verify that tags can be customized and edited.**

**\*\*Test Case Description:\*\* This test case verifies that users can edit existing tags and rename them or change their attributes. The test case will edit a tag's name and color, ensuring the changes are reflected in the task list and filtering functionality.**

**\*\*Test Suite:\*\* Task Management - Task Tags**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* User is logged in.

\* A tag is already created.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the tag management settings.
2. Select an existing tag.
3. Change the tag's name and/or color.
4. Save the changes.

**\*\*Postconditions:\*\***

\* The tag is updated with the changes.

**\*\*Expected Result:\*\* The system should update the tag details and reflect the changes in the task list and filtering options.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* TC\_003**

**\*\*Test Case Title:\*\* Verify that tasks can be filtered and sorted based on tags and labels.**

**\*\*Test Case Description:\*\* This test case verifies that the system allows users to filter and sort tasks based on tags and labels. The test case will create tasks with various tags and then filter the task list by specific tags, ensuring the correct results are displayed.**

**\*\*Test Suite:\*\* Task Management - Task Filtering**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged in.

\* Multiple tasks are created with various tags.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the task list.
2. Select the "Filter" or "Sort" options.
3. Choose a specific tag to filter by.
4. Check if the task list displays only tasks with the selected tag.

**\*\*Postconditions:\*\***

\* Task list is filtered based on the selected tag.

**\*\*Expected Result:\*\* The system should display only tasks with the selected tag after filtering.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* TC\_004**

**\*\*Test Case Title:\*\*** Verify that the system displays tags and labels prominently in the task list.

**\*\*Test Case Description:\*\*** This test case verifies that the system displays tags and labels clearly and prominently in the task list. The test case will create a task with multiple tags and check if the tags are easily visible in the task list view.

**\*\*Test Suite:\*\*** Task Management - Task Display

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in.

\* A task is created with multiple tags.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Go to the task list.

2. Observe the task list view.

3. Check if the tags are displayed prominently in the task list view (e.g., alongside the task title or in a separate field).

**\*\*Postconditions:\*\***

\* Task list is displayed.

**\*\*Expected Result:\*\*** The system should display the tags associated with each task clearly and prominently in the task list.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Usability

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC\_005**

**\*\*Test Case Title:\*\* Verify that users can remove tags and labels from tasks.**

**\*\*Test Case Description:\*\* This test case verifies that users can remove tags from tasks. The test case will create a task with multiple tags and remove one or more of the tags, ensuring the removal is reflected in the task list.**

**\*\*Test Suite:\*\* Task Management - Task Tags**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* User is logged in.

\* A task is created with multiple tags.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the task details page.
2. Select a tag to remove.
3. Click on the "Remove Tag" button.
4. Save the task.

**\*\*Postconditions:\*\***

\* The tag is removed from the task.

**\*\*Expected Result:\*\* The system should successfully remove the selected tag from the task and update the task list accordingly.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*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 Status Tracking

**Feature description -**

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

**Userstory title -**

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

**Acceptance criteria -**

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

**Generated Test Cases:**

## Test Cases for Task Management Tool - Task Status Tracking

**\*\*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:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** Task Status Tracking

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing,

etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_TST\_01

**\*\*Test Case Title:\*\*** Verify that users can update the status of tasks.

**\*\*Test Case Description:\*\*** This test case checks if users are able to change the status of a task from one state to another (e.g., "To Do" to "In Progress" or "In Progress" to "Completed").

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- A task is created and assigned to the user.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Go to the task list.
2. Select the created task.
3. Click on the current status of the task (e.g., "To Do").
4. Select a different status from the dropdown (e.g., "In Progress").
5. Save the changes.

**\*\*Postconditions:\*\*** The task status is updated to the newly selected status.

**\*\*Expected Result:\*\*** The task status is successfully updated in the system, and the user can see the new status reflected in the task list.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_TST\_02

**\*\*Test Case Title:\*\*** Verify that the system displays task status prominently in the task list.

**\*\*Test Case Description:\*\*** This test case verifies that the task status is clearly visible to the user in the task list, ensuring it stands out and is easily identifiable.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- Several tasks with different statuses are created.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Go to the task list.
2. Observe the task list and focus on the status display for each task.
3. Check if the status is clearly visible and identifiable for each task.

**\*\*Postconditions:\*\*** No postconditions.

**\*\*Expected Result:\*\*** The status of each task is clearly visible in the task list, using a distinct color, icon, or font style to differentiate statuses.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Usability

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***



**\*\*Test Case ID:\*\* TC\_TST\_03**

**\*\*Test Case Title:\*\* Verify that users can view the progress percentage of tasks.**

**\*\*Test Case Description:\*\* This test case verifies that the system displays a progress percentage for tasks, reflecting the completion status of the task.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.
- A task with sub-tasks is created.
- Some sub-tasks are marked as completed.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the task list.
2. Select the task with sub-tasks.
3. Observe the progress percentage displayed for the task.
4. Verify that the percentage reflects the number of completed sub-tasks compared to the total number of sub-tasks.

**\*\*Postconditions:\*\* No postconditions.**

**\*\*Expected Result:\*\* The progress percentage for the task accurately reflects the number of completed sub-tasks, giving the user a clear visual representation of task progress.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_TST\_04**

**\*\*Test Case Title:\*\*** Verify that the system provides notifications for status changes.

**\*\*Test Case Description:\*\*** This test case ensures that users receive notifications when the status of a task changes, keeping them informed about updates.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.
- A task is assigned to the user.
- The task is set to "In Progress".

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Log out of the system.
2. Have another user change the status of the assigned task to "Completed".
3. Log back into the system.
4. Verify that a notification appears, indicating the status change of the task.

**\*\*Postconditions:\*\*** The notification is displayed and can be viewed by the user.

**\*\*Expected Result:\*\*** A notification is sent to the user when the task status changes, informing them about the updated status of the task.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_TST\_05

**\*\*Test Case Title:\*\*** Verify that task status can be filtered and sorted.

**\*\*Test Case Description:\*\*** This test case ensures that users can filter and sort the task list based on task

status, allowing for efficient task management.

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.
- Multiple tasks with different statuses are created.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the task list.
2. Utilize the filter options to display only tasks with a specific status (e.g., "To Do" or "Completed").
3. Verify that only tasks with the selected status are displayed.
4. Sort the task list by status (ascending or descending order).
5. Verify that the tasks are correctly sorted based on the chosen order.

**\*\*Postconditions:\*\* No postconditions.**

**\*\*Expected Result:\*\* The system successfully filters and sorts the task list based on the selected status and order, making it easier for users to manage tasks.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

-----

**Data - 32**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Filtering and Sorting

### **Feature description -**

Users can filter and sort tasks based on various criteria such as due date, priority, assignee, or project. Filtering and sorting options help users focus on specific tasks or view tasks from different perspectives, enhancing task management and organization.

### **Userstory title -**

As a user, I want to filter and sort tasks so that I can focus on specific tasks or view tasks from different perspectives.

### **Acceptance criteria -**

['Users can filter tasks based on criteria such as due date, priority, assignee, or project.', 'Tasks can be sorted by various attributes.', 'The system provides multiple filter and sort options in the task list.', 'Users can save and apply custom filter and sort settings.', 'Filters and sorts do not affect the actual data, only the view.']

### **Generated Test Cases:**

## Test Cases for Task Filtering and Sorting Feature

**\*\*Project Context:\*\***

\* Project Name: Task Management Tool

\* Project Description: 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.

\* User Story 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:

- 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 Creation Template:\*\*

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Type of Testing	Test Case Approach
---	---	---	---	---	---	---	---	---	---	---	---	---
TC_FS_01	Verify that users can filter tasks based on due date	This test case verifies that the system allows users to filter tasks based on due date by selecting specific date ranges.	Task Filtering and Sorting	High	User is logged in and has created at least 3 tasks with different due dates	Task list with at least 3 tasks with different due dates	1. Navigate to the task list. 2. Select the filter option. 3. Choose "Due Date" as the filter criteria. 4. Select a specific date range. 5. Apply the filter.	Filtered task list displays only tasks within the selected date range.	The system correctly filters tasks based on the selected due date range. Only tasks within the chosen date range are displayed in the filtered list.	Major	Functional	Positive
TC_FS_02	Verify that tasks can be sorted by priority	This test case verifies that the system allows users to sort tasks by priority in ascending or descending order.	Task Filtering and Sorting	High	User is logged in and has created at least 3 tasks with different priorities	Task list with at least 3 tasks with different priorities	1. Navigate to the task list. 2. Select the sort option. 3. Choose "Priority" as the sort criteria. 4. Select					

"Ascending" or "Descending" order. 5. Apply the sort. | Task list is sorted by priority in the selected order. |

Tasks are correctly sorted based on priority, either in ascending or descending order. | Major | Functional | Positive |

| TC\_FS\_03 | Verify that the system provides multiple filter and sort options in the task list | This test case verifies that the system offers various filter and sort options, including due date, priority, assignee, project, and others. | Task Filtering and Sorting | High | User is logged in and has access to the task list | No test data needed | 1. Navigate to the task list. 2. Observe the available filter and sort options. | All available filter and sort options are displayed in the task list. | The system provides multiple filter and sort options, including due date, priority, assignee, project, and other relevant criteria. | Major | Functional | Positive |

| TC\_FS\_04 | Verify that users can save and apply custom filter and sort settings | This test case verifies that users can save custom filter and sort configurations and apply them later. | Task Filtering and Sorting | Medium | User is logged in and has created at least 3 tasks | Task list with at least 3 tasks | 1. Navigate to the task list. 2. Apply filters and sorts to the task list. 3. Save the custom filter and sort configuration with a unique name. 4. Select the saved configuration from the list. 5. Verify that the task list is filtered and sorted according to the saved configuration. | Saved configuration is applied to the task list, filtering and sorting tasks based on the saved settings. | Major | Functional | Positive |

| TC\_FS\_05 | Verify that filters and sorts do not affect the actual data, only the view | This test case verifies that applying filters and sorts only affects the view of the task list, not the underlying data. | Task Filtering and Sorting | High | User is logged in and has created at least 3 tasks | Task list with at least 3 tasks | 1. Navigate to the task list. 2. Apply a filter or sort to the task list. 3. Check the task details (e.g., due date, priority) of individual tasks before and after applying the filter or sort. | Task details remain unchanged, even after applying filters or sorts. | The system filters and sorts tasks without altering the actual task data. Only the displayed view of the task list is affected. | Major | Functional | 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 -**

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

\*\*Test Case 1\*\*

**\*\*Test Case ID:\*\* TT-01**

**\*\*Test Case Title:\*\* Verify that users can start and stop a timer for each task**

**\*\*Test Case Description:\*\* This test case verifies the functionality of the timer feature for tasks. Users should be able to start and stop timers for individual tasks to accurately record the time spent working on them.**

**\*\*Test Suite:\*\* Time Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged in and has created a task

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the task list and select a task.
2. Locate the "Start Timer" button for the selected task.
3. Click the "Start Timer" button to initiate the timer.
4. After a few minutes, click the "Stop Timer" button to end the timer.

**\*\*Postconditions:\*\***

- \* The timer stops.
- \* The total time spent on the task is displayed.

**\*\*Expected Result:\*\* The timer starts and stops successfully, recording the time spent on the task.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TT-02**

**\*\*Test Case Title:\*\* Verify that the system records and displays total time spent on tasks**

**\*\*Test Case Description:\*\* This test case verifies that the system accurately records and displays the total**



time spent on each task, ensuring the time tracking function is working correctly.

**\*\*Test Suite:\*\*** Time Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in and has multiple tasks with recorded time.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the task list.
2. View the details of each task.
3. Observe the "Time Spent" field for each task.

**\*\*Postconditions:\*\***

\* The "Time Spent" field accurately reflects the total time recorded for each task.

**\*\*Expected Result:\*\*** The system accurately records and displays the total time spent on each task, including the sum of multiple timer sessions.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TT-03

**\*\*Test Case Title:\*\*** Verify that users can manually enter or adjust time entries

**\*\*Test Case Description:\*\*** This test case checks if users can manually enter or adjust the time spent on tasks, offering flexibility in recording time data.

**\*\*Test Suite:\*\*** Time Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged in and has a task with recorded time.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the task list and select a task.
2. Access the time tracking details for the task.
3. Modify the existing time entry manually by entering a different value.
4. Save the changes.

**\*\*Postconditions:\*\***

\* The time entry is updated.

**\*\*Expected Result:\*\*** The system allows users to manually enter or adjust the time entries for tasks, updating the recorded time accordingly.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TT-04

**\*\*Test Case Title:\*\*** Verify that time tracking data is stored and can be exported

**\*\*Test Case Description:\*\*** This test case ensures that time tracking data is stored securely and can be exported for analysis or reporting purposes.

**\*\*Test Suite:\*\*** Time Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged in and has multiple tasks with recorded time.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the "Time Tracking" section or settings.
2. Locate the "Export Time Data" option.
3. Select the desired format for export (e.g., CSV, PDF).
4. Initiate the export process.

**\*\*Postconditions:\*\***

\* Time tracking data is successfully exported to the chosen format.

**\*\*Expected Result:\*\*** The system stores time tracking data persistently and allows users to export the data in various formats.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** TT-05

**\*\*Test Case Title:\*\*** Verify that the system provides reports on time spent per task or project

**\*\*Test Case Description:\*\*** This test case checks the functionality of time tracking reports, which provide insights into time allocation for tasks and projects.

**\*\*Test Suite:\*\*** Time Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged in and has multiple tasks assigned to different projects.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the "Reports" section or settings.
2. Select the "Time Spent" report option.
3. Choose a reporting period (e.g., daily, weekly, monthly).

4. Filter the report by task or project.

5. View the generated report.

**\*\*Postconditions:\*\***

\* A report is generated and displayed.

**\*\*Expected Result:\*\*** The system provides reports that display time spent per task or project, with filter options for various time periods and categories.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 34**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Templates

### **Feature description -**

Users can create and use templates for common task types or project workflows, streamlining task creation and standardizing processes across projects. Task templates can include predefined task structures, descriptions, assignees, and due dates, saving time and ensuring consistency in task execution.

### **Userstory title -**

As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

## Acceptance criteria -

['Users can create and save task templates.', 'Templates can include predefined task structures, descriptions, assignees, and due dates.', 'Users can apply templates to new tasks or projects.', 'Templates can be edited or deleted as needed.', 'The system displays a list of available templates for selection.']

## Generated Test Cases:

## Test Cases for Task Management Tool - Task Templates

**Project Name:** Task Management Tool

**Feature Name:** Task Templates

**User Story Title:** 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 1**

**Test Case ID:** TT-001

**Test Case Title:** Verify that users can create and save task templates.

**\*\*Test Case Description:\*\*** This test case verifies that users can successfully create and save task templates with basic information.

**\*\*Test Suite:\*\*** Task Templates

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- User is authorized to create templates

**\*\*Test Data:\*\***

- Template Name: "Project Kick-Off"
- Description: "Initial tasks for a new project"
- Assignee: "Default Team"
- Due Date: 2 days from now

**\*\*Test Steps:\*\***

1. Navigate to the "Templates" section.
2. Click on "Create New Template".
3. Enter the template name, description, assignee, and due date.
4. Click "Save Template".

**\*\*Postconditions:\*\***

- Template is saved successfully
- Template is visible in the list of available templates

**\*\*Expected Result:\*\*** The system creates and saves the template with the provided information.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TT-002**

**\*\*Test Case Title:\*\* Verify that templates can include predefined task structures, descriptions, assignees, and due dates.**

**\*\*Test Case Description:\*\* This test case ensures that templates can incorporate predefined task structures, descriptions, assignees, and due dates.**

**\*\*Test Suite:\*\* Task Templates**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- User is authorized to create templates

**\*\*Test Data:\*\***

- Template Name: "Design Sprint"
- Task 1:
  - Title: "Research & Ideation"
  - Description: "Gather research, brainstorm ideas, create wireframes"
  - Assignee: "Design Team"
  - Due Date: 1 day from now
- Task 2:
  - Title: "Prototyping"
  - Description: "Build interactive prototypes"
  - Assignee: "Design Team"
  - Due Date: 2 days from now

**\*\*Test Steps:\*\***

1. Navigate to the "Templates" section.
2. Click on "Create New Template".
3. Add multiple tasks with different titles, descriptions, assignees, and due dates.

4. Click "Save Template".

**\*\*Postconditions:\*\***

- Template is saved successfully
- All task information is preserved

**\*\*Expected Result:\*\*** The system saves the template with all the predefined tasks and their details.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TT-003

**\*\*Test Case Title:\*\*** Verify that users can apply templates to new tasks or projects.

**\*\*Test Case Description:\*\*** This test case checks if users can apply existing templates to create new tasks or projects.

**\*\*Test Suite:\*\*** Task Templates

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- A template exists
- User is authorized to create tasks/projects

**\*\*Test Data:\*\***

- Existing Template: "Project Kick-Off"
- New Task Title: "Project Alpha"

**\*\*Test Steps:\*\***

1. Navigate to the "Tasks" or "Projects" section.



2. Click on "Create New Task/Project".
3. Select the "Project Kick-Off" template.
4. Click "Create Task/Project".

**\*\*Postconditions:\*\***

- New task/project is created
- The task/project includes the details from the selected template

**\*\*Expected Result:\*\*** The system creates a new task or project with the details from the selected template.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TT-004

**\*\*Test Case Title:\*\*** Verify that templates can be edited or deleted as needed.

**\*\*Test Case Description:\*\*** This test case ensures that users can edit or delete existing templates.

**\*\*Test Suite:\*\*** Task Templates

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in
- A template exists
- User is authorized to edit/delete templates

**\*\*Test Data:\*\***

- Template Name: "Design Sprint"
- New Template Name: "Design Sprint 2.0"

**\*\*Test Steps:\*\***

1. Navigate to the "Templates" section.
2. Select the "Design Sprint" template.
3. Click "Edit Template".
4. Modify the template name to "Design Sprint 2.0" and save changes.
5. Select the "Design Sprint 2.0" template.
6. Click "Delete Template".
7. Confirm the deletion.

**\*\*Postconditions:\*\***

- Template is edited successfully
- Template is deleted successfully

**\*\*Expected Result:\*\*** The system allows users to edit or delete existing templates.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** TT-005

**\*\*Test Case Title:\*\*** Verify that the system displays a list of available templates for selection.

**\*\*Test Case Description:\*\*** This test case checks if the system displays a clear and accessible list of available templates to users.

**\*\*Test Suite:\*\*** Task Templates

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in
- Multiple templates exist

**\*\*Test Data:\*\***

- Existing Templates: "Project Kick-Off", "Design Sprint 2.0"

**\*\*Test Steps:\*\***

1. Navigate to the "Tasks" or "Projects" section.
2. Click on "Create New Task/Project".
3. Observe the "Templates" section for available options.

**\*\*Postconditions:\*\***

- The list of templates is displayed
- Templates are listed in a clear and organized manner

**\*\*Expected Result:\*\*** The system displays a list of available templates to the user for selection.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Usability

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 35**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Calendar Integration

**Feature description -**

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments

alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

### **Userstory title -**

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

### **Acceptance criteria -**

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

### **Generated Test Cases:**

## Test Cases for Task Management Tool Calendar Integration

**\*\*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 1:\*\***

**\*\*Test Case ID:\*\* TC-001**

**\*\*Test Case Title:\*\* Verify that users can sync tasks with external calendars (Google Calendar)**

**\*\*Test Case Description:\*\* This test case verifies that users can connect and sync their tasks with their Google Calendar accounts.**

**\*\*Test Suite:\*\* Calendar Integration**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the task management tool.
- User has a Google Calendar account.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the Calendar Integration settings within the task management tool.
2. Select "Google Calendar" as the calendar provider.
3. Follow the prompts to authorize the task management tool to access the user's Google Calendar account.
4. Create a new task within the tool and set a due date.

**\*\*Postconditions:\*\***

- The task is successfully synced with the user's Google Calendar.

**\*\*Expected Result:\*\* The task with its assigned due date appears in the user's Google Calendar.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Integration Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC-002**

**\*\*Test Case Title:\*\* Verify that task deadlines are displayed in the calendar view**

**\*\*Test Case Description:\*\* This test case checks that tasks with deadlines are correctly displayed on the**

calendar view within the task management tool.

**\*\*Test Suite:\*\*** Calendar Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User has synced their calendar with the task management tool.
- A task with a deadline is created.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the calendar view within the task management tool.
2. Verify that the task deadline is displayed on the corresponding date in the calendar.

**\*\*Postconditions:\*\***

- The calendar view is accessed and reviewed.

**\*\*Expected Result:\*\*** The task deadline is accurately displayed as an event on the specified date in the calendar view.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC-003

**\*\*Test Case Title:\*\*** Verify that users can set reminders for tasks via the calendar

**\*\*Test Case Description:\*\*** This test case verifies that users can set reminders for tasks within the calendar view of the task management tool.

**\*\*Test Suite:\*\*** Calendar Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User has synced their calendar with the task management tool.

- A task with a deadline is created.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the calendar view within the task management tool.

2. Select the task with a deadline.

3. Choose the "Reminder" option and set a reminder time.

**\*\*Postconditions:\*\***

- The reminder is set for the selected task.

**\*\*Expected Result:\*\*** The system sets a reminder for the task at the specified time, either via calendar notifications or other communication channels.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC-004

**\*\*Test Case Title:\*\*** Verify that calendar events update automatically when tasks are changed

**\*\*Test Case Description:\*\*** This test case checks that changes made to a task, such as updating the deadline, are automatically reflected in the synced calendar.

**\*\*Test Suite:\*\*** Calendar Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User has synced their calendar with the task management tool.

- A task with a deadline is created.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task management tool and edit the task's deadline.
2. Verify that the updated deadline is reflected in the synced calendar.

**\*\*Postconditions:\*\***

- The task's deadline is modified in the task management tool.

**\*\*Expected Result:\*\*** The corresponding calendar event for the task is automatically updated with the new deadline.

**\*\*Severity:\*\*** Critical

**\*\*Type of Testing:\*\*** Integration Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC-005

**\*\*Test Case Title:\*\*** Verify that users can manage calendar sync settings within the app

**\*\*Test Case Description:\*\*** This test case verifies that users can manage their calendar integration settings, such as enabling or disabling syncing, choosing a different calendar provider, or modifying sync frequency.

**\*\*Test Suite:\*\*** Calendar Integration

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User has synced their calendar with the task management tool.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the Calendar Integration settings within the task management tool.
2. Explore the available options for managing sync settings (e.g., enabling/disabling sync, choosing a different calendar provider, adjusting sync frequency).
3. Change one or more settings and verify the changes are applied.



**\*\*Postconditions:\*\***

- Calendar integration settings are modified.

**\*\*Expected Result:\*\*** The system allows users to adjust the calendar integration settings as needed, and the changes are applied correctly.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 36**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Duplication

### **Feature description -**

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

### **Userstory title -**

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

### **Acceptance criteria -**

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including

descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.')

**Generated Test Cases:**

## Test Cases for Task Duplication Feature

**Project Name:** Task Management Tool

**Feature Name:** Task Duplication

**User Story Title:** 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 Creation Template:**

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Type of Testing	Test Case Approach
TC_DUP_01	Verify that users can duplicate tasks	This test case ensures that users can duplicate existing tasks.	Task Duplication	High	- User is logged in. - A task is created.	Existing task details: title,						

description, assignee, deadline, etc. | 1. Navigate to the task list. 2. Locate the task to be duplicated. 3. Click on the "Duplicate" button/option. | - A new task is created. | A new task identical to the original task is created and displayed in the task list. | Major | Functional | Positive |

| TC\_DUP\_02 | Verify that the duplicated task retains all original details | This test case checks if the duplicated task retains all details from the original task. | Task Duplication | High | - User is logged in. - A task is created with details: title, description, assignee, deadline. | Existing task details: title, description, assignee, deadline, etc. | 1. Navigate to the task list. 2. Locate the task to be duplicated. 3. Click on the "Duplicate" button/option. 4. Verify the details of the newly created task. | - The duplicated task is displayed. | The duplicated task displays the same title, description, assignee, deadline, and other details as the original task. | Critical | Functional | Positive |

| TC\_DUP\_03 | Verify that users can modify the duplicated task | This test case ensures that users can edit the duplicated task after creation. | Task Duplication | Medium | - User is logged in. - A task is duplicated. | Existing task details: title, description, assignee, deadline, etc. | 1. Navigate to the task list. 2. Locate the duplicated task. 3. Edit the title, description, assignee, or deadline. 4. Save the changes. | - The task details are updated. | The system allows modifications to the duplicated task and updates the task details accordingly. | Minor | Functional | Positive |

| TC\_DUP\_04 | Verify that the system clearly indicates task duplication | This test case verifies that the system provides clear visual feedback after task duplication. | Task Duplication | Medium | - User is logged in. - A task is created. | Existing task details: title, description, assignee, deadline, etc. | 1. Navigate to the task list. 2. Locate the task to be duplicated. 3. Click on the "Duplicate" button/option. 4. Observe the system feedback. | - Duplication is confirmed. | The system displays a clear notification or visual cue confirming that the task has been duplicated. | Minor | Usability | Positive |

| TC\_DUP\_05 | Verify that duplicated tasks are saved and displayed in the task list | This test case ensures that duplicated tasks are correctly saved and displayed in the task list. | Task Duplication | High | - User is logged in. - A task is duplicated. | Existing task details: title, description, assignee, deadline, etc. | 1. Navigate to the task list. 2. Locate the task to be duplicated. 3. Click on the "Duplicate" button/option. 4. Check for the duplicated task in the task list. | - Duplicated task is present. | The duplicated task is correctly saved and

appears in the task list alongside the original task. | Major | Functional | 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:**

## Test Cases for Task Management Tool - Task History Feature

**\*\*Project Context:\*\***

Project Name: Task Management Tool

Project Description: A 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.

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 ID**	**Test Case Title**	**Test Case Description**	**Test Suite**	**Test Priority**	**Preconditions**	**Test Data**	**Test Steps**	**Postconditions**	**Expected Result**	**Severity**	**Type of Testing**	**Test Case Approach**
TC_TH_01	Verify that users can view a detailed history of changes to tasks	This test case ensures that users can access and view a comprehensive history of changes made to a task.	Task History	High	-	User is						

logged in. - A task with at least one modification exists. | No test data needed | 1. Navigate to the task details page. 2. Locate the task history section. 3. Verify that the history section displays a list of changes made to the task. | - Task history section is displayed. | The system displays a detailed history of all changes made to the task. | Major | Functional | Positive |

| TC\_TH\_02 | Verify that the history includes modifications, comments, and other relevant activity | This test case validates that the task history accurately captures all relevant changes, including modifications, comments, and other activity. | Task History | High | - User is logged in. - A task exists with modifications, comments, and other activity. | No test data needed | 1. Navigate to the task details page. 2. Access the task history section. 3. Verify that the history includes entries for modifications, comments, and other relevant activity. | - Task history is displayed. | The history section accurately reflects all modifications, comments, and other relevant activity related to the task. | Major | Functional | Positive |

| TC\_TH\_03 | Verify that users can filter the history by type of change or date | This test case ensures users can filter the task history to view specific changes based on type or date. | Task History | Medium | - User is logged in. - A task exists with a history of changes. | No test data needed | 1. Access the task details page. 2. Navigate to the task history section. 3. Use the filter options to filter the history by type of change (e.g., modification, comment) or date range. 4. Verify that the filtered results only display changes matching the selected criteria. | - Task history is filtered according to the user's selections. | The system displays only the task history entries that match the applied filter criteria. | Minor | Functional | Positive |

| TC\_TH\_04 | Verify that the system provides timestamps for all changes | This test case validates that timestamps are associated with each change in the task history, providing a clear record of when changes were made. | Task History | Medium | - User is logged in. - A task exists with a history of changes. | No test data needed | 1. Access the task details page. 2. View the task history section. 3. Check each entry in the history and verify that a timestamp is associated with each change. | - Task history entries are displayed. | The system displays a timestamp for each change in the task history, indicating the time and date of the modification. | Minor | Functional | Positive |

| TC\_TH\_05 | Verify that task history is accessible from the task details page | This test case ensures that users can easily access the task history from the task details page. | Task History | High | - User is logged in. - A task

exists with a history of changes. | No test data needed | 1. Navigate to the task details page. 2. Locate the task history section or button. 3. Access the task history section. | - Task history section is displayed. | The task history is readily accessible from the task details page, allowing users to easily view changes made to the task. | Minor | Usability | Positive |

---

## **Data - 38**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Notifications

### **Feature description -**

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

### **Userstory title -**

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

### **Acceptance criteria -**

['Users receive notifications for task updates, comments, mentions, and other relevant activities.',

'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.',

'The system logs notification history for future reference.']

**Generated Test Cases:**

## Test Cases for Task Management Tool - Task Notifications

**\*\*Project Context:\*\***

- \* Project Name: Task Management Tool
- \* Project Description: A tool for users to create, assign, prioritize, and track tasks and projects.
- \* Feature Name: Task Notifications
- \* Feature Description: Users receive notifications for task updates, comments, mentions, and other relevant activity.
- \* User Story Title: As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

**\*\*Acceptance Criteria:\*\***

1. Users receive notifications for task updates, comments, mentions, and other relevant activities.
2. Notifications can be delivered via email, mobile push notifications, or in-app alerts.
3. Users can customize notification settings based on preferences.
4. Notifications provide detailed information about the activity.
5. The system logs notification history 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 | Type of Testing | Test Case Approach |



|---|---|---|---|---|---|---|---|---|---|

| TC-TN-01 | Verify that users receive notifications for task updates | This test case checks whether a user receives a notification when a task they are assigned to is updated. | Task Notifications | High | User is logged in, task is created and assigned to user | Task details, user preferences | 1. Create a task and assign it to the user. 2. Update the task description. | User receives a notification about the update | The user receives a notification via email, push notification, or in-app alert informing them about the task update. | Major | Functional | Positive |

| TC-TN-02 | Verify that users receive notifications for comments on tasks | This test case verifies if users get notified about comments posted on tasks they are involved in. | Task Notifications | High | User is logged in, task is created with comments | Task details, user preferences | 1. Create a task and assign it to the user. 2. Another user posts a comment on the task. | User receives a notification about the comment | The user receives a notification about the comment posted on the task via email, push notification, or in-app alert. | Major | Functional | Positive |

| TC-TN-03 | Verify that users can customize notification settings | This test case confirms if users can modify their notification preferences according to their needs. | Task Notifications | Medium | User is logged in | User preferences | 1. Go to notification settings. 2. Disable email notifications for task updates. 3. Save changes. | Notification settings are updated | The user's notification preferences are successfully updated, and they no longer receive email notifications for task updates. | Minor | Functional | Positive |

| TC-TN-04 | Verify that notifications provide detailed information about the activity | This test case ensures that notifications contain relevant information about the task and activity. | Task Notifications | High | User is logged in, task is created with updates, comments, mentions | Task details, user preferences | 1. Create a task and assign it to the user. 2. Update the task and add a comment mentioning the user. | User receives a notification | The notification received by the user clearly displays the task name, task status, the type of activity (update, comment, mention), and relevant details. | Major | Functional | Positive |

| TC-TN-05 | Verify that the system logs notification history | This test case checks if the system maintains a record of all notifications sent to users. | Task Notifications | Low | User is logged in, notifications are sent | User preferences, notification logs | 1. Create a task and assign it to the user. 2. Update the task and add a

comment. 3. Check notification history. | Notification history is available | The system successfully logs all notifications sent to the user, including date, time, content, and related task information. | Minor | System | Positive |

---

## **Data - 39**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Archiving

### **Feature description -**

Completed or inactive tasks can be archived or moved to a separate section, decluttering the task list and maintaining focus on active tasks. Task archiving helps users keep their task lists organized and prioritized, without losing access to important historical data.

### **Userstory title -**

As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks.

### **Acceptance criteria -**

['Users can archive completed or inactive tasks.', 'Archived tasks are moved to a separate section for easy access.', 'Users can search and view archived tasks.', 'The system provides an option to restore archived tasks to the active list.', 'Archived tasks retain all original details and history.']

### **Generated Test Cases:**

## ## Test Cases for Task Archiving Feature

**\*\*Project Name:\*\* Task Management Tool**

**\*\*Feature Name:\*\* Task Archiving**

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC\_Archiving\_01**

**\*\*Test Case Title:\*\* Verify that users can archive completed or inactive tasks.**

**\*\*Test Case Description:\*\* This test case validates the functionality of archiving tasks, ensuring that users can move completed or inactive tasks to the archive section.**

**\*\*Test Suite:\*\* Task Archiving**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- User has created at least one task
- The task is marked as completed or inactive

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the task list.
2. Select the completed or inactive task.
3. Locate the "Archive" option (button, icon, or menu).
4. Click on the "Archive" option.

**\*\*Postconditions:\*\***

- The task is removed from the active task list.
- The task is displayed in the archive section.

**\*\*Expected Result:\*\* The system successfully archives the selected task, moving it to the archive section.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TC\_Archiving\_02**

**\*\*Test Case Title:\*\* Verify that archived tasks are moved to a separate section for easy access.**

**\*\*Test Case Description:\*\* This test case checks if the archived tasks are moved to a distinct section within the application, ensuring easy access and distinction from active tasks.**

**\*\*Test Suite:\*\* Task Archiving**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- At least one task is archived

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the archive section.
2. Verify that the section is clearly labeled as "Archive" or similar.
3. Confirm that the archived task(s) are displayed in this section.

**\*\*Postconditions:\*\***

- The archive section displays all archived tasks.

**\*\*Expected Result:\*\* The system displays all archived tasks in a dedicated section, distinct from the active task list.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* TC\_Archiving\_03**

**\*\*Test Case Title:\*\* Verify that users can search and view archived tasks.**

**\*\*Test Case Description:\*\* This test case ensures that users can effectively search for and view archived tasks, making it easy to locate specific information from previously completed tasks.**

**\*\*Test Suite:\*\* Task Archiving**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in
- At least one task is archived

**\*\*Test Data:\*\***

- Search term relevant to archived task (e.g., task name, description)

**\*\*Test Steps:\*\***

1. Navigate to the archive section.
2. Use the search bar within the archive section.
3. Enter a search term relevant to an archived task.
4. Perform the search.

**\*\*Postconditions:\*\***

- The search results display the matching archived task(s).

**\*\*Expected Result:\*\* The system displays relevant archived tasks based on the search term entered.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* TC\_Archiving\_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 validates the ability to restore an archived task back to the active task list, ensuring users can retrieve previously completed tasks if needed.**

**\*\*Test Suite:\*\* Task Archiving**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in
- At least one task is archived

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the archive section.
2. Select the desired archived task.
3. Locate the "Restore" option (button, icon, or menu).
4. Click on the "Restore" option.

**\*\*Postconditions:\*\***

- The task is removed from the archive section.
- The task is displayed in the active task list.

**\*\*Expected Result:\*\* The system successfully restores the selected archived task to the active task list.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC\_Archiving\_05**

**\*\*Test Case Title:\*\*** Verify that archived tasks retain all original details and history.

**\*\*Test Case Description:\*\*** This test case ensures that all original task details and history (including deadlines, descriptions, attachments, and collaborators) are preserved when a task is archived.

**\*\*Test Suite:\*\*** Task Archiving

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in
- At least one task is archived, including details like deadlines, descriptions, attachments, and collaborators

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Archive a task with multiple details (deadline, description, attachment, collaborator).
2. Navigate to the archive section.
3. View the archived task.
4. Verify that all original details (deadline, description, attachment, collaborator) are displayed.

**\*\*Postconditions:\*\***

- The archived task displays all original details.

**\*\*Expected Result:\*\*** The archived task retains all its original details and history.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 40**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Guest Access

### **Feature description -**

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

### **Userstory title -**

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

### **Acceptance criteria -**

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

### **Generated Test Cases:**

## Test Cases for Guest Access Feature

**\*\*Project Context:\*\***

**\* \*\*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.

**\*\*User Story Title:\*\*** As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

**\*\*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 Creation Template:\*\***

**\*\*Test Case ID:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module)

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\*** TC\_GA\_01

**\*\*Test Case Title:\*\*** Verify that users can invite external collaborators as guests with limited access.

**\*\*Test Case Description:\*\*** This test case verifies that users can invite external collaborators as guests, granting them limited access to specific tasks or projects.

**\*\*Test Suite:\*\*** Guest Access

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A user is logged in with full platform access.

\* A task or project exists.

**\*\*Test Data:\*\***

\* Valid email address of the external collaborator.

**\*\*Test Steps:\*\***

1. Navigate to the task or project.
2. Click on the "Share" or "Collaborators" option.
3. Enter the valid email address of the external collaborator.

4. Select the "Guest" access level.

5. Send the invitation.

**\*\*Postconditions:\*\***

\* The invitation is sent to the collaborator's email address.

\* The system records the guest collaborator's information.

**\*\*Expected Result:\*\*** The system successfully invites the external collaborator as a guest with limited access to the selected task or project.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** TC\_GA\_02

**\*\*Test Case Title:\*\*** Verify that guests can view and collaborate on specific tasks or projects.

**\*\*Test Case Description:\*\*** This test case verifies that guests can view and interact with specific tasks or projects they are granted access to.

**\*\*Test Suite:\*\*** Guest Access

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A guest user has been invited and accepted the invitation.

\* A task or project is shared with the guest user.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. The guest user logs in to the platform.

2. Navigate to the shared task or project.

3. View the task or project details.

4. Add a comment or complete a subtask within the shared task or project.

**\*\*Postconditions:\*\***

\* The guest user's activity is logged.

\* The task or project is updated with the guest user's contribution.

**\*\*Expected Result:\*\*** The guest user can access and interact with the shared task or project, with limitations based on their guest access level.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TC\_GA\_03

**\*\*Test Case Title:\*\*** Verify that the system provides clear distinctions between guest and full access roles.

**\*\*Test Case Description:\*\*** This test case verifies that the system clearly differentiates between guest and full access users, ensuring that each user type has appropriate permissions and visibility.

**\*\*Test Suite:\*\*** Guest Access

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* A guest user and a full access user are logged in.

\* A task or project exists.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Both users navigate to the same task or project.

2. Observe the functionalities available to each user (e.g., viewing, editing, deleting).

3. Compare the user interfaces (e.g., user profiles, dashboards, navigation menus).

**\*\*Postconditions:\*\***

\* No changes are made to the system.

**\*\*Expected Result:\*\*** The system should clearly distinguish between guest and full access users by displaying different interfaces, functionalities, and permissions.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Usability

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TC\_GA\_04

**\*\*Test Case Title:\*\*** Verify that users can manage and revoke guest access at any time.

**\*\*Test Case Description:\*\*** This test case verifies that the user with full access can manage and revoke guest access to shared tasks or projects.

**\*\*Test Suite:\*\*** Guest Access

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A guest user has been invited and accepted the invitation.

\* A task or project is shared with the guest user.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. The user with full access navigates to the shared task or project.
2. Click on the "Collaborators" or "Share" option.
3. Locate the guest user's entry.
4. Revoke the guest user's access.

**\*\*Postconditions:\*\***

\* The guest user is no longer able to access the task or project.

\* The system records the revocation event.

**\*\*Expected Result:\*\*** The system successfully revokes the guest user's access to the task or project, preventing further access and collaboration.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Negative

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** TC\_GA\_05

**\*\*Test Case Title:\*\*** Verify that guests cannot access sensitive or restricted areas of the platform.

**\*\*Test Case Description:\*\*** This test case verifies that guests cannot access features or areas that require full access permissions.

**\*\*Test Suite:\*\*** Guest Access

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A guest user has been invited and accepted the invitation.

\* The system has restricted areas or features requiring full access.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. The guest user attempts to access a sensitive or restricted area of the platform (e.g., user management, system settings, payment information).

2. Monitor the user's actions and the system's response.

**\*\*Postconditions:\*\***

\* The guest user is prevented from accessing the restricted area.

\* The system logs the attempted access.

**\*\*Expected Result:\*\*** The system should block the guest user's access to restricted areas, displaying error messages or redirecting them to appropriate pages.

**\*\*Severity:\*\*** Critical

**\*\*Type of Testing:\*\*** Security

**\*\*Test Case Approach:\*\*** Negative

-----

**Data - 41**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Import/Export

**Feature description -**

Users can import tasks from external sources or export tasks to other task management tools or file formats. Import/export capabilities facilitate data migration and integration with other systems, ensuring seamless collaboration and workflow management across platforms.

**Userstory title -**

As a user, I want to import and export tasks so that I can migrate data between different task management tools.

**Acceptance criteria -**

['Users can import tasks from external sources in various formats (e.g., CSV, Excel).', 'Users can export tasks to other task management tools or file formats.', 'The system provides clear mapping of fields during import/export.', 'Users can preview imported data before finalizing the process.', 'The system logs import/export activities for future reference.']

**Generated Test Cases:**

## Test Cases for Task Management Tool - Task Import/Export

**\*\*Project Context:\*\***

- \* **\*\*Project Name:\*\*** Task Management Tool
- \* **\*\*Project Description:\*\*** Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects.
- \* **\*\*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.
- \* **\*\*User Story Title:\*\*** 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	Type of Testing	Test Case Approach	



|---|---|---|---|---|---|---|---|---|---|

| TC-IMP-01 | Verify that users can import tasks from CSV files | Ensure that the system allows users to import tasks from CSV files with correct field mapping. | Task Import/Export | High | User is logged in. | CSV file with task data (Task name, description, due date, priority, assignee). | 1. Navigate to the import tasks section. 2. Select CSV file format. 3. Upload the CSV file. 4. Verify that the system maps the CSV file columns to the task fields. 5. Review the imported task data. | Imported tasks are displayed with correct data and field mapping. | Imported tasks are successfully added to the task list. | Major | Functional Testing | Positive |

| TC-IMP-02 | Verify that users can preview imported data before finalizing the import process | Ensure that the system allows users to preview imported task data before confirming the import. | Task Import/Export | High | User is logged in. Imported data is available. | CSV file with task data. | 1. Navigate to the import tasks section. 2. Upload the CSV file. 3. Review the previewed task data displayed. 4. Verify that the data is accurate and complete. | Previewed data is accurate and displayed correctly. | Imported tasks are successfully added to the task list. | Major | Functional Testing | Positive |

| TC-EXP-03 | Verify that users can export tasks to Excel files | Ensure that the system allows users to export tasks to Excel files with correct field data. | Task Import/Export | High | User is logged in. Tasks are created in the task list. | No test data needed | 1. Select the tasks to export. 2. Choose Excel file format. 3. Export the selected tasks. 4. Download the generated Excel file. 5. Open the Excel file and verify the exported data. | Exported Excel file is successfully generated and downloaded. | Exported data is complete and accurate. | Major | Functional Testing | Positive |

| TC-LOG-04 | Verify that the system logs import/export activities | Ensure that the system logs all import and export activities for future reference. | Task Import/Export | Medium | User is logged in. Tasks are imported and exported. | No test data needed | 1. Import tasks from a CSV file. 2. Export tasks to an Excel file. 3. Check the system logs for import and export activities. 4. Verify that the logs contain timestamp, user details, file name, and import/export type. | Import and export logs are successfully recorded with complete information. | Minor | Logging Testing | Positive |

| TC-IMP-05 | Verify that the system handles invalid data during import | Ensure that the system gracefully

handles invalid data during import and provides informative error messages. | Task Import/Export | Medium | User is logged in. | CSV file with invalid data (incorrect field format, missing data). | 1. Upload a CSV file with invalid data. 2. Observe the system's response and error messages. 3. Verify that the system prevents the import of invalid data and displays a clear error message. | Invalid data is not imported and a clear error message is displayed to the user. | Minor | Error Handling Testing | Negative |

-----

## **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

**Feature Name:** Task Prioritization

**User Story Title:** 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:** | **Type of Testing:** | **Test Case Approach:**

----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | -----  
**TC-01** | **Verify that users can set and adjust task priorities** | This test case checks if users can assign and modify priority levels for tasks. It ensures the system accurately reflects the chosen priority for each task. |

Task Prioritization | High | - User is logged in<br>- A task is created | Priority levels: Low, Medium, High | 1. Create a new task.<br>2. Select the "Priority" option.<br>3. Choose a priority level (e.g., High).<br>4. Save the task.<br>5. Edit the task and change the priority to Medium.<br>6. Save changes. | - The task's priority is updated.<br>- The system reflects the new priority level for the task. | The system allows users to set and adjust task priorities successfully. | Major | Functional | Positive

**\*\*TC-02\*\*** | **\*\*Verify that tasks can be sorted and filtered based on priority levels\*\*** | This test case validates the ability to organize and filter tasks based on their assigned priorities. It ensures that tasks are displayed in the expected order and that filters function correctly. | Task Prioritization | High | - User is logged in<br>- Multiple tasks with varying priorities exist | Priority levels: Low, Medium, High | 1. Create multiple tasks with different priorities.<br>2. Sort the tasks by "Priority" (ascending or descending).<br>3. Filter the tasks by a specific priority level (e.g., High). | - Tasks are sorted correctly based on the chosen priority level.<br>- Tasks with the selected priority are displayed and others are hidden. | The system successfully sorts and filters tasks based on their priority levels. | Major | Functional | Positive

**\*\*TC-03\*\*** | **\*\*Verify that the system provides visual indicators of task priority\*\*** | This test case assesses whether the system uses visual cues to indicate the priority level of each task. It checks if these indicators are easily recognizable and consistent with the assigned priority. | Task Prioritization | Medium | - User is logged in<br>- Multiple tasks with varying priorities exist | Priority levels: Low, Medium, High | 1. Create multiple tasks with different priority levels.<br>2. Observe the visual indicators (e.g., color, icon) associated with each task.<br>3. Compare the visual indicators to the assigned priority level. | - Each task displays a visual indicator that corresponds to its priority level.<br>- Visual indicators are distinct and easy to differentiate. | The system effectively uses visual indicators to highlight task priority levels. | Minor | Usability | Positive

**\*\*TC-04\*\*** | **\*\*Verify that users receive notifications for high-priority tasks\*\*** | This test case checks if the system sends notifications to users about upcoming deadlines for tasks marked as high priority. It verifies that notifications are timely and contain relevant information. | Task Prioritization | High | - User is logged in<br>- A high-priority task is created with a deadline | High-priority task with a deadline | 1. Create a high-priority task with a deadline.<br>2. Observe if a notification is received before the deadline.<br>3. Check if the notification includes task details and deadline information. | - A notification is received before the deadline for

the high-priority task.<br>- The notification displays task details and the approaching deadline. | The system effectively sends notifications for high-priority tasks with deadlines. | Major | Functional | Positive

**\*\*TC-05\*\*** | **\*\*Verify that priority changes are logged and tracked\*\*** | This test case validates the system's ability to record changes made to task priorities. It ensures that a history of priority updates is available for review and analysis. | Task Prioritization | Medium | - User is logged in<br>- A task has been created and its priority changed | Task with initial and changed priority levels | 1. Create a task and assign a priority level (e.g., Medium).<br>2. Change the priority level to High.<br>3. Check the task history or audit log for the priority change. | - The task history or audit log records the initial and changed priority levels.<br>- The history includes information on who made the change and when it was made. | The system accurately logs and tracks changes made to task priorities. | Minor | Functional | Positive

These test cases are designed to cover all the acceptance criteria for the task prioritization feature. They ensure that the system functions correctly, provides user-friendly features, and maintains a record of changes. By executing these test cases, you can identify potential defects early in the development cycle and contribute to a high-quality software product.

-----

**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 Name:\*\* Task Management Tool**

**\*\*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:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Task Collaboration

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

- User is logged in.
- Team members are added to the project.
- A task is created.

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_001

**\*\*Test Case Title:\*\*** Verify that users can assign tasks to team members.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of assigning a task to a specific team member.

**\*\*Test Priority:\*\* High**

**\*\*Test Steps:\*\***

1. Go to the task list.
2. Select the task to be assigned.
3. Click on the "Assign" button.
4. Choose a team member from the dropdown list.
5. Confirm the assignment.

**\*\*Postconditions:\*\***

- The task is assigned to the selected team member.
- The task's assignee is updated to reflect the assigned team member.

**\*\*Expected Result:\*\*** The system successfully assigns the selected task to the chosen team member.

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC\_002**

**\*\*Test Case Title:\*\*** Verify that users can share updates and communicate through comments or messages.

**\*\*Test Case Description:\*\*** This test case verifies the ability to add comments and messages to a task, facilitating communication among team members.

**\*\*Test Priority:\*\* High**

**\*\*Test Steps:\*\***

1. Go to the task details page.
2. Click on the "Add Comment" or "Message" field.
3. Type a comment or message.
4. Submit the comment or message.



**\*\*Postconditions:\*\***

- The comment or message is added to the task.
- The comment or message is visible to all assigned team members.

**\*\*Expected Result:\*\*** The system adds the comment or message to the task, making it visible to the assigned team members.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_003

**\*\*Test Case Title:\*\*** Verify that task updates and comments are visible to assigned team members.

**\*\*Test Case Description:\*\*** This test case checks if assigned team members can view updates and comments made on a task.

**\*\*Test Priority:\*\*** High

**\*\*Test Steps:\*\***

1. Assign a task to a team member.
2. As the task creator, add a comment or update to the task.
3. Log in as the assigned team member.
4. Go to the task details page.

**\*\*Postconditions:\*\***

- The comment or update is visible to the assigned team member.

**\*\*Expected Result:\*\*** The assigned team member can view the added comments and updates on the task.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_004**

**\*\*Test Case Title:\*\* Verify that users receive notifications for task updates and comments.**

**\*\*Test Case Description:\*\* This test case ensures that users receive notifications when there are updates or comments made on a task they are assigned to.**

**\*\*Test Priority:\*\* High**

**\*\*Test Steps:\*\***

1. Assign a task to a team member.
2. Add a comment or update to the task.
3. Verify that the assigned team member receives a notification.

**\*\*Postconditions:\*\***

- The notification is delivered to the assigned team member.

**\*\*Expected Result:\*\* The assigned team member receives a notification about the comment or update made on the task.**

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC\_005**

**\*\*Test Case Title:\*\* Verify that the system provides a clear view of task collaboration history.**

**\*\*Test Case Description:\*\* This test case ensures that the system provides a chronological view of all comments, updates, and task assignments, allowing users to track the collaboration history.**

**\*\*Test Priority:\*\* High**

**\*\*Test Steps:\*\***

1. Assign a task to a team member.
2. Add comments and updates to the task.

3. View the task details page.

**\*\*Postconditions:\*\***

- The system displays a clear chronological list of all updates and comments.
- All task assignment changes are recorded and visible.

**\*\*Expected Result:\*\*** The system provides a clear view of all comments, updates, and assignments made on the task in chronological order, allowing users to track the collaboration history.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 44**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Tracking

### **Feature description -**

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

### **Userstory title -**

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

### **Acceptance criteria -**

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.',

'System provides visual indicators for task progress.', "]

## Generated Test Cases:

### ## Test Cases for Task Management Tool - Task Tracking Feature

#### \*\*Project Context:\*\*

\* Project Name: Task Management Tool

\* Project Description: 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.

\* User Story 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:\*\*

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 Creation Template:\*\*

## **\*\*Test Case 1\*\***

**\* \*\*Test Case ID:\*\* TC-TT-01**

**\* \*\*Test Case Title:\*\*** Verify that users can view task statuses.

**\* \*\*Test Case Description:\*\*** This test case verifies that users can view the current status of a task, including "To Do", "In Progress", and "Completed".

**\* \*\*Test Suite:\*\*** Task Tracking

**\* \*\*Test Priority:\*\*** High

**\* \*\*Preconditions:\*\***

- \* User is logged in.

- \* A task has been created.

**\* \*\*Test Data:\*\*** No test data needed.

**\* \*\*Test Steps:\*\***

1. Navigate to the task list.
2. Select a task.
3. Observe the task's status displayed.

**\* \*\*Postconditions:\*\*** None.

**\* \*\*Expected Result:\*\*** The task status is displayed accurately (e.g., "To Do", "In Progress", "Completed").

**\* \*\*Severity:\*\*** Major

**\* \*\*Type of Testing:\*\*** Functional

**\* \*\*Test Case Approach:\*\*** Positive

## **\*\*Test Case 2\*\***

**\* \*\*Test Case ID:\*\*** TC-TT-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 made on a task and monitor

its deadline.

\* \*\*Test Suite:\*\* Task Tracking

\* \*\*Test Priority:\*\* High

\* \*\*Preconditions:\*\*

- \* User is logged in.

- \* A task has been created with a deadline.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. Navigate to the task list.
2. Select a task.
3. Observe the task's progress bar or percentage completion.
4. Verify the deadline is displayed for the task.

\* \*\*Postconditions:\*\* None.

\* \*\*Expected Result:\*\* The task progress is displayed accurately and the deadline is visible.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Functional

\* \*\*Test Case Approach:\*\* Positive

**\*\*Test Case 3\*\***

\* \*\*Test Case ID:\*\* TC-TT-03

\* \*\*Test Case Title:\*\* Verify that tasks are updated in real-time.

\* \*\*Test Case Description:\*\* This test case verifies that changes to task status or other attributes are reflected in the task view in real-time for all users viewing the task.

\* \*\*Test Suite:\*\* Task Tracking

\* \*\*Test Priority:\*\* High

\* \*\*Preconditions:\*\*

- \* Two users are logged in and viewing the same task.

- \* The task is in "To Do" status.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. User 1 changes the task status to "In Progress".

2. User 2 observes the task view.

3. User 2 verifies the task status has changed to "In Progress".

\* \*\*Postconditions:\*\* None.

\* \*\*Expected Result:\*\* The task status updates in real-time for User 2, reflecting the change made by User 1.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Integration

\* \*\*Test Case Approach:\*\* Positive

\*\*Test Case 4\*\*

\* \*\*Test Case ID:\*\* TC-TT-04

\* \*\*Test Case Title:\*\* Verify that the system provides visual indicators for task progress.

\* \*\*Test Case Description:\*\* This test case verifies that the system uses visual cues, such as progress bars, colors, or icons, to clearly indicate the progress of tasks.

\* \*\*Test Suite:\*\* Task Tracking

\* \*\*Test Priority:\*\* High

\* \*\*Preconditions:\*\*

- \* User is logged in.

- \* A task is in progress.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. Navigate to the task list.

2. Observe the task view for any visual indicators of progress (e.g., progress bar, color change, icon).

\* \*\*Postconditions:\*\* None.

\* \*\*Expected Result:\*\* The system uses visual indicators to clearly display the task progress.

\* \*\*Severity:\*\* Major

\* \*\*Type of Testing:\*\* Usability

\* \*\*Test Case Approach:\*\* Positive

\*\*Test Case 5\*\*

\* \*\*Test Case ID:\*\* TC-TT-05

\* \*\*Test Case Title:\*\* Verify that task progress updates when a task is marked as completed.

\* \*\*Test Case Description:\*\* This test case verifies that a task's progress indicator changes to reflect completion when the task is marked as "Completed".

\* \*\*Test Suite:\*\* Task Tracking

\* \*\*Test Priority:\*\* Medium

\* \*\*Preconditions:\*\*

\* User is logged in.

\* A task is in progress.

\* \*\*Test Data:\*\* No test data needed.

\* \*\*Test Steps:\*\*

1. Navigate to the task list.
2. Select the task that is in progress.
3. Mark the task as "Completed".
4. Observe the task's progress indicator.

\* \*\*Postconditions:\*\* None.

\* \*\*Expected Result:\*\* The task's progress indicator changes to reflect completion (e.g., a filled progress bar, green color, checkmark icon).



\* \*\*Severity:\*\* Minor

\* \*\*Type of Testing:\*\* Functional

\* \*\*Test Case Approach:\*\* Positive

These test cases cover all the acceptance criteria for the "Task Tracking" feature and provide a solid foundation for verifying the functionality and usability of the system. By executing these test cases, the development team can ensure that the task tracking feature meets the user's needs and expectations.

-----

**Data - 45**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Notifications

**Feature description -**

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

**Userstory title -**

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

**Acceptance criteria -**

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.',

'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', "]

## Generated Test Cases:

### ## Test Cases for Task Management Tool - Task Notifications

**\*\*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.

**\*\*User Story 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.

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* TN-01**

**\*\*Test Case Title:\*\*** Verify that users receive notifications for new tasks

**\*\*Test Case Description:\*\*** This test case verifies that users are notified when a new task is assigned to them.

**\*\*Test Suite:\*\* Task Notifications**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- User is assigned a new task

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Create a new task and assign it to the user.
2. Check the user's notification center.

**\*\*Postconditions:\*\***

- The notification should be displayed in the notification center.

**\*\*Expected Result:\*\*** The system displays a notification indicating a new task assignment.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** 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 assigned to is updated or commented on.

**\*\*Test Suite:\*\*** Task Notifications

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- User is assigned to a task

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Update the assigned task by adding a description or changing the priority.
2. Comment on the assigned task.
3. Check the user's notification center.

**\*\*Postconditions:\*\***

- The notification should be displayed in the notification center.

**\*\*Expected Result:\*\*** The system displays notifications indicating updates and comments on the assigned task.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** 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 upcoming task deadlines.

**\*\*Test Suite:\*\*** Task Notifications

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- User is assigned to a task with a deadline

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Set a deadline for the assigned task.
2. Wait until the deadline is approaching (e.g., 24 hours, 1 hour before deadline).
3. Check the user's notification center.

**\*\*Postconditions:\*\***

- The notification should be displayed in the notification center.

**\*\*Expected Result:\*\*** The system displays a reminder notification about the approaching deadline.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** 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 preferences based on their needs.

**\*\*Test Suite:\*\*** Task Notifications

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access the notification settings page.
2. Change notification settings for various events (e.g., turn off new task notifications, adjust reminder frequency).
3. Create a new task or update an existing one.
4. Check the user's notification center.

**\*\*Postconditions:\*\***

- Notifications should be displayed based on the customized settings.

**\*\*Expected Result:\*\*** The system displays notifications according to the user's configured preferences.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TN-05**

**\*\*Test Case Title:\*\* Verify that users can disable notifications for specific tasks**

**\*\*Test Case Description:\*\* This test case verifies that users can mute notifications for individual tasks if they don't want to be alerted for those specific tasks.**

**\*\*Test Suite:\*\* Task Notifications**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in
- User is assigned to a task

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the assigned task details.
2. Disable notifications for this specific task.
3. Update the task or comment on it.
4. Check the user's notification center.

**\*\*Postconditions:\*\***

- No notification for the specific task should be displayed.

**\*\*Expected Result:\*\* The system does not display notifications for the task that has been muted.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*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 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.

**\*\*User Story 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.,

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* TC-CAT-01**

**\*\*Test Case Title:\*\*** Verify that users can create and assign tasks to different categories or projects

**\*\*Test Case Description:\*\*** This test case verifies the functionality of creating new categories/projects and assigning existing tasks to them.

**\*\*Test Suite:\*\* Task Categorization**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.

**\*\*Test Data:\*\***

- Category Name: "Project A"

- Category Name: "Project B"

- Task Name: "Task 1"

- Task Name: "Task 2"



- Task Name: "Task 3"

**\*\*Test Steps:\*\***

1. Navigate to the task list view.
2. Create a new category/project named "Project A".
3. Create another category/project named "Project B".
4. Create a task named "Task 1" and assign it to "Project A".
5. Create another task named "Task 2" and assign it to "Project A".
6. Create a task named "Task 3" and assign it to "Project B".

**\*\*Postconditions:\*\***

- Three tasks are created and assigned to the respective categories/projects.
- Task list displays the new categories/projects.

**\*\*Expected Result:\*\*** The system allows creating new categories/projects and assigning tasks to them successfully.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC-CAT-02

**\*\*Test Case Title:\*\*** Verify that users can view tasks grouped by categories or projects

**\*\*Test Case Description:\*\*** This test case verifies that the system displays tasks grouped by their assigned category/project.

**\*\*Test Suite:\*\*** Task Categorization

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.

- Tasks are assigned to different categories/projects (as per TC-CAT-01).

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list view.
2. Select the "Project A" category/project from the filter or navigation.
3. Verify that only tasks assigned to "Project A" are displayed.
4. Select the "Project B" category/project from the filter or navigation.
5. Verify that only tasks assigned to "Project B" are displayed.

**\*\*Postconditions:\*\***

- The system displays task lists based on the selected category/project.

**\*\*Expected Result:\*\*** The system displays tasks grouped by their assigned categories/projects accurately.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC-CAT-03

**\*\*Test Case Title:\*\*** Verify that tasks can be moved between categories or projects

**\*\*Test Case Description:\*\*** This test case checks the ability to move tasks from one category/project to another.

**\*\*Test Suite:\*\*** Task Categorization

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.

- Tasks are assigned to different categories/projects (as per TC-CAT-01).

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list view.
2. Select the "Project A" category/project.
3. Select the task named "Task 1" and move it to "Project B".
4. Refresh the task list view.
5. Verify that "Task 1" is no longer displayed in the "Project A" list.
6. Verify that "Task 1" is displayed in the "Project B" list.

**\*\*Postconditions:\*\***

- The task is moved successfully to a different category/project.

**\*\*Expected Result:\*\*** The system allows moving tasks between categories/projects, updating the task view accordingly.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC-CAT-04

**\*\*Test Case Title:\*\*** Verify that category or project information is accurately reflected in task views

**\*\*Test Case Description:\*\*** This test case checks if the task view displays the assigned category/project information correctly.

**\*\*Test Suite:\*\*** Task Categorization

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.
- Tasks are assigned to different categories/projects (as per TC-CAT-01).

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list view.
2. Select a task assigned to "Project A".
3. Verify that the task view displays "Project A" as its assigned category/project.
4. Select a task assigned to "Project B".
5. Verify that the task view displays "Project B" as its assigned category/project.

**\*\*Postconditions:\*\***

- Category/project information is displayed correctly in the task view.

**\*\*Expected Result:\*\*** The system accurately displays the assigned category/project for each task.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC-CAT-05

**\*\*Test Case Title:\*\*** Verify that creating a category/project with special characters is prevented

**\*\*Test Case Description:\*\*** This test case checks if the system prevents users from creating categories/projects with invalid characters.

**\*\*Test Suite:\*\*** Task Categorization

**\*\*Test Priority:\*\*** Low

**\*\*Preconditions:\*\***

- User is logged in.

**\*\*Test Data:\*\***

- Category Name: "Project !@#\$\$%"

**\*\*Test Steps:\*\***

1. Navigate to the task list view.

2. Attempt to create a new category/project named "Project !@#%\$".
3. Observe if the system allows creating a category/project with special characters.

**\*\*Postconditions:\*\***

- The system should not allow creation of category/project with special characters.

**\*\*Expected Result:\*\*** The system prevents the creation of categories/projects with invalid characters, displaying an error message or preventing the submission.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Negative

-----

## **Data - 47**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Search

### **Feature description -**

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

### **Userstory title -**

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

## Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

## Generated Test Cases:

## Test Cases for Task Management Tool - Task Search Feature

**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.

**Acceptance Criteria:**

1. Users can search for tasks by name.
2. Users can search for tasks by description.
3. Users can filter search results by category, priority, or due date.
4. Search results are accurate and relevant.

**Test Case Creation Template:**

**Test Case ID:** TC\_TS\_01

**Test Case Title:** Verify that users can search for tasks by name.

**Test Case Description:** This test case aims to validate that the system allows users to search for tasks using the task name as a search criterion.

**Test Suite:** Task Search

**Test Priority:** High

**\*\*Preconditions:\*\***

- User is logged in.
- At least one task with a unique name exists.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list.
2. Enter a task name in the search bar.
3. Click the search button.

**\*\*Postconditions:\*\***

- The search results should display only tasks that match the entered task name.

**\*\*Expected Result:\*\*** The system should return a list of tasks matching the provided task name.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_TS\_02

**\*\*Test Case Title:\*\*** Verify that users can search for tasks by description.

**\*\*Test Case Description:\*\*** This test case validates that the system allows users to search for tasks using the task description as a search criterion.

**\*\*Test Suite:\*\*** Task Search

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- At least one task with a unique description exists.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list.

2. Enter a task description in the search bar.

3. Click the search button.

**\*\*Postconditions:\*\***

- The search results should display only tasks that match the entered task description.

**\*\*Expected Result:\*\*** The system should return a list of tasks matching the provided task description.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_TS\_03

**\*\*Test Case Title:\*\*** Verify that users can filter search results by category, priority, or due date.

**\*\*Test Case Description:\*\*** This test case validates that the system allows users to filter search results based on task category, priority level, or due date.

**\*\*Test Suite:\*\*** Task Search

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.

- Multiple tasks with various categories, priorities, and due dates exist.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the task list.

2. Enter a general keyword in the search bar (e.g., "Project A").

3. Select a category filter.

4. Select a priority filter.

5. Select a due date filter.

6. Click the search button.

**\*\*Postconditions:\*\*** The search results should display only tasks that match the selected filter criteria.



**\*\*Expected Result:\*\*** The system should filter search results based on the selected category, priority, and due date filters.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_TS\_04

**\*\*Test Case Title:\*\*** Verify that search results are accurate and relevant.

**\*\*Test Case Description:\*\*** This test case validates that the system returns accurate and relevant search results for a specific search term, ensuring no irrelevant or missing tasks are included.

**\*\*Test Suite:\*\*** Task Search

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- Multiple tasks with various names, descriptions, and categories exist.

**\*\*Test Data:\*\*** A unique search term related to task names, descriptions, or categories.

**\*\*Test Steps:\*\***

1. Navigate to the task list.
2. Enter a specific search term in the search bar (e.g., "Meeting").
3. Click the search button.

**\*\*Postconditions:\*\***

- All tasks matching the search term should be displayed in the results.
- No irrelevant tasks should be included.

**\*\*Expected Result:\*\*** The system should accurately display all relevant tasks matching the search term, without any irrelevant results.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_TS\_05**

**\*\*Test Case Title:\*\* Verify that the search function handles invalid or incomplete search terms correctly.**

**\*\*Test Case Description:\*\* This test case validates that the system handles invalid or incomplete search terms correctly, preventing errors and providing appropriate feedback to the user.**

**\*\*Test Suite:\*\* Task Search**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the task list.
2. Enter an incomplete search term (e.g., "me").
3. Click the search button.
4. Enter invalid search characters (e.g., "@@%\$").
5. Click the search button.

**\*\*Postconditions:\*\***

- The system should display appropriate feedback messages for invalid or incomplete search terms.
- The system should not throw any errors.

**\*\*Expected Result:\*\* The system should handle invalid or incomplete search terms without errors, providing relevant feedback to the user.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Negative**

-----

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Collaboration

**Feature description -**

Users can collaborate on tasks by adding comments, attaching files, and sharing tasks with other users. Task collaboration aims to enhance team communication and productivity.

**Userstory title -**

As a user, I want to collaborate on tasks by adding comments, attaching files, and sharing tasks with other users so that I can work more effectively with my team.

**Acceptance criteria -**

['Users can add comments to tasks.', 'Users can attach files to tasks.', 'Users can share tasks with other users.', 'Task collaboration features work seamlessly and enhance team communication.', '']

**Generated Test Cases:**

## Test Cases for Task Management Tool - Task Collaboration

**\*\*Project Name:\*\*** Task Management Tool

**\*\*Feature Name:\*\*** Task Collaboration

**\*\*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:\*\*** Task Collaboration

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\***

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\*** TC\_01\_TaskCollaboration

**\*\*Test Case Title:\*\*** Verify that users can add comments to tasks.

**\*\*Test Case Description:\*\*** This test case validates the functionality of adding comments to tasks, ensuring that users can leave messages on tasks, and that the comments are properly displayed and associated with the respective task.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.

- A task is created.

**\*\*Test Data:\*\***

- Comment text: "This is a test comment."

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Click on the comment section.
3. Enter the test comment.
4. Click on the "Post comment" button.

**\*\*Postconditions:\*\***

- The comment is successfully posted and displayed under the task.

**\*\*Expected Result:\*\*** The comment is added to the task, and the user can see the comment under the task.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** TC\_02\_TaskCollaboration

**\*\*Test Case Title:\*\*** Verify that users can attach files to tasks.

**\*\*Test Case Description:\*\*** This test case checks the ability to attach files to tasks, ensuring that users can upload files, view attached files, and download them as needed.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- A task is created.

**\*\*Test Data:\*\***

- A test file (e.g., .pdf, .txt, .jpg)

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Click on the "Attach File" button.
3. Select the test file.
4. Click on the "Upload" button.

**\*\*Postconditions:\*\***

- The file is successfully attached to the task.

**\*\*Expected Result:\*\*** The file is uploaded and displayed as an attachment to the task. The user can also download the attached file.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TC\_03\_TaskCollaboration

**\*\*Test Case Title:\*\*** Verify that users can share tasks with other users.

**\*\*Test Case Description:\*\*** This test case verifies that users can share tasks with other users within the system, granting them access to view and collaborate on the task.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- A task is created.
- Another user is registered and logged in.

**\*\*Test Data:\*\***

- Username of the other user.

**\*\*Test Steps:\*\***

1. Navigate to the task.
2. Click on the "Share Task" button.
3. Enter the username of the other user.
4. Click on the "Share" button.

**\*\*Postconditions:\*\***

- The task is successfully shared with the other user.

**\*\*Expected Result:\*\*** The other user receives a notification that they have been granted access to the task.

They can then view and interact with the shared task.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TC\_04\_TaskCollaboration

**\*\*Test Case Title:\*\*** Verify that task collaboration features work seamlessly and enhance team communication.

**\*\*Test Case Description:\*\*** This test case aims to verify that all the collaboration features (commenting, file attachment, sharing) work together seamlessly to facilitate effective team communication and collaboration.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Two users are logged in.
- A task is created with one user as the owner.

**\*\*Test Data:\*\***

- Comment text.
- Test file.

**\*\*Test Steps:\*\***

1. User 1 creates a task and shares it with User 2.
2. User 2 adds a comment to the task.
3. User 1 attaches a file to the task.
4. User 2 downloads the file and adds another comment.

**\*\*Postconditions:\*\***

- Both users can see the added comments and the attached file.

**\*\*Expected Result:\*\*** All actions are executed successfully without issues. Users are notified of updates via notifications or by viewing the task.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** TC\_05\_TaskCollaboration

**\*\*Test Case Title:\*\*** Verify that task sharing respects permissions.

**\*\*Test Case Description:\*\*** This test case verifies that when a task is shared with another user, the shared user only has the permissions specified by the owner of the task.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.
- A task is created.
- Another user is registered and logged in.

**\*\*Test Data:\*\***

- Username of the other user.

**\*\*Test Steps:\*\***

1. Navigate to the task.



2. Click on the "Share Task" button.
3. Select "Read-Only" access for the other user.
4. Enter the username of the other user.
5. Click on the "Share" button.
6. The other user logs in and tries to edit the task.

**\*\*Postconditions:\*\***

- The other user is unable to edit the task.

**\*\*Expected Result:\*\*** The other user should only be able to view and comment on the task, not edit it.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Security

**\*\*Test Case Approach:\*\*** Negative

-----

## **Data - 49**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Course Creation Tools

### **Feature description -**

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to structure content effectively, ensuring an engaging and informative learning experience for students.

## **Userstory title -**

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

## **Acceptance criteria -**

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

## **Generated Test Cases:**

## Test Cases for Online Learning Platform - Course Creation Tools

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\*** OLPT-CC-01

**\*\*Test Case Title:\*\*** Verify that instructors can create courses using videos, presentations, quizzes, and assignments.

**\*\*Test Case Description:\*\*** This test case validates that instructors have the functionality to create courses using various media elements, including videos, presentations, quizzes, and assignments.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is created and logged in

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the Course Creation section.
2. Select "Create New Course."
3. Add a course title and description.
4. Upload a video to the course content.

5. Add a presentation file to the course content.
6. Create a quiz with multiple-choice questions.
7. Create an assignment with a specific deadline and description.

**\*\*Postconditions:\*\***

- The created course should include uploaded video, presentation, quiz, and assignment.

**\*\*Expected Result:\*\*** The system successfully allows instructors to create courses using videos, presentations, quizzes, and assignments.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** OLPT-CC-02

**\*\*Test Case Title:\*\*** Verify that course creation tools are user-friendly and require minimal technical skills.

**\*\*Test Case Description:\*\*** This test case assesses the usability of course creation tools by a user with minimal technical skills. The goal is to ensure that the tools are intuitive and easy to navigate.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is created and logged in

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the Course Creation section.
2. Select "Create New Course."
3. Attempt to create a course using the tools provided (add a title, description, upload content, create a quiz, etc.).

4. Observe the ease of use and intuitive nature of the tools.

5. Check if help documentation or tutorials are available and easily accessible.

**\*\*Postconditions:\*\***

- The instructor successfully creates a basic course.

**\*\*Expected Result:\*\*** The course creation tools are intuitive and user-friendly, requiring minimal technical skills.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Usability

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** OLPT-CC-03

**\*\*Test Case Title:\*\*** Verify that courses can be previewed before publishing.

**\*\*Test Case Description:\*\*** This test case validates that instructors can preview their courses before publishing them to students. This allows them to check content and functionality before making the course live.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Instructor account is created and logged in

- A course is partially created (with at least one content item added)

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the created course.

2. Click on the "Preview Course" button or equivalent option.

3. Observe the course content and functionality from a student's perspective.

4. Check the navigation and multimedia elements (videos, presentations, etc.).

5. Return to the course editing mode.

**\*\*Postconditions:\*\***

- The instructor returns to the course editing mode without any changes to the course content.

**\*\*Expected Result:\*\*** The system successfully allows instructors to preview their courses before publishing them, providing a student-like experience.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** OLPT-CC-04

**\*\*Test Case Title:\*\*** Verify that instructors can save courses as drafts.

**\*\*Test Case Description:\*\*** This test case checks the functionality of saving a course as a draft. This allows instructors to work on a course over time and save progress without publishing it immediately.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Instructor account is created and logged in

- A course is partially created

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the created course.

2. Click on the "Save Draft" button or equivalent option.

3. Exit the course editing mode.

4. Log out and log back in as the same instructor.

5. Navigate to the "My Courses" section.

6. Verify that the saved draft course is listed.

**\*\*Postconditions:\*\***

- The draft course is saved and accessible for further editing.

**\*\*Expected Result:\*\*** The system successfully allows instructors to save their courses as drafts for later editing.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** OLPT-CC-05

**\*\*Test Case Title:\*\*** Verify that courses can be edited post-publishing.

**\*\*Test Case Description:\*\*** This test case tests the functionality of editing a course after it has been published. This feature allows instructors to make updates, add new content, or fix errors even after the course is available to students.

**\*\*Test Suite:\*\*** Course Creation Tools

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Instructor account is created and logged in

- A course is published

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the published course.

2. Click on the "Edit Course" button or equivalent option.

3. Make changes to the course content (add new content, modify existing content, etc.).

4. Save the changes.

5. Observe the updated course content and verify the changes are reflected.

**\*\*Postconditions:\*\***

- The published course content is updated with the applied changes.

**\*\*Expected Result:\*\*** The system successfully allows instructors to edit published courses, enabling updates and modifications.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*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 Online Learning Platform - Interactive Lessons

**Project Name:** Online Learning Platform

**Feature Name:** Interactive Lessons

**User Story Title:** As a learner, I want interactive lessons so that I can engage with the content more effectively.

**Test Case Creation Template:**

**Test Case ID:**

**Test Case Title:**

**Test Case Description:**

**Test Suite:** Interactive Lessons

**Test Priority:**

**Preconditions:**

**Test Data:**

**Test Steps:**

**Postconditions:**

**Expected Result:**

**Severity:**

**Type of Testing:**



**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TL-01

**\*\*Test Case Title:\*\*** Verify that lessons include quizzes, polls, simulations, and discussions.

**\*\*Test Case Description:\*\*** This test case verifies the presence of all interactive elements within a lesson.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- A lesson with interactive elements is available

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access a lesson with interactive elements.
2. Check for the presence of quizzes, polls, simulations, and discussions within the lesson content.
3. Verify that each interactive element is accessible and functional.

**\*\*Postconditions:\*\***

- User is logged in

**\*\*Expected Result:\*\*** All interactive elements (quizzes, polls, simulations, and discussions) are present and accessible within the selected lesson.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TL-02

**\*\*Test Case Title:\*\*** Verify that interactive elements are integrated seamlessly into the lesson content.

**\*\*Test Case Description:\*\*** This test case checks the smooth flow and integration of interactive elements within the lesson structure.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- A lesson with interactive elements is available

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access a lesson with interactive elements.
2. Navigate through the lesson content, focusing on transitions between interactive elements and regular content.
3. Check for any disruptions or inconsistencies in the flow of the lesson due to interactive elements.

**\*\*Postconditions:\*\***

- User is logged in

**\*\*Expected Result:\*\*** Interactive elements seamlessly integrate with the lesson content, without disrupting the flow or causing confusion.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Usability

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TL-03

**\*\*Test Case Title:\*\*** Verify that learners can track their progress through interactive activities.

**\*\*Test Case Description:\*\*** This test case ensures that the platform provides progress tracking for interactive elements within lessons.

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in
- A lesson with interactive elements is available

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access a lesson with interactive elements.
2. Complete an interactive element (e.g., a quiz, poll, or simulation).
3. Check if the progress of completing this element is reflected in the lesson progress tracker or a dedicated activity tracker.

**\*\*Postconditions:\*\***

- User is logged in

**\*\*Expected Result:\*\* Completion of interactive elements is reflected in the progress tracker, providing learners with a clear view of their progress within the lesson.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TL-04**

**\*\*Test Case Title:\*\* Verify that feedback is provided after each interactive component.**

**\*\*Test Case Description:\*\* This test case checks for the presence and clarity of feedback after completing interactive elements.**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in

- A lesson with interactive elements is available

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access a lesson with interactive elements.
2. Complete an interactive element (e.g., a quiz, poll, or simulation).
3. Observe if feedback is provided after completing the interactive element.
4. Verify that the feedback is clear, relevant, and informative.

**\*\*Postconditions:\*\***

- User is logged in

**\*\*Expected Result:\*\*** Clear, informative feedback is provided after each interactive element, indicating the learner's performance and offering relevant insights.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TL-05

**\*\*Test Case Title:\*\*** Verify that interactive lessons can be accessed on various devices.

**\*\*Test Case Description:\*\*** This test case checks the compatibility and functionality of interactive lessons across different devices.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- A lesson with interactive elements is available

**\*\*Test Data:\*\***

- List of devices (e.g., desktop, laptop, smartphone, tablet)
- Different web browsers (e.g., Chrome, Firefox, Safari, Edge)

**\*\*Test Steps:\*\***

1. Access the lesson with interactive elements on different devices and web browsers.
2. Check for any issues with loading, rendering, or functionality of interactive elements on each device.
3. Verify that all interactive elements function correctly and seamlessly across different devices and browsers.

**\*\*Postconditions:\*\***

- User is logged in

**\*\*Expected Result:\*\*** Interactive lessons are accessible, responsive, and function correctly across different devices and web browsers without any rendering or functional issues.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Compatibility

**\*\*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

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\*** PLP-TC-01

**\*\*Test Case Title:\*\*** Verify that learners can choose courses and modules based on their interests.

**\*\*Test Case Description:\*\*** This test case verifies that learners can browse and select courses and modules based on their interests. It ensures that the platform offers a clear and intuitive interface for filtering courses and modules based on various criteria such as subject, topic, level, and skill.

**\*\*Test Suite:\*\*** Personalized Learning Paths

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Learner is logged in.
- A set of courses and modules are available.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the 'Courses' or 'Modules' section.
2. Use filters or search functionality to narrow down options based on different interests.
3. Select a course or module that matches the desired interest.
4. Verify that the selected course or module is displayed correctly.

**\*\*Postconditions:\*\***

- The selected course or module is added to the learner's personalized learning path.

**\*\*Expected Result:\*\*** Learners are able to successfully filter and select courses and modules based on their interests, and the selected items are added to their personalized learning paths.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** PLP-TC-02

**\*\*Test Case Title:\*\*** Verify that personalized learning paths adjust dynamically based on learner progress.

**\*\*Test Case Description:\*\*** This test case checks if the personalized learning path dynamically adapts as the learner progresses through courses and modules. It evaluates whether the platform recommends appropriate next steps based on completed courses, quizzes, and assessments.

**\*\*Test Suite:\*\*** Personalized Learning Paths

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Learner is logged in.
- A personalized learning path is created.
- Some courses or modules are completed.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Complete a course or module within the personalized learning path.
2. Navigate back to the personalized learning path.
3. Observe whether the path has been updated to suggest relevant courses or modules based on progress.

**\*\*Postconditions:\*\***

- The personalized learning path is updated.

**\*\*Expected Result:\*\*** The personalized learning path dynamically adjusts based on learner progress, suggesting new courses or modules aligned with the completed content.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** PLP-TC-03

**\*\*Test Case Title:\*\*** Verify that learners receive recommendations for new courses and modules.

**\*\*Test Case Description:\*\*** This test case ensures that the platform provides relevant recommendations for new courses and modules based on learner interests and past activities. It checks whether the recommendations are personalized and align with the learner's learning objectives.

**\*\*Test Suite:\*\*** Personalized Learning Paths

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Learner is logged in.
- A personalized learning path is created.
- Learner has interacted with some courses and modules.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the 'Recommendations' section or a similar area within the platform.



2. Observe the recommended courses and modules.

3. Verify that the recommendations are relevant to the learner's interests and previous learning activity.

**\*\*Postconditions:\*\***

- Recommendations are displayed.

**\*\*Expected Result:\*\*** Learners receive personalized recommendations for courses and modules that align with their interests, learning path, and previous activity on the platform.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** PLP-TC-04

**\*\*Test Case Title:\*\*** Verify that progress tracking is personalized to each learner's path.

**\*\*Test Case Description:\*\*** This test case verifies that progress tracking accurately reflects the learner's individual path. It ensures that the system tracks completed courses, modules, quizzes, and assessments specific to the learner's chosen path.

**\*\*Test Suite:\*\*** Personalized Learning Paths

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Learner is logged in.
- A personalized learning path is created.
- The learner has completed some courses and modules.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the 'Progress' or 'Dashboard' section.
2. Verify that the displayed progress accurately reflects the completed courses and modules within the

learner's personalized path.

**\*\*Postconditions:\*\***

- Progress data is displayed.

**\*\*Expected Result:\*\*** The progress tracking system accurately reflects the learner's completed courses, modules, and activities within their personalized learning path.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** PLP-TC-05

**\*\*Test Case Title:\*\*** Verify that learners can adjust their learning paths at any time.

**\*\*Test Case Description:\*\*** This test case checks that learners have the flexibility to modify their personalized learning path as needed. It ensures that the platform allows learners to add, remove, or reorder courses and modules within their path.

**\*\*Test Suite:\*\*** Personalized Learning Paths

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Learner is logged in.
- A personalized learning path is created.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the learner's personalized learning path.
2. Use the available options to add, remove, or reorder courses and modules within the path.
3. Verify that the changes are reflected in the updated path.

**\*\*Postconditions:\*\***

- The personalized learning path is updated.

**\*\*Expected Result:\*\*** Learners can easily adjust their learning paths by adding, removing, or reordering courses and modules at any time.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 52**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Progress Tracking

### **Feature description -**

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

### **Userstory title -**

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

### **Acceptance criteria -**

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their

progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

## **Generated Test Cases:**

### **## Test Cases for Online Learning Platform - Progress Tracking**

**\*\*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.

**\*\*User Story Title:\*\*** As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

**\*\*Acceptance Criteria:\*\***

1. Progress tracking shows completed activities and grades.
2. Learners can view detailed reports on their progress.
3. Achievements are highlighted in the progress tracker.
4. Learners receive notifications about their progress.
5. Progress tracking can be exported as reports.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Progress Tracking

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_PT\_01

**\*\*Test Case Title:\*\*** Verify that progress tracking shows completed activities and grades.

**\*\*Test Case Description:\*\*** This test case verifies that the system displays a list of completed activities and corresponding grades within the progress tracking section.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Learner is logged in.
- Learner has completed at least one activity within a course.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Progress" section of the platform.
2. Select the relevant course.
3. Observe the displayed progress tracker.

4. Verify the presence of completed activities and their associated grades.

**\*\*Postconditions:\*\*** No follow-up actions needed.

**\*\*Expected Result:\*\*** The system should display a list of completed activities and their corresponding grades within the progress tracker for the selected course.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_PT\_02

**\*\*Test Case Title:\*\*** Verify that learners can view detailed reports on their progress.

**\*\*Test Case Description:\*\*** This test case verifies that learners can access and view detailed reports on their overall course progress, including completed activities, grades, and time spent on each module.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Learner is logged in.
- Learner has completed at least one activity within a course.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Progress" section of the platform.
2. Select the relevant course.
3. Look for an option to view "Detailed Report" or similar.
4. Click on the "Detailed Report" option.
5. Verify the presence of detailed information on course progress, including completed activities, grades, and time spent on each module.

**\*\*Postconditions:\*\*** No follow-up actions needed.

**\*\*Expected Result:\*\*** The system should provide a detailed report on the learner's progress in the selected

course.

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC\_PT\_03**

**\*\*Test Case Title:\*\* Verify that achievements are highlighted in the progress tracker.**

**\*\*Test Case Description:\*\* This test case verifies that achievements unlocked by the learner (e.g., completing a specific module or achieving a certain score) are highlighted or displayed prominently within the progress tracker.**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- Learner is logged in.
- Learner has achieved at least one milestone within a course.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the "Progress" section of the platform.
2. Select the relevant course.
3. Observe the progress tracker for any highlighted or displayed achievements.
4. Verify that achievements are visually distinguishable and correspond to the learner's progress.

**\*\*Postconditions:\*\* No follow-up actions needed.**

**\*\*Expected Result:\*\* The progress tracker should visually highlight any achievements earned by the learner within the selected course.**

**\*\*Severity:\*\* Minor**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_PT\_04**

**\*\*Test Case Title:\*\* Verify that learners receive notifications about their progress.**

**\*\*Test Case Description:\*\* This test case verifies that learners receive timely notifications about their progress, such as completing a module, achieving a certain score, or approaching a deadline.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Learner is logged in.
- Learner has configured notification preferences.
- Learner has made progress in a course that triggers a notification.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Complete an activity within a course.
2. Monitor notification settings and in-platform messages.
3. Verify the presence of a notification about the completed activity or achieved milestone.

**\*\*Postconditions:\*\* No follow-up actions needed.**

**\*\*Expected Result:\*\* The system should notify the learner about significant progress made within the course, according to their notification preferences.**

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC\_PT\_05**

**\*\*Test Case Title:\*\* Verify that progress tracking can be exported as reports.**

**\*\*Test Case Description:\*\* This test case verifies that learners can export their progress data as reports in**



various formats (e.g., PDF, CSV) to save or share their achievements.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Learner is logged in.
- Learner has completed at least one activity within a course.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Progress" section of the platform.
2. Select the relevant course.
3. Look for an option to "Export Report" or similar.
4. Click on the "Export Report" option.
5. Choose a desired format (e.g., PDF, CSV).
6. Verify that the report is successfully downloaded and contains accurate progress data.

**\*\*Postconditions:\*\*** No follow-up actions needed.

**\*\*Expected Result:\*\*** The system should allow learners to export their progress data as reports in various formats.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 53**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Assessment and Grading

**Feature description -**

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

**Userstory title -**

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

**Acceptance criteria -**

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

**Generated Test Cases:**

## Test Cases for Online Learning Platform - Assessment and Grading

**Project Name:** Online Learning Platform

**Feature Name:** Assessment and Grading

**User Story Title:** As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

**Test Case 1:**

**Test Case ID:** TC\_001

**\*\*Test Case Title:\*\*** Verify that instructors can create quizzes

**\*\*Test Case Description:\*\*** This test case ensures that instructors have the functionality to create quizzes within the online learning platform.

**\*\*Test Suite:\*\*** Assessment Creation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is logged in
- A course has been created

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the course management section.
2. Select the "Assessments" tab.
3. Choose "Create Quiz".
4. Provide a title and description for the quiz.
5. Save the quiz.

**\*\*Postconditions:\*\***

- The newly created quiz is visible in the list of assessments.

**\*\*Expected Result:\*\*** A new quiz with the specified title and description is successfully created and saved.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_002

**\*\*Test Case Title:\*\*** Verify that assessments can include various question types

**\*\*Test Case Description:\*\*** This test case verifies that instructors can add multiple question types to a single

assessment.

**\*\*Test Suite:\*\*** Assessment Question Types

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is logged in
- A quiz is created

**\*\*Test Data:\*\***

- Multiple Choice Question (MCQ) with answer options
- Short Answer Question with answer key
- Essay Question

**\*\*Test Steps:\*\***

1. Navigate to the quiz editing section.
2. Add a new MCQ question with multiple answer options.
3. Add a new short answer question with the expected answer.
4. Add a new essay question without any answer key.
5. Save the quiz.

**\*\*Postconditions:\*\***

- The quiz contains all three types of questions.

**\*\*Expected Result:\*\*** All question types are successfully added to the quiz. The MCQ and short answer questions are ready for automatic grading.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_003

**\*\*Test Case Title:\*\*** Verify that automatic grading is available for objective questions

**\*\*Test Case Description:\*\*** This test case validates that the system correctly grades objective questions (e.g., MCQs, short answers) based on the answer key.

**\*\*Test Suite:\*\*** Automatic Grading

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is logged in
- A quiz with objective questions is created
- A learner attempts the quiz

**\*\*Test Data:\*\***

- Learner answers for MCQ and short answer questions

**\*\*Test Steps:\*\***

1. Submit the quiz with correct and incorrect answers for objective questions.
2. Check the grading results for the quiz.

**\*\*Postconditions:\*\***

- The system provides immediate feedback to the learner with a score.

**\*\*Expected Result:\*\*** The objective questions are graded automatically and the score is displayed correctly.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_004

**\*\*Test Case Title:\*\*** Verify that manual grading is available for subjective questions

**\*\*Test Case Description:\*\*** This test case ensures that instructors can manually grade subjective questions (e.g., essays) and provide feedback to learners.

**\*\*Test Suite:\*\*** Manual Grading

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is logged in
- A quiz with subjective questions is created
- A learner attempts the quiz

**\*\*Test Data:\*\***

- Learner answer for an essay question

**\*\*Test Steps:\*\***

1. Navigate to the "Grading" section.
2. Access the submitted essay question.
3. Provide feedback and assign a grade to the answer.
4. Save the grading.

**\*\*Postconditions:\*\***

- The assigned grade and feedback are visible to the learner.

**\*\*Expected Result:\*\*** The system provides a manual grading interface for subjective questions, allowing instructors to provide feedback and assign grades.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_005

**\*\*Test Case Title:\*\*** Verify that feedback is provided to learners after grading

**\*\*Test Case Description:\*\*** This test case verifies that learners receive appropriate feedback after an assessment is graded, regardless of the question type.

**\*\*Test Suite:\*\* Feedback Mechanism**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Learner account is logged in
- A quiz is completed and graded

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the graded quiz.
2. Review the feedback provided for each question.

**\*\*Postconditions:\*\***

- Learners can see their performance and feedback.

**\*\*Expected Result:\*\*** The system provides feedback to the learner, including the correct answer for objective questions and detailed comments for subjective questions.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

-----

**Data - 54**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

## Discussion Forums

### **Feature description -**

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge sharing, enriching the learning experience through diverse perspectives and active participation.

### **Userstory title -**

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

### **Acceptance criteria -**

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all devices.']

### **Generated Test Cases:**

## Test Cases for Online Learning Platform - Discussion Forums

**\*\*Project Context:\*\***

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.

User Story Title: As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.



## Acceptance Criteria:

1. Learners can post questions and responses in forums.
2. Forums support threaded discussions.
3. Instructors can moderate discussions.
4. Learners receive notifications for new posts.
5. Forums are accessible on all devices.

## **\*\*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 ensures that learners can successfully create new posts (questions) and reply to existing posts within the discussion forums.

**\*\*Test Suite:\*\*** Discussion Forums

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in as a learner.
- A discussion forum exists.

**\*\*Test Data:\*\***

- A sample question for the forum.

**\*\*Test Steps:\*\***

1. Navigate to the discussion forum.
2. Click on the "New Post" button.
3. Enter a sample question in the provided text field.
4. Click on the "Post" button.
5. Verify that the question appears in the forum list.
6. Click on the question to view the thread.

7. Enter a sample response in the reply field.
8. Click on the "Reply" button.
9. Verify that the response appears in the thread.

**\*\*Postconditions:\*\***

- The new question and reply are visible in the forum.

**\*\*Expected Result:\*\*** The system successfully creates the post (question) and the reply within the forum thread.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*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 the forum supports threaded discussions by allowing users to reply to existing posts, creating a hierarchical structure of replies.

**\*\*Test Suite:\*\*** Discussion Forums

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in as a learner.
- A discussion thread exists with at least one post.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the discussion forum.
2. Select a thread with at least one existing post.
3. Click on the "Reply" button below an existing post.
4. Enter a reply in the text field.

5. Click on the "Reply" button.

6. Verify that the reply is nested under the original post, creating a threaded structure.

**\*\*Postconditions:\*\***

- The reply is visible under the original post in the thread.

**\*\*Expected Result:\*\*** The system successfully creates a reply nested under the original post, creating a threaded discussion structure.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*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 ensures that instructors can moderate discussion forums by deleting inappropriate posts, editing content, and marking posts as resolved.

**\*\*Test Suite:\*\*** Discussion Forums

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in as an instructor.

- A discussion forum exists with posts.

**\*\*Test Data:\*\***

- An inappropriate or irrelevant post in the forum.

**\*\*Test Steps:\*\***

1. Navigate to the discussion forum.

2. Identify an inappropriate post in the forum.

3. Click on the "Moderation" or "Edit" options for the post.

4. Select "Delete" to remove the post or "Edit" to modify the content.

5. Verify that the post is deleted or the content is edited.

6. Select a post that requires resolution.
7. Click on the "Mark as Resolved" option for the post.
8. Verify that the post is marked as resolved.

**\*\*Postconditions:\*\***

- The inappropriate post is deleted or edited, and the relevant post is marked as resolved.

**\*\*Expected Result:\*\*** The instructor can successfully delete or edit posts, and mark posts as resolved within the forum.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*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 ensures that learners receive notifications whenever a new post is made in a forum they are subscribed to or following.

**\*\*Test Suite:\*\*** Discussion Forums

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in as a learner.
- User is subscribed to a discussion forum.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the subscribed forum.
2. Log out of the platform.
3. Have another user post a new question in the forum.
4. Log back in as the original user.

5. Check the notification center or inbox for a notification about the new post.

6. Verify the content of the notification.

**\*\*Postconditions:\*\***

- The notification is received by the user.

**\*\*Expected Result:\*\*** The system successfully sends a notification to the user about the new post in the subscribed forum.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*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 that discussion forums can be accessed seamlessly across different devices and operating systems, including desktops, tablets, and smartphones.

**\*\*Test Suite:\*\*** Discussion Forums

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.

- Different devices (desktop, tablet, smartphone) are available.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Access the online learning platform on a desktop browser.

2. Navigate to a discussion forum.

3. Verify that all forum features (posting, replying, viewing threads, etc.) work correctly.

4. Repeat steps 1-3 on a tablet and smartphone.

5. Ensure that all functionalities work smoothly across all devices.

**\*\*Postconditions:\*\***

- The user can access the discussion forum and perform all necessary actions on each device.

**\*\*Expected Result:\*\*** The discussion forum is accessible and functional on all devices without any major issues or inconsistencies.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Compatibility

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 55**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Live Classes and Webinars

**Feature description -**

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

**Userstory title -**

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

**Acceptance criteria -**

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

## **Generated Test Cases:**

### **## Test Cases for Online Learning Platform - Live Classes and Webinars**

#### **\*\*Project Context:\*\***

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.

User Story Title: 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:\*\*** Live Classes and Webinars

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\***

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_Live\_Class\_Video\_Conf

**\*\*Test Case Title:\*\*** Verify that live classes and webinars are conducted through video conferencing tools.

**\*\*Test Case Description:\*\*** This test case aims to ensure that the platform utilizes video conferencing tools for conducting live classes and webinars, allowing learners to see and hear the instructor in real-time.

**\*\*Test Suite:\*\*** Live Classes and Webinars

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in as a learner

\* A live class or webinar is scheduled



**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access the live class/webinar session.
2. Verify the presence of a video conferencing tool (e.g., Zoom, Google Meet).
3. Observe the instructor's video and audio feed.

**\*\*Postconditions:\*\***

\* Exit the live session.

**\*\*Expected Result:\*\*** The platform utilizes a video conferencing tool, and the learner can see and hear the instructor in real-time.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_Live\_Class\_QnA

**\*\*Test Case Title:\*\*** Verify that learners can ask questions in real-time during live classes and webinars.

**\*\*Test Case Description:\*\*** This test case assesses the functionality of real-time question asking during live sessions, ensuring learners can engage with the instructor and receive immediate responses.

**\*\*Test Suite:\*\*** Live Classes and Webinars

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in as a learner

\* A live class or webinar is ongoing

**\*\*Test Data:\*\*** A sample question relevant to the session topic

**\*\*Test Steps:\*\***

1. Access the live class/webinar session.

2. Use the platform's Q&A feature to submit a question.
3. Observe if the question is visible to both the instructor and other learners.
4. Wait for a response from the instructor.

**\*\*Postconditions:\*\***

\* Exit the live session.

**\*\*Expected Result:\*\*** The learner's question is displayed in the Q&A section, and the instructor responds in real-time.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_Live\_Class\_Recording

**\*\*Test Case Title:\*\*** Verify that live class sessions are recorded and available for later viewing.

**\*\*Test Case Description:\*\*** This test case validates the recording feature for live sessions, ensuring that learners can access the recording after the session concludes for review or if they missed the live session.

**\*\*Test Suite:\*\*** Live Classes and Webinars

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged in as a learner

\* A live class or webinar has concluded

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the session's details page.
2. Look for a "Recording" or "Replay" option.
3. Access the recorded session.

4. Verify that the recording plays without errors.

**\*\*Postconditions:\*\***

\* Exit the recording.

**\*\*Expected Result:\*\*** The platform provides a recorded version of the live session, accessible for viewing after the session has ended.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_Live\_Class\_Screen\_Share

**\*\*Test Case Title:\*\*** Verify that instructors can share screen and resources during live classes and webinars.

**\*\*Test Case Description:\*\*** This test case tests the instructor's ability to share screen content and resources during live sessions, enhancing the interactive and informative nature of the session.

**\*\*Test Suite:\*\*** Live Classes and Webinars

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged in as an instructor

\* A live class or webinar is ongoing

**\*\*Test Data:\*\*** A presentation or document to share.

**\*\*Test Steps:\*\***

1. Access the live session as the instructor.
2. Use the platform's screen sharing feature.
3. Share a presentation or resource file.
4. Verify that the content is visible to learners.

**\*\*Postconditions:\*\***

\* Stop sharing the screen.

**\*\*Expected Result:\*\*** The instructor can successfully share their screen and relevant resources during the live session, making them visible to learners.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_Live\_Class\_Notification

**\*\*Test Case Title:\*\*** Verify that notifications are sent before live class sessions start.

**\*\*Test Case Description:\*\*** This test case validates the notification system, ensuring that learners receive timely reminders about upcoming live sessions to ensure participation and minimize missed opportunities.

**\*\*Test Suite:\*\*** Live Classes and Webinars

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in as a learner

\* A live class or webinar is scheduled

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Set a time before the scheduled live session start.

2. Verify that the learner receives a notification (e.g., email, in-app notification) about the upcoming live session.

3. Check the notification's content (date, time, session details).

**\*\*Postconditions:\*\***

\* Dismiss the notification.

**\*\*Expected Result:\*\*** The system sends notifications to learners about upcoming live sessions, including

details like date, time, and session title, a sufficient time before the session starts.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 56**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Mobile Accessibility

### **Feature description -**

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

### **Userstory title -**

As a learner, I want mobile access to the platform so that I can learn on the go.

### **Acceptance criteria -**

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

## Generated Test Cases:

### ## Test Cases for Online Learning Platform - Mobile Accessibility

**\*\*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.

**\*\*User Story Title:\*\*** As a learner, I want mobile access to the platform so that I can learn on the go.

**\*\*Acceptance Criteria:\*\***

1. The platform is accessible on smartphones and tablets.
2. Mobile UI is optimized for touch navigation.
3. Course materials are accessible and readable on mobile devices.
4. Notifications are sent to mobile devices.
5. The platform supports offline access to downloaded materials.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC-Mobile-01

**\*\*Test Case Title:\*\*** Verify that the platform is accessible on smartphones and tablets

**\*\*Test Case Description:\*\*** Ensure that the platform's website or app functions correctly and displays content appropriately on different smartphone and tablet devices with varying screen sizes and resolutions.

**\*\*Test Suite:\*\*** Mobile Accessibility

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- A user account is created.
- The user is logged in.

**\*\*Test Data:\*\***

- List of supported smartphone and tablet devices (e.g., iPhone 14, Samsung Galaxy S23, iPad Pro, etc.)
- List of different screen sizes and resolutions

**\*\*Test Steps:\*\***

1. Access the platform using the listed smartphone and tablet devices.
2. Navigate through various sections of the platform, including course listings, course content, user profiles, and settings.
3. Observe the platform's responsiveness, layout, and content display on different devices.

**\*\*Postconditions:\*\***

- User is still logged in.
- Platform functionality remains intact after testing.

**\*\*Expected Result:\*\*** The platform should be accessible and functional on all supported smartphone and tablet devices. The user interface should adapt seamlessly to different screen sizes and resolutions, ensuring content is displayed correctly and navigatable without any issues.

**\*\*Severity:\*\*** Critical

**\*\*Type of Testing:\*\*** Compatibility Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-Mobile-02

**\*\*Test Case Title:\*\*** Verify that the mobile UI is optimized for touch navigation

**\*\*Test Case Description:\*\*** Ensure that the platform's mobile UI is designed for touch interaction, with elements like buttons, menus, and links being easily tappable and responsive.

**\*\*Test Suite:\*\*** Mobile Accessibility

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- A user account is created.
- The user is logged in.
- The platform is accessed on a smartphone or tablet.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Navigate through the platform using touch gestures.
2. Tap on buttons, menu items, links, and other interactive elements.
3. Check the responsiveness and accuracy of touch interactions.

**\*\*Postconditions:\*\***

- User is still logged in.
- Platform functionality remains intact after testing.

**\*\*Expected Result:\*\*** All UI elements should be easily accessible and responsive to touch interactions. Tappable areas should be large enough and clearly defined to avoid accidental clicks.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Usability Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-Mobile-03

**\*\*Test Case Title:\*\*** Verify that course materials are accessible and readable on mobile devices

**\*\*Test Case Description:\*\*** Ensure that course content, including videos, text, images, and downloadable resources, is optimized for display and readability on mobile screens.



**\*\*Test Suite:\*\*** Mobile Accessibility

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- A user account is created.
- The user is logged in.
- The platform is accessed on a smartphone or tablet.

**\*\*Test Data:\*\***

- A selection of courses with different content types.

**\*\*Test Steps:\*\***

1. Access various courses and review their content, including video lectures, text materials, images, and downloadable files.
2. Verify that all content is displayed correctly and is legible on the mobile screen.
3. Check the video player functionality, playback controls, and video quality on mobile devices.

**\*\*Postconditions:\*\***

- User is still logged in.
- Platform functionality remains intact after testing.

**\*\*Expected Result:\*\*** All course materials should be displayed correctly and legibly on mobile devices.

Videos should play smoothly, and other content should be easily accessible and readable.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-Mobile-04

**\*\*Test Case Title:\*\*** Verify that notifications are sent to mobile devices

**\*\*Test Case Description:\*\*** Ensure that the platform sends relevant notifications to the user's mobile device, informing them about new courses, deadlines, updates, and other important information.

**\*\*Test Suite:\*\*** Mobile Accessibility

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- A user account is created.
- The user is logged in.
- The platform is accessed on a smartphone or tablet.
- The user has enabled mobile notifications for the platform.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Simulate an event that triggers a notification, such as enrolling in a new course, submitting an assignment, or receiving a message on a discussion forum.
2. Observe the notification delivery on the user's mobile device.
3. Verify that the notification content is relevant and informative.

**\*\*Postconditions:\*\***

- User is still logged in.
- Platform functionality remains intact after testing.

**\*\*Expected Result:\*\*** The platform should send timely and relevant notifications to the user's mobile device, informing them about important events and updates. Notifications should be clearly displayed and easily accessible on the mobile device.

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC-Mobile-05**

**\*\*Test Case Title:\*\* Verify that the platform supports offline access to downloaded materials**

**\*\*Test Case Description:\*\* Ensure that the platform allows users to download course materials, such as videos, presentations, and documents, for offline access on their mobile devices.**

**\*\*Test Suite:\*\*** Mobile Accessibility

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- A user account is created.
- The user is logged in.
- The platform is accessed on a smartphone or tablet.
- The user has a stable internet connection to download materials.

**\*\*Test Data:\*\***

- A selection of course materials available for download (e.g., video lectures, PDF documents).

**\*\*Test Steps:\*\***

1. Download a selected course material on the platform while connected to the internet.
2. Disconnect from the internet and attempt to access the downloaded material.
3. Verify that the downloaded content is accessible and playable/viewable in offline mode.

**\*\*Postconditions:\*\***

- User is still logged in.
- Platform functionality remains intact after testing.

**\*\*Expected Result:\*\*** Users should be able to download course materials while online and access them offline on their mobile devices. The platform should provide a clear indication of downloaded content and provide a convenient way to access it in offline mode.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 57**

**Project name -**

## Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Resource Library

### **Feature description -**

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

### **Userstory title -**

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

### **Acceptance criteria -**

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

### **Generated Test Cases:**

## Test Cases for Online Learning Platform Resource Library

**\*\*Project Name:\*\*** Online Learning Platform

**\*\*Project Description:\*\*** An online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics.

**\*\*Feature Name:\*\*** Resource Library

**\*\*Feature Description:\*\*** A comprehensive repository of supplementary materials like e-books, articles, and reference guides enriches the learning experience.

**\*\*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:\*\*** Resource Library

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the platform.
- User is enrolled in a course.

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

---

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* TC-RL-01**

**\*\*Test Case Title:\*\* Verify that the Resource Library includes e-books, articles, and reference guides.**

**\*\*Test Case Description:\*\* Ensure that the Resource Library section contains a variety of e-books, articles, and reference guides related to the course content.**

**\*\*Test Data:\*\* List of e-books, articles, and reference guides relevant to the course.**

**\*\*Test Steps:\*\***

1. Navigate to the course page.
2. Access the Resource Library section.
3. Verify the presence of e-books, articles, and reference guides.
4. Check if the content is relevant to the course topics.

**\*\*Postconditions:\*\* None.**

**\*\*Expected Result:\*\* The Resource Library displays a diverse range of e-books, articles, and reference guides relevant to the course curriculum.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

---

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC-RL-02

**\*\*Test Case Title:\*\*** Verify that learners can search for specific materials in the Resource Library.

**\*\*Test Case Description:\*\*** Ensure that learners can effectively search for specific materials within the Resource Library using keywords or filters.

**\*\*Test Data:\*\*** A specific keyword or filter related to the course content.

**\*\*Test Steps:\*\***

1. Navigate to the Resource Library section.
2. Use the search bar or filter options to search for a specific resource.
3. Verify that relevant results are displayed based on the search criteria.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** The search function returns accurate and relevant results matching the learner's search query.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

---

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC-RL-03

**\*\*Test Case Title:\*\*** Verify that resources are categorized and tagged for easy navigation in the Resource Library.

**\*\*Test Case Description:\*\*** Ensure that resources are categorized and tagged appropriately to enable learners to easily find relevant materials.

**\*\*Test Data:\*\*** Different resource categories and tags related to the course content.

**\*\*Test Steps:\*\***

1. Navigate to the Resource Library section.
2. Verify that resources are categorized into relevant groups (e.g., articles, e-books, reference guides).
3. Examine if tags are applied to each resource, making it easier to find materials based on specific topics or themes.
4. Check if filtering or sorting options are available based on categories and tags.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** Resources are organized into distinct categories with appropriate tags, allowing learners to browse and filter based on their needs.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

---

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC-RL-04

**\*\*Test Case Title:\*\*** Verify that learners can download resources for offline access from the Resource Library.

**\*\*Test Case Description:\*\*** Ensure that learners can download selected resources from the Resource Library for offline access.

**\*\*Test Data:\*\*** A resource file (e.g., PDF e-book, article, reference guide) in the Resource Library.

**\*\*Test Steps:\*\***

1. Navigate to the Resource Library section.
2. Select a resource to download.



3. Verify that a download button/option is available.
4. Initiate the download process.
5. Confirm that the resource is successfully downloaded.

**\*\*Postconditions:\*\*** The resource file is downloaded and saved to the learner's device.

**\*\*Expected Result:\*\*** Learners are able to successfully download resources from the Resource Library for offline viewing or access.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

---

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC-RL-05

**\*\*Test Case Title:\*\*** Verify that instructors can recommend resources to learners from the Resource Library.

**\*\*Test Case Description:\*\*** Ensure that instructors have the functionality to recommend specific resources to learners from the Resource Library.

**\*\*Test Data:\*\*** An instructor account and a specific resource from the Resource Library.

**\*\*Test Steps:\*\***

1. Log in as an instructor.
2. Navigate to the course page.
3. Access the Resource Library section.
4. Select a resource to recommend.
5. Verify the presence of a "Recommend" or "Share" button/option.
6. Initiate the recommendation process to learners.
7. Verify that the recommended resource is displayed to learners.

**\*\*Postconditions:\*\*** The recommended resource is visible to the learners.

**\*\*Expected Result:\*\*** Instructors are able to recommend relevant resources from the Resource Library to their learners within the course environment.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 58**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Peer Review

### **Feature description -**

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

### **Userstory title -**

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

### **Acceptance criteria -**

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers

remain anonymous to ensure unbiased feedback.']

## Generated Test Cases:

## Test Cases for Online Learning Platform - Peer Review Feature

**Project Name:** Online Learning Platform

**Feature Name:** Peer Review

**User Story Title:** As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

**Test Case Creation Template:**

**Test Case ID:** (Provide a unique identifier for each test case)

**Test Case Title:** (Must start with "Verify that..." and clearly state the feature/function being tested)

**Test Case Description:** (A brief description of the test case, between 70-100 words)

**Test Suite:** Peer Review

**Test Priority:** (Priority level: High, Medium, Low)

**Preconditions:**

- User is logged in as a learner.
- A course with assignments is available.
- The learner has submitted an assignment for peer review.

**Test Data:** (Specify data required for execution; if none, state "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, in unordered list)

**Expected Result:** (Overall expected outcome of the test case)

**Severity:** (Blocker, Critical, Major, Minor, Trivial)

**Type of Testing:** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional

Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_PR\_01

**\*\*Test Case Title:\*\*** Verify that learners can submit assignments for peer review.

**\*\*Test Case Description:\*\*** This test case ensures that learners can submit their completed assignments for peer review by other learners within the course.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course page and select the assignment.
2. Click on the "Submit for Review" button.
3. Verify that the assignment is successfully submitted for peer review.

**\*\*Postconditions:\*\***

- The submitted assignment is visible in the "Peer Review" section of the course.
- The learner can view their own submitted assignment.

**\*\*Expected Result:\*\*** The system successfully receives the learner's assignment and displays it in the "Peer Review" section of the course.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_PR\_02

**\*\*Test Case Title:\*\*** Verify that peers can provide feedback on submitted assignments.

**\*\*Test Case Description:\*\*** This test case verifies that learners can access and provide feedback on other learners' submitted assignments.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Peer Review" section of the course.
2. Select an assignment to review.
3. Provide feedback using the provided feedback form, including comments and ratings.
4. Submit the review.

**\*\*Postconditions:\*\***

- The feedback is saved and associated with the reviewed assignment.
- The reviewer can view their submitted feedback.

**\*\*Expected Result:\*\*** The system allows reviewers to provide feedback on the assignment and saves the submitted feedback.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC\_PR\_03**

**\*\*Test Case Title:\*\* Verify that a structured review process guides the feedback.**

**\*\*Test Case Description:\*\* This test case verifies that the system provides a structured framework for providing feedback, ensuring quality and consistency.**

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to an assignment requiring peer review.
2. Access the review form and observe the provided feedback guidelines.
3. Verify that the form includes fields for specific comments, ratings, and areas for improvement.

**\*\*Postconditions:\*\***

- The feedback form provides clear instructions for reviewers.

**\*\*Expected Result:\*\* The review form provides structured elements for feedback, such as specific questions, rating scales, and prompts for constructive criticism.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Usability**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_PR\_04**

**\*\*Test Case Title:\*\* Verify that learners can view feedback on their work.**

**\*\*Test Case Description:\*\* This test case verifies that learners can view the feedback provided on their submitted assignments.**

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the "My Submissions" section within the course.
2. Select the submitted assignment.
3. View the provided feedback, including comments, ratings, and suggestions.

**\*\*Postconditions:\*\***

- The learner can view all feedback provided on the assignment.

**\*\*Expected Result:\*\*** The system displays all feedback received from reviewers on the specific assignment.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_PR\_05

**\*\*Test Case Title:\*\*** Verify that reviewers remain anonymous to ensure unbiased feedback.

**\*\*Test Case Description:\*\*** This test case verifies that reviewers remain anonymous to promote unbiased and honest feedback.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Submit an assignment for peer review.
2. View the feedback provided on the assignment.
3. Verify that the reviewer's identity is hidden.

**\*\*Postconditions:\*\***

- The identity of the reviewer is not displayed to the assignment submitter.

**\*\*Expected Result:\*\*** The system only displays the feedback and does not reveal the reviewer's identity.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Security

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 59**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Certificates and Badges

### **Feature description -**

Upon successful completion of courses or specific milestones, learners receive certificates and badges to recognize their achievements and accomplishments. These credentials provide tangible recognition of learners' efforts, skills, and expertise, enhancing their credentials and demonstrating their commitment to continuous learning and professional development.

### **Userstory title -**

As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

### **Acceptance criteria -**

['Learners earn certificates upon course completion.', 'Badges are awarded for specific achievements and milestones.', 'Certificates and badges are accessible through learner profiles.', 'Learners can share their certificates and badges on social media.', 'Instructors can issue custom badges for special achievements.']

### **Generated Test Cases:**



## ## Test Cases for Online Learning Platform - Certificates and Badges

### \*\*Project Context:\*\*

\* 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.

\* User Story Title: 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:\*\* TC-001

**\*\*Test Case Title:\*\*** Verify that learners earn certificates upon course completion.

**\*\*Test Case Description:\*\*** This test case verifies that learners successfully receive certificates upon completing a course. The test involves a learner enrolling in a course, completing all required modules, and verifying that a certificate is generated and accessible.

**\*\*Test Suite:\*\*** Certificate and Badge Management

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* A learner account is created and logged in.

\* A course is available for enrollment.

**\*\*Test Data:\*\***

\* Course details (title, modules, completion criteria).

**\*\*Test Steps:\*\***

1. Enroll in the selected course.
2. Complete all required modules and assessments within the course.
3. Navigate to the learner's profile.
4. Check for the generated certificate in the "Certificates" section.

**\*\*Postconditions:\*\***

\* The learner's profile displays the newly earned certificate.

**\*\*Expected Result:\*\*** The system generates and displays the certificate upon successful completion of the course.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-002

**\*\*Test Case Title:\*\*** Verify that badges are awarded for specific achievements and milestones.

**\*\*Test Case Description:\*\*** This test case verifies that badges are awarded to learners who achieve predefined

milestones or complete specific actions within a course or the platform.

**\*\*Test Suite:\*\*** Certificate and Badge Management

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* A learner account is created and logged in.

\* A course is available for enrollment with defined milestones or achievements.

**\*\*Test Data:\*\***

\* Course details (milestones/achievements with corresponding badges).

**\*\*Test Steps:\*\***

1. Enroll in the selected course.

2. Complete a defined milestone or achievement within the course.

3. Navigate to the learner's profile.

4. Check for the awarded badge in the "Badges" section.

**\*\*Postconditions:\*\***

\* The learner's profile displays the earned badge.

**\*\*Expected Result:\*\*** The system awards the appropriate badge upon completion of the milestone or achievement.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-003

**\*\*Test Case Title:\*\*** Verify that certificates and badges are accessible through learner profiles.

**\*\*Test Case Description:\*\*** This test case verifies that certificates and badges earned by learners are accessible through their profiles on the platform.

**\*\*Test Suite:\*\*** Learner Profile Management

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* A learner account is created and logged in.
- \* The learner has earned at least one certificate and one badge.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the learner's profile.
2. Check for the "Certificates" and "Badges" sections.
3. Verify that the earned certificates and badges are displayed in the respective sections.

**\*\*Postconditions:\*\***

- \* The learner's profile displays the earned certificates and badges.

**\*\*Expected Result:\*\*** The earned certificates and badges are accessible and visible within the learner's profile.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-004

**\*\*Test Case Title:\*\*** Verify that learners can share their certificates and badges on social media.

**\*\*Test Case Description:\*\*** This test case verifies that learners can share their earned certificates and badges on social media platforms like LinkedIn, Facebook, or Twitter.

**\*\*Test Suite:\*\*** Social Media Integration

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- \* A learner account is created and logged in.
- \* The learner has earned at least one certificate and one badge.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the learner's profile.
2. Select a certificate or badge to share.
3. Choose the desired social media platform.
4. Verify that the sharing options are available and function correctly.

**\*\*Postconditions:\*\***

\* The selected certificate or badge is shared successfully on the chosen platform.

**\*\*Expected Result:\*\*** The learner can share their certificates and badges on social media platforms with ease.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC-005

**\*\*Test Case Title:\*\*** Verify that instructors can issue custom badges for special achievements.

**\*\*Test Case Description:\*\*** This test case verifies that instructors have the capability to create and issue custom badges for specific achievements within their courses.

**\*\*Test Suite:\*\*** Instructor Management

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* An instructor account is created and logged in.

\* A course is available for enrollment.

**\*\*Test Data:\*\***

\* Custom badge details (name, description, image, criteria).

**\*\*Test Steps:\*\***

1. Navigate to the instructor's dashboard.
2. Access the course settings.
3. Create a custom badge with a unique name, description, and criteria.

4. Assign the custom badge to a specific achievement within the course.

5. A learner completes the designated achievement.

6. Verify that the custom badge is awarded to the learner.

**\*\*Postconditions:\*\***

\* The custom badge is successfully created and assigned within the course.

**\*\*Expected Result:\*\*** The system enables instructors to create custom badges and award them to learners for specific achievements within their courses.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 60**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Gamification Elements

### **Feature description -**

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

### **Userstory title -**

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

### **Acceptance criteria -**

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

### **Generated Test Cases:**

## Test Cases for Online Learning Platform - Gamification Elements

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Gamification Elements**

**\*\*User Story Title:\*\* As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.**

**\*\*Acceptance Criteria:\*\***

1. Learners earn points for completing activities.
2. Levels are achieved based on accumulated points.
3. Leaderboards display top-performing learners.
4. Badges are awarded for reaching milestones.
5. Learners can view their progress in a gamification dashboard.

**\*\*Test Case Template:\*\***

**\*\*Test Case ID:\*\* (Unique identifier)**

**\*\*Test Case Title:\*\* Verify that... (Clear statement of the feature being tested)**

**\*\*Test Case Description:\*\*** (Brief description of the test case)

**\*\*Test Suite:\*\*** Gamification Elements

**\*\*Test Priority:\*\*** (High, Medium, Low)

**\*\*Preconditions:\*\***

- User is logged in
- User has enrolled in a course

**\*\*Test Data:\*\*** (If applicable)

**\*\*Test Steps:\*\*** (List of steps to execute the test case)

**\*\*Postconditions:\*\*** (Follow-up actions after test execution)

**\*\*Expected Result:\*\*** (Desired outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Functional, Performance, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_Gamification\_Points

**\*\*Test Case Title:\*\*** Verify that learners earn points for completing activities.

**\*\*Test Case Description:\*\*** This test case verifies that points are awarded to learners upon successful completion of learning activities within a course, such as watching videos, completing quizzes, or engaging in discussions.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** Video lectures, quizzes, discussion forum posts.

**\*\*Test Steps:\*\***

1. Enroll in a course with video lectures, quizzes, and a discussion forum.
2. Watch a video lecture in the course.
3. Complete a quiz within the course.



4. Post a message in the course discussion forum.

5. Check the learner's points balance.

**\*\*Postconditions:\*\*** Points should be reflected in the learner's profile.

**\*\*Expected Result:\*\*** The learner's points balance should increase after completing each activity.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_Gamification\_Levels

**\*\*Test Case Title:\*\*** Verify that levels are achieved based on accumulated points.

**\*\*Test Case Description:\*\*** This test case ensures that learners progress through levels based on their earned points, with clear point thresholds for each level.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** Predefined point thresholds for each level.

**\*\*Test Steps:\*\***

1. Check the initial level of the learner.
2. Complete activities to earn points (e.g., watch videos, complete quizzes).
3. Track the learner's point accumulation and level progression.
4. Compare the earned points with the predefined point thresholds for each level.

**\*\*Postconditions:\*\*** The learner's level should update based on their earned points.

**\*\*Expected Result:\*\*** The learner's level should be updated in their profile as they cross point thresholds for subsequent levels.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

### **\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_Gamification\_Leaderboard

**\*\*Test Case Title:\*\*** Verify that leaderboards display top-performing learners.

**\*\*Test Case Description:\*\*** This test case validates the functionality of the leaderboard, ensuring it accurately displays the top-performing learners based on accumulated points, level, and other relevant metrics.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** A list of learners with varying point totals and levels.

**\*\*Test Steps:\*\***

1. Access the leaderboard for the course or the entire platform.
2. Verify that the leaderboard displays a ranked list of learners.
3. Check if the learners are ranked correctly based on their accumulated points and/or levels.

**\*\*Postconditions:\*\*** The leaderboard should display the learners accurately, with updates reflecting their progress.

**\*\*Expected Result:\*\*** The leaderboard should display the top-performing learners, reflecting their positions based on their achievements.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

### **\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_Gamification\_Badges

**\*\*Test Case Title:\*\*** Verify that badges are awarded for reaching milestones.

**\*\*Test Case Description:\*\*** This test case ensures that the system automatically awards badges to learners when they achieve predefined milestones within the learning platform.

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\*** Predefined milestones and associated badges (e.g., completing a course, achieving a certain level, completing a set number of quizzes).

**\*\*Test Steps:\*\***

1. Enroll in a course with defined badges.
2. Complete activities to meet the criteria for different badges (e.g., complete a certain percentage of the course, achieve a specific level).
3. Check for awarded badges in the learner's profile.

**\*\*Postconditions:\*\*** Badges should be awarded to the learner's profile.

**\*\*Expected Result:\*\*** The system should automatically award the appropriate badges to the learner's profile when they reach the predefined milestones.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC\_Gamification\_Dashboard**

**\*\*Test Case Title:\*\* Verify that learners can view their progress in a gamification dashboard.**

**\*\*Test Case Description:\*\*** This test case ensures that learners can access and view their progress in the gamification dashboard, including their earned points, levels, badges, and other relevant gamification metrics.

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the gamification dashboard in the learner's profile.
2. Verify that the dashboard displays the learner's points, level, earned badges, and progress towards any other gamification goals.

3. Check if the dashboard allows learners to track their progress over time, including their point history and level progression.

**\*\*Postconditions:\*\*** The learner should be able to access and understand their gamification progress.

**\*\*Expected Result:\*\*** The gamification dashboard should display the learner's progress in an easily understandable and interactive manner.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*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 Online Learning Platform - Multilingual Support

**Project Name:** Online Learning Platform

**Feature Name:** Multilingual Support

**User Story Title:** As a learner, I want multilingual support so that I can access the platform in my preferred language.

**Test Case Creation Template:**

**Test Case ID:** (Provide a unique identifier for each test case)

**Test Case Title:** (Must start with "Verify that..." and clearly state the feature/function being tested)

**Test Case Description:** (A brief description of the test case, between 70-100 words)

**Test Suite:** Multilingual Support

**Test Priority:** (Priority level: High, Medium, Low)

**Preconditions:** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**Test Data:** (Specify data required for execution; if none, state "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, in unordered list)

**Expected Result:** (Overall expected outcome of the test case)

**Severity:** (Blocker, Critical, Major, Minor, Trivial)

**Type of Testing:** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional

Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_MLS\_01

**\*\*Test Case Title:\*\*** Verify that the platform interface is available in multiple languages.

**\*\*Test Case Description:\*\*** This test case checks if the platform's user interface (UI) is translated and accessible in multiple languages.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the platform.
- The platform supports multiple languages (e.g., English, Spanish, French).

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the platform's homepage.
2. Locate the language selection dropdown/option.
3. Select a language different from the default language.
4. Observe the platform's UI elements (menu items, buttons, text) for accurate translation.

**\*\*Postconditions:\*\*** The platform's UI is displayed in the selected language.

**\*\*Expected Result:\*\*** The platform's interface elements are translated into the selected language.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC\_MLS\_02**

**\*\*Test Case Title:\*\*** Verify that course materials are available in different languages.

**\*\*Test Case Description:\*\*** This test case verifies if course materials (including video lectures, transcripts, assignments, and quizzes) are translated and accessible in multiple languages.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the platform.
- The platform supports multiple languages (e.g., English, Spanish, French).
- A course is selected.

**\*\*Test Data:\*\*** Selected course material (e.g., video lecture, transcript).

**\*\*Test Steps:\*\***

1. Enroll in a course.
2. Access the course materials.
3. Locate the language selection option for the course material.
4. Select a language different from the default language.
5. Observe if the course material is translated and displayed in the selected language.

**\*\*Postconditions:\*\*** Course material is displayed in the selected language.

**\*\*Expected Result:\*\*** Course materials are translated and accessible in the selected language.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC\_MLS\_03**

**\*\*Test Case Title:\*\* Verify that learners can select their preferred language at any time.**

**\*\*Test Case Description:\*\* This test case checks if learners can switch their preferred language on the platform at any time during their session.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the platform.
- The platform supports multiple languages.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to any page on the platform.
2. Locate the language selection option (dropdown, button, etc.).
3. Select a language different from the current language.
4. Observe the platform's UI elements (menu items, buttons, text) for accurate translation.

**\*\*Postconditions:\*\* The platform's UI is displayed in the newly selected language.**

**\*\*Expected Result:\*\* Learners can switch their preferred language on the platform at any time, and the UI updates accordingly.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_MLS\_04**

**\*\*Test Case Title:\*\* Verify that multilingual support includes translated help and support resources.**

**\*\*Test Case Description:\*\* This test case ensures that help and support resources (FAQs, contact information, user manuals, etc.) are translated into the supported languages.**



**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in to the platform.
- The platform supports multiple languages.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the platform's help and support section.
2. Select a language different from the default language.
3. Observe if the help and support content is accurately translated.
4. Verify that contact information and other relevant details are also translated.

**\*\*Postconditions:\*\* Help and support resources are displayed in the selected language.**

**\*\*Expected Result:\*\* All help and support resources are translated into the supported languages, providing a seamless experience for users.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC\_MLS\_05**

**\*\*Test Case Title:\*\* Verify that instructors can provide content in multiple languages.**

**\*\*Test Case Description:\*\* This test case checks if instructors have the ability to upload and provide course content in multiple languages.**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- Instructor is logged in to the platform.
- The platform supports multiple languages.

- Instructor is creating or editing a course.

**\*\*Test Data:\*\*** Course content (e.g., video lectures, text documents, quizzes).

**\*\*Test Steps:\*\***

1. Log in as an instructor and create a new course.
2. Upload course materials.
3. Choose the option to provide materials in multiple languages.
4. Observe if the platform allows selecting different languages for each uploaded material.
5. Upload materials in different languages.
6. Verify that the uploaded materials are associated with their respective languages.

**\*\*Postconditions:\*\*** Course content is associated with the selected languages.

**\*\*Expected Result:\*\*** Instructors can provide course content in multiple languages, allowing learners to access materials in their preferred language.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*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 learning platform offering courses, lectures, tutorials, and quizzes on various subjects and topics.

\* 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.

\* 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	Type of Testing	Test Case Approach
TC-01	Verify that the platform is compatible with screen readers	This test case ensures that the online learning platform functions correctly with popular screen reader software, allowing users with visual impairments to access and navigate the platform.	Accessibility Features	High	- User is logged in.	No test data needed.	1. Launch the platform with a screen reader (e.g., JAWS, NVDA) enabled. 2. Navigate through the platform using screen reader commands. 3. Access and interact with key elements like course listings, videos, quizzes, and forum discussions.	- No errors or crashes are encountered.	The platform is fully compatible with the screen reader, allowing the user to navigate and interact with all features using the screen reader.	Blocker	Accessibility Testing, Integration Testing	Positive
TC-02	Verify that learners can adjust font sizes for better readability	This test case ensures that learners can adjust the font size of the content on the platform to suit their individual needs and preferences.	Accessibility Features	High	- User is logged in.	No test data needed.	1. Access the platform. 2. Locate the font size adjustment options within the settings or user preferences. 3. Increase and decrease the font size to multiple levels.	- Font size changes are reflected across all platform elements.	The font size adjusts seamlessly across all content on the platform, enhancing readability for learners with various visual needs.	Major	Usability Testing, Functional Testing	Positive

| TC-03 | Verify that color contrast options are available for improved visibility | This test case ensures that learners can select from different color contrast options to enhance visibility and readability of the content on the platform. | Accessibility Features | High | - User is logged in. | No test data needed. | 1. Access the platform. 2. Locate the color contrast options within the settings or user preferences. 3. Choose different contrast options. | - The color contrast changes affect the entire platform interface. | The color contrast options improve visibility of text and other content, making the platform more accessible to learners with low vision or colorblindness. | Major | Accessibility Testing, Functional Testing | Positive |

| TC-04 | Verify that accessibility features are easy to activate and customize | This test case checks if the accessibility features are readily accessible and easy to use, enabling learners to customize their experience without unnecessary difficulty. | Accessibility Features | Medium | - User is logged in. | No test data needed. | 1. Access the platform. 2. Locate the accessibility settings menu. 3. Activate and deactivate various accessibility features (screen reader mode, font size, contrast settings). 4. Verify that the features activate and deactivate smoothly, and changes are reflected instantly on the platform. | - Users can easily activate, deactivate, and customize accessibility features through a user-friendly interface. | Accessibility features are accessible and customizable through a clear and intuitive user interface, ensuring ease of use for learners with diverse needs. | Minor | Usability Testing, Functional Testing | Positive |

| TC-05 | Verify that support resources are available to assist learners with disabilities | This test case checks if the platform provides readily available support resources for learners with disabilities, ensuring they can access assistance when needed. | Accessibility Features | Medium | - User is logged in. | No test data needed. | 1. Access the platform. 2. Look for sections or links related to accessibility or support for learners with disabilities. 3. Verify the presence of information, contact details, or FAQs related to accessibility and disability support. | - Support resources are clearly accessible on the platform. | Users can easily access information, contact details, and support resources related to accessibility and disability assistance on the platform. | Minor | Accessibility Testing, Usability 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 -**

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 Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Discussion Moderation**

**\*\*User Story Title:\*\*** 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 Creation Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Discussion Moderation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor is logged in
- A discussion forum with posts exists

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC\_DM\_01**

**\*\*Test Case Title:\*\* Verify that instructors can edit posts in discussions.**

**\*\*Test Case Description:\*\* This test case ensures that an instructor can edit a post within a discussion forum.**

**\*\*Test Data:\*\* A post with incorrect information.**

**\*\*Test Steps:\*\***

1. Log in as an instructor.
2. Navigate to a discussion forum.
3. Select a post to edit.
4. Edit the content of the post.
5. Save the changes.

**\*\*Postconditions:\*\***

- The edited post is displayed correctly in the discussion forum.

**\*\*Expected Result:\*\* The post is successfully edited, and the updated content is reflected in the discussion forum.**

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TC\_DM\_02**

**\*\*Test Case Title:\*\* Verify that instructors can delete posts in discussions.**

**\*\*Test Case Description:\*\* This test case verifies that an instructor can delete a post from a discussion forum.**

**\*\*Test Data:\*\* A post that violates community guidelines.**

**\*\*Test Steps:\*\***



1. Log in as an instructor.
2. Navigate to a discussion forum.
3. Select a post to delete.
4. Confirm the deletion of the post.

**\*\*Postconditions:\*\***

- The deleted post is no longer visible in the discussion forum.

**\*\*Expected Result:\*\*** The post is deleted successfully, and it is no longer accessible within the discussion.

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TC\_DM\_03

**\*\*Test Case Title:\*\*** Verify that instructors can mute disruptive users.

**\*\*Test Case Description:\*\*** This test case checks that an instructor can mute a user who is engaging in disruptive behavior within a discussion forum.

**\*\*Test Data:\*\*** A user who repeatedly posts irrelevant or offensive content.

**\*\*Test Steps:\*\***

1. Log in as an instructor.
2. Navigate to a discussion forum.
3. Identify a user who is disruptive.
4. Mute the user.

**\*\*Postconditions:\*\***

- The muted user cannot post in the discussion forum.

**\*\*Expected Result:\*\*** The disruptive user is successfully muted, and their posts are hidden from the discussion view.

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* TC\_DM\_04**

**\*\*Test Case Title:\*\*** Verify that community guidelines are visible to all participants.

**\*\*Test Case Description:\*\*** This test case verifies that the community guidelines are easily accessible and visible to all participants in the platform.

**\*\*Test Data:\*\*** No specific test data required.

**\*\*Test Steps:\*\***

1. Log in as a student.
2. Navigate to the discussion forum.
3. Locate the community guidelines section.
4. Verify that the guidelines are clearly displayed and easy to understand.

**\*\*Postconditions:\*\***

- The user is able to read and understand the community guidelines.

**\*\*Expected Result:\*\*** The community guidelines are clearly visible and accessible to all participants, regardless of their role.

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC\_DM\_05**

**\*\*Test Case Title:\*\*** Verify that notifications are sent for moderated actions.

**\*\*Test Case Description:\*\*** This test case verifies that a notification is sent to the user when a moderation action is taken on their post or account.

**\*\*Test Data:\*\*** A post that violates community guidelines.

**\*\*Test Steps:\*\***

1. Log in as a student.
2. Post a comment that violates community guidelines in a discussion forum.
3. Log in as an instructor.
4. Delete the student's post.

**\*\*Postconditions:\*\***

- The student receives a notification informing them about the deleted post.

**\*\*Expected Result:\*\*** The student receives a notification detailing the moderation action and the reason for the action being taken.

-----

## **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 Online Learning Platform - Learning Analytics

**\*\*Project Context:\*\***

\* 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.

\* User Story Title: As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

**\*\*Acceptance Criteria:\*\***

1. Learning analytics dashboards display engagement metrics.
2. Performance trends are visualized over time.
3. Instructors can access detailed reports on learner progress.
4. Analytics data is updated in real-time.
5. Instructors can export analytics reports for further analysis.

**\*\*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	Type of Testing	Test Case Approach
---	---	---	---	---	---	---	---	---	---	---	---	---
TC-LA-01	Verify that Learning Analytics dashboard displays engagement metrics	Ensure that the Learning Analytics dashboard displays relevant engagement metrics like course views, video completion rates, forum participation, quiz attempts, and assignment submissions.	Learning Analytics	High	Instructor is logged in and has access to a course with student activity.	No test data needed	1. Navigate to the Learning Analytics dashboard for the chosen course. 2. Verify that the dashboard displays metrics like course views, video completion rates, forum participation, quiz attempts, and assignment submissions. 3. Check if the displayed data is relevant and accurate.	No postconditions.	The Learning Analytics dashboard should display relevant engagement metrics for the chosen course.	Major	Functional	Positive
TC-LA-02	Verify that Performance trends are visualized over time	Ensure that the Learning Analytics dashboard visualizes performance trends over time, showing how learner engagement and performance evolve over the course duration.	Learning Analytics	High	Instructor is logged in and has access to a course with sufficient student activity over time.	No test data needed	1. Navigate to the Learning Analytics dashboard for the chosen course. 2. Verify that the dashboard displays graphs or charts that visualize performance trends over time. 3. Check if these visualizations cover key metrics like video completion rates, quiz scores, and assignment grades.	No postconditions.	Performance trends should be visualized over time using graphs or charts, covering key metrics like video completion rates, quiz scores, and assignment grades.	Major	Functional	Positive
TC-LA-03	Verify that Instructors can access detailed reports on learner progress	Ensure that instructors can access detailed reports that provide individual learner progress data, including quiz scores, assignment grades, and overall course performance.	Learning Analytics	High	Instructor is logged in and has access to a course with student activity.	No test data needed	1. Navigate to the Learning Analytics dashboard for the chosen course. 2. Access the individual learner progress report section. 3. Verify that detailed reports provide					

data on each learner's quiz scores, assignment grades, and overall course performance. | No postconditions. |

The Learning Analytics dashboard should allow instructors to access detailed reports that provide individual learner progress data, including quiz scores, assignment grades, and overall course performance. | Major |

Functional | Positive |

| TC-LA-04 | Verify that Analytics data is updated in real-time | Ensure that the Learning Analytics data is updated in real-time, reflecting any new student activity or changes in the course. | Learning Analytics | High | Instructor is logged in and has access to a course with active student participation. | No test data needed | 1.

Navigate to the Learning Analytics dashboard for the chosen course. 2. Observe the data displayed for various metrics, like video views, quiz attempts, or forum posts. 3. Initiate a new student activity, such as watching a video, taking a quiz, or posting in the forum. 4. Verify that the Learning Analytics dashboard updates the data in real-time, reflecting the new activity. | No postconditions. | The Learning Analytics dashboard should update the data in real-time, reflecting any new student activity or changes in the course. | Major | Functional | Positive |

| TC-LA-05 | Verify that Instructors can export analytics reports for further analysis | Ensure that instructors can export analytics reports in various formats, such as CSV, PDF, or Excel, for offline analysis and use in other tools. | Learning Analytics | High | Instructor is logged in and has access to a course with student activity. | No test data needed | 1. Navigate to the Learning Analytics dashboard for the chosen course. 2. Locate the export options for available reports. 3. Select a desired export format (e.g., CSV, PDF, Excel). 4. Download and verify the exported report. | The report should be downloaded successfully in the chosen format. | Minor | Functional | 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 Context:\*\***

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.

User Story Title: As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Acceptance Criteria:

- 1. Learners can create and join groups for projects.
- 2. Collaboration tools are available for group communication.
- 3. Learners can share resources and feedback within groups.
- 4. Instructors can assign group projects and monitor progress.
- 5. Social sharing features allow learners to share 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	Type of Testing	Test Case Approach
SLF-TC-01	Verify that learners can create and join groups for projects	This test case verifies the functionality of group creation and joining for learners. It ensures that the system allows learners to create new groups for projects and join existing groups based on their interests or project needs.	Social Learning Features	High	User is logged in	No test data needed	1. Navigate to the 'Groups' or 'Social' section of the platform. 2. Click on the 'Create Group' button. 3. Enter the group name, description, and any relevant settings. 4. Click on the 'Create Group' button to create the group. 5. Navigate to the 'Groups' section and search for the newly created group. 6. Click on the 'Join Group' button for the selected group.	The user is successfully added to the group.	The system allows learners to create and join groups successfully.	Major	Functional	Positive



| SLF-TC-02 | Verify that collaboration tools are available for group communication | This test case ensures the availability of communication tools within the group for collaboration. It checks whether users can communicate with other group members through messages, forums, or other methods. | Social Learning Features | High | User is logged in, a group is created | No test data needed | 1. Navigate to the group's page. 2. Access the group's communication tools (e.g., message board, forum, chat). 3. Send a message to other members within the group. 4. Receive a message from another member in the group. 5. Respond to the message and check if the message is delivered. | Messages are exchanged between group members. | The system provides communication tools for effective collaboration within the group. | Major | Functional | Positive |

| SLF-TC-03 | Verify that learners can share resources and feedback within groups | This test case validates the functionality of resource sharing and feedback provision within groups. It ensures that learners can share relevant files, links, or comments within their groups. | Social Learning Features | High | User is logged in, a group is created | - A document file, a web link, and a comment | 1. Navigate to the group's page. 2. Access the 'Resources' or 'Files' section. 3. Upload a document file, share a web link, or post a comment. 4. View the shared resource or comment from other group members. 5. Provide feedback or comments on shared resources. | Shared resources and feedback are visible to all members. | The system allows learners to share resources and provide feedback effectively within groups. | Major | Functional | Positive |

| SLF-TC-04 | Verify that instructors can assign group projects and monitor progress | This test case ensures that instructors can assign group projects to learners and monitor their progress. It checks the system's functionality for assigning projects and viewing individual and group progress. | Social Learning Features | High | User is logged in as an instructor, a group is created | A group project description, submission deadline, and group members | 1. Log in as an instructor. 2. Select a group from the list. 3. Go to the 'Assignments' or 'Projects' section. 4. Click on the 'Create Assignment' or 'Create Project' button. 5. Create a project, assign it to the group, and set a deadline. 6. View the progress of the project by checking individual and group submissions. | The project is successfully assigned, and progress tracking is available for both individual and group submissions. | Major | Functional | Positive |

| SLF-TC-05 | Verify that social sharing features allow learners to share achievements | This test case checks

the functionality of social sharing features for learner achievements. It ensures that learners can share their progress, certificates, or milestones on the platform or social media. | Social Learning Features | Medium | User is logged in, a course is completed | No test data needed | 1. Complete a course or achieve a milestone (e.g., complete a quiz with a high score). 2. Navigate to the profile or achievement section. 3. Find the option to share the achievement (e.g., 'Share Certificate,' 'Share Progress'). 4. Choose a platform to share the achievement (e.g., platform feed, social media). 5. Share the achievement on the selected platform. | The achievement is shared successfully on the chosen platform. | Minor | Functional | Positive |

**Data - 66**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Content Authoring Tools

**Feature description -**

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

**Userstory title -**

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

## Acceptance criteria -

[ 'Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.' ]

## Generated Test Cases:

### ## Test Cases for Online Learning Platform - Content Authoring Tools

**\*\*Test Case ID:\*\*** TC001

**\*\*Test Case Title:\*\*** Verify that instructors can create multimedia presentations

**\*\*Test Case Description:\*\*** This test case verifies that instructors are able to create multimedia presentations using the content authoring tools, including adding slides, images, videos, and audio.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is created and logged in
- Course is created

**\*\*Test Data:\*\***

- Sample text for slides
- Images for slides
- Video file for slides
- Audio file for slides

**\*\*Test Steps:\*\***

1. Navigate to the content authoring tool for the course.
2. Select the option to create a new multimedia presentation.
3. Add multiple slides to the presentation.
4. Add text, images, a video, and audio to different slides.

5. Preview the presentation to ensure all elements are displayed correctly.

**\*\*Postconditions:\*\***

- Multimedia presentation is saved in the course content library.
- Presentation content is accessible for preview.

**\*\*Expected Result:\*\*** Instructors can successfully create a multimedia presentation with various media elements, and the presentation can be previewed without errors.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC002

**\*\*Test Case Title:\*\*** Verify that interactive modules can be developed and integrated

**\*\*Test Case Description:\*\*** This test case checks if instructors can create interactive modules within the course content, including quizzes, polls, and simulations.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is created and logged in
- Course is created

**\*\*Test Data:\*\***

- Questions for a quiz
- Options for a poll
- Interactive simulation content

**\*\*Test Steps:\*\***

1. Navigate to the content authoring tool for the course.
2. Select the option to create a new interactive module.
3. Create a quiz with multiple choice questions and feedback options.

4. Create a poll with multiple options for students to vote.
5. Integrate a simple interactive simulation into the module.
6. Preview the module to check if interactive elements function correctly.

**\*\*Postconditions:\*\***

- Interactive module is saved in the course content library.
- Module elements are functional during preview.

**\*\*Expected Result:\*\*** Instructors can create different types of interactive modules, and these modules can be previewed with their interactive elements working as intended.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC003

**\*\*Test Case Title:\*\*** Verify that rich media resources, such as videos and animations, are supported

**\*\*Test Case Description:\*\*** This test case validates that the authoring tools allow instructors to upload and embed various rich media files, including videos and animations, into their course content.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is created and logged in
- Course is created

**\*\*Test Data:\*\***

- Video file (different formats like .mp4, .mov)
- Animation file (different formats like .gif, .swf)

**\*\*Test Steps:\*\***

1. Navigate to the content authoring tool for the course.
2. Upload a video file to the course content library.

3. Attempt to embed the uploaded video into a presentation or module.
4. Upload an animation file to the course content library.
5. Attempt to embed the uploaded animation into a presentation or module.
6. Preview the content to ensure the media files play correctly.

**\*\*Postconditions:\*\***

- Media files are uploaded to the course content library.
- Media files can be embedded in course content.

**\*\*Expected Result:\*\*** The platform supports uploading and embedding various formats of video and animation files, ensuring they play correctly during preview.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC004

**\*\*Test Case Title:\*\*** Verify that content authoring tools are user-friendly and intuitive

**\*\*Test Case Description:\*\*** This test case assesses the user experience of the content authoring tools by testing their ease of use and intuitive interface.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is created and logged in
- Course is created

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the content authoring tool for the course.
2. Explore the different options and features of the authoring tools.
3. Attempt to create basic content, like a text-based slide or a simple quiz.

4. Observe the layout, design, and functionality of the tools.

5. Rate the user experience based on intuitiveness, ease of use, and clarity of instructions.

**\*\*Postconditions:\*\*** No postconditions

**\*\*Expected Result:\*\*** The authoring tools should have a user-friendly interface with clear navigation, intuitive functionality, and easy-to-understand instructions, allowing instructors to quickly create basic content.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Usability

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC005

**\*\*Test Case Title:\*\*** Verify that instructors can preview and edit content before publishing

**\*\*Test Case Description:\*\*** This test case validates that instructors can preview their created content before publishing it to students, and they have the option to edit the content before making it live.

**\*\*Test Suite:\*\*** Content Authoring Tools

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor account is created and logged in
- Course is created
- Content is created using the authoring tools (e.g., a presentation or module)

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the content authoring tool for the course.
2. Access the previously created content.
3. Utilize the preview function to check the content's appearance and functionality.
4. Identify any errors or areas that require edits.
5. Edit the content based on the identified issues.
6. Preview the edited content again to confirm the corrections.

**\*\*Postconditions:\*\***

- Content is saved with the edits.
- Content is ready for publishing.

**\*\*Expected Result:\*\*** Instructors can preview their created content, identify any issues, edit the content, and then preview the edited content again to ensure the desired outcome before publishing it to students.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 67**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Learning Paths and Curriculum

**Feature description -**

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence and continuity in learners' educational journeys.

**Userstory title -**

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a



systematic manner.

### **Acceptance criteria -**

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

### **Generated Test Cases:**

#### **## Test Cases for Online Learning Platform - Learning Paths and Curriculum**

**\*\*Project Context:\*\*** Online Learning Platform offering courses, lectures, tutorials, and quizzes on various subjects.

**\*\*Feature:\*\*** Learning Paths and Curriculum

**\*\*User Story:\*\*** As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

**\*\*Acceptance Criteria:\*\***

1. Learning paths include sequential courses and modules.
2. Curricula are aligned with specific learning objectives.
3. Learners can track their progress within a learning path.
4. Instructors can customize learning paths for different needs.
5. Notifications are sent for upcoming courses or milestones.

**\*\*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 | Type of Testing | Test Case Approach |  
|---|---|---|---|---|---|---|---|---|---|---|---|---|

| TC-LP-01 | Verify that learning paths include sequential courses and modules | This test case ensures that a learning path is comprised of multiple courses and modules, presented in a specific order. | Learning Paths and Curriculum | High | User is logged in | Learning path with 3 courses and 5 modules | 1. Navigate to a learning path. 2. Verify that the courses and modules are listed in a specific order. 3. Verify that each course and module contains appropriate content. | | The courses and modules are presented in a sequential order, and their content is relevant to the learning path. | Major | Functional Testing | Positive |

| TC-LP-02 | Verify that curricula are aligned with specific learning objectives | This test case validates that the learning path's courses and modules contribute to achieving the stated learning objectives. | Learning Paths and Curriculum | High | User is logged in | Learning path with learning objectives, courses, and modules | 1. Navigate to a learning path. 2. Check the learning objectives for the learning path. 3. Verify that the courses and modules' content align with the learning objectives. | | The courses and modules within a learning path effectively contribute to achieving the specified learning objectives. | Major | Functional Testing | Positive |

| TC-LP-03 | Verify that learners can track their progress within a learning path | This test case ensures that learners can monitor their progress through the learning path, including completed courses and modules. | Learning Paths and Curriculum | Medium | User is logged in and enrolled in a learning path | Learning path with multiple courses and modules, user progress data | 1. Navigate to a learning path. 2. Check the progress tracker. 3. Verify that completed courses and modules are marked as completed. | | The progress tracker accurately reflects the learner's completion status for each course and module within the learning path. | Minor | Functional Testing | Positive |

| TC-LP-04 | Verify that instructors can customize learning paths for different needs | This test case verifies the instructor's ability to modify learning paths, including adding, removing, or reordering courses and modules. | Learning Paths and Curriculum | Medium | Instructor is logged in and has access to a learning path | | 1. Navigate to a learning path as an instructor. 2. Attempt to add, remove, or reorder courses and modules within the learning path. | | The instructor can successfully customize the learning path by adding, removing, or

reordering courses and modules. | Minor | Functional Testing | Positive |

| TC-LP-05 | Verify that notifications are sent for upcoming courses or milestones | This test case confirms that learners receive timely notifications about upcoming courses or milestones within a learning path. | Learning Paths and Curriculum | Low | User is logged in and enrolled in a learning path, email address is configured | Upcoming course or milestone within the learning path | 1. Set an upcoming course or milestone within a learning path. 2. Verify that the learner receives a notification about the upcoming course or milestone via email or within the platform. | | The learner receives a notification about the upcoming course or milestone within the specified timeframe. | Minor | Functional Testing | 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 Online Learning Platform - Attendance Tracking**

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Attendance Tracking**

**\*\*User Story Title:\*\* 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:\*\* Attendance Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Instructor is logged in
- A live session is scheduled and started

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* TC\_ATT\_01**

**\*\*Test Case Title:\*\* Verify that attendance is automatically recorded during live sessions.**

**\*\*Test Case Description:\*\* This test case verifies that the system automatically records attendance for learners who join a live session. The test checks if the system accurately captures the entry and exit timestamps for each participant.**

**\*\*Test Steps:\*\***

1. As an instructor, start a live session.
2. As a learner, join the live session.
3. Stay in the session for a few minutes, then leave the session.
4. As an instructor, end the session.

**\*\*Postconditions:\*\***

- The session is ended.

**\*\*Expected Result:\*\* The system should have recorded attendance for the learner, including the entry and exit timestamps for the live session.**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC\_ATT\_02**

**\*\*Test Case Title:\*\* Verify that instructors can view attendance reports for each session.**

**\*\*Test Case Description:\*\*** This test case checks if instructors can access detailed attendance reports for each live session they have conducted. The report should include the list of attendees, their entry and exit timestamps, and possibly other participation metrics.

**\*\*Test Steps:\*\***

1. As an instructor, start a live session.
2. As a learner, join the live session.
3. Stay in the session for a few minutes, then leave the session.
4. As an instructor, end the session.
5. Navigate to the session details page.
6. Access the attendance report.

**\*\*Postconditions:\*\***

- The attendance report is displayed.

**\*\*Expected Result:\*\*** The attendance report should display a list of all learners who attended the session, including their entry and exit times.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_ATT\_03

**\*\*Test Case Title:\*\*** Verify that learners can see their attendance records.

**\*\*Test Case Description:\*\*** This test case ensures that learners can access their attendance history for past live sessions. This information should be easily accessible and displayed in a clear format.

**\*\*Test Steps:\*\***

1. As a learner, join a live session.
2. Stay in the session for a few minutes, then leave the session.
3. Navigate to the learner's profile page or the session details page.

4. Access the attendance record section.

**\*\*Postconditions:\*\***

- Attendance record section is displayed.

**\*\*Expected Result:\*\*** The system should display a record of the learner's attendance for the specific session, including the session name, date, entry and exit times, and possibly the duration of attendance.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_ATT\_04

**\*\*Test Case Title:\*\*** Verify that attendance data is integrated with participation metrics.

**\*\*Test Case Description:\*\*** This test case checks if the attendance data is linked or correlated with other participation metrics, such as the time spent in the session, questions asked, or interactions within the session. The integration should provide a holistic view of learner engagement and participation.

**\*\*Test Steps:\*\***

1. As a learner, join a live session.
2. Participate in the session by asking questions, answering polls, or engaging with other learners.
3. Leave the session after a few minutes.
4. Access the session details page or learner profile page.
5. Review the participation metrics and attendance data.

**\*\*Postconditions:\*\***

- Participation metrics are displayed.

**\*\*Expected Result:\*\*** The attendance data should be integrated with participation metrics like time spent in the session, interaction count, questions asked, etc. The system should provide a comprehensive view of the learner's participation and engagement during the session.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_ATT\_05

**\*\*Test Case Title:\*\*** Verify that notifications are sent for missed sessions.

**\*\*Test Case Description:\*\*** This test case verifies that learners receive notifications for missed live sessions.

These notifications should be timely and informative, reminding learners of their missed sessions and potentially providing links to recordings or other relevant information.

**\*\*Test Steps:\*\***

1. As a learner, register for a live session.
2. Do not join the live session at the scheduled time.
3. Check the learner's notification settings and communication channels.

**\*\*Postconditions:\*\***

- Notifications are displayed.

**\*\*Expected Result:\*\*** The system should send a notification to the learner, informing them about the missed live session, possibly including details about the session topic, instructor, and links to any recordings or materials.

**\*\*Test Case Approach:\*\*** 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 Online Learning Platform - Content Licensing and Rights Management

**\*\*Project Name:\*\*** Online Learning Platform

**\*\*Feature Name:\*\*** Content Licensing and Rights Management

**\*\*User Story Title:\*\*** As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** Content Licensing and Rights Management

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** CLRM-TC-01

**\*\*Test Case Title:\*\*** Verify that content licenses are assigned to each resource.

**\*\*Test Case Description:\*\*** This test case verifies that the system automatically assigns a content license to each uploaded resource (e.g., video lecture, document, quiz). The test checks if the correct license is assigned based on the creator's selection and whether the system correctly displays the license information associated with the resource.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Content creator is logged in.
- A new resource (e.g., video lecture) is uploaded.

**\*\*Test Data:\*\***

- New video lecture resource

**\*\*Test Steps:\*\***

1. Log in as a content creator.
2. Upload a new video lecture resource.
3. Select a specific content license (e.g., Creative Commons Attribution-NonCommercial).
4. Save and publish the resource.
5. View the resource details page.

**\*\*Postconditions:\*\***

- The resource is published and visible.

**\*\*Expected Result:\*\*** The system assigns the selected license to the resource, and the license information (name, details, and usage restrictions) is correctly displayed on the resource details page.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** CLRM-TC-02

**\*\*Test Case Title:\*\*** Verify that usage rights and permissions are clearly defined.

**\*\*Test Case Description:\*\*** This test case verifies that the system clearly defines the usage rights and permissions associated with each content license. The test checks if the system provides clear and comprehensive information about what is allowed and restricted under each license, including aspects like distribution, modification, and commercial use.

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Content creator is logged in.
- A resource with a content license assigned is available.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Log in as a content creator.
2. View a resource with a content license assigned.
3. Access the "License Details" section.

**\*\*Postconditions:\*\***

- No postconditions.

**\*\*Expected Result:\*\*** The system provides a clear and comprehensive description of the usage rights and permissions granted by the selected license, including the limitations and restrictions.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* CLRM-TC-03**

**\*\*Test Case Title:\*\* Verify that content creators can set restrictions on usage and distribution.**

**\*\*Test Case Description:\*\*** This test case verifies that the system allows content creators to set specific restrictions on usage and distribution of their resources. The test checks if the system enables content creators to customize license terms beyond the standard options provided, including the ability to restrict specific types of usage, limit geographical access, or prevent commercial use.

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Content creator is logged in.
- A new resource is uploaded.

**\*\*Test Data:\*\***

- New resource (e.g., quiz, document)

**\*\*Test Steps:\*\***

1. Log in as a content creator.
2. Upload a new resource (e.g., quiz).
3. Select a content license.
4. Access the advanced settings or customization options for the license.
5. Set specific restrictions (e.g., restrict commercial use, limit access to specific regions, disable download).
6. Save and publish the resource.

**\*\*Postconditions:\*\***

- The resource is published and visible.

**\*\*Expected Result:\*\*** The system allows content creators to set specific restrictions on usage and distribution according to their needs, and the resource is published with the chosen restrictions applied.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** CLRM-TC-04

**\*\*Test Case Title:\*\*** Verify that instructors and learners are informed of content licensing terms.

**\*\*Test Case Description:\*\*** This test case verifies that the system effectively communicates content licensing terms to instructors and learners. The test checks if the system clearly displays the license information associated with each resource, including the license name, details, and relevant restrictions, to both instructors

accessing the resource and learners consuming the content.

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Instructor and learner accounts are logged in.
- A resource with a content license assigned is available.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Log in as an instructor.
2. Access a resource with a content license assigned.
3. View the license information associated with the resource.
4. Log in as a learner.
5. Access the same resource with a content license.
6. View the license information associated with the resource.

**\*\*Postconditions:\*\***

- No postconditions.

**\*\*Expected Result:\*\*** The system displays the content license information (name, details, restrictions) prominently and clearly on the resource page for both instructors and learners, ensuring transparency and awareness of license terms.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Usability**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* CLRM-TC-05**

**\*\*Test Case Title:\*\* Verify that notifications are sent for license expiration or renewal.**

**\*\*Test Case Description:\*\*** This test case verifies that the system sends timely notifications to content creators when a license associated with their resources is approaching expiration or requires renewal. The test checks if the system triggers notifications with relevant information about the expiring license and provides options for renewal or extension.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Content creator is logged in.
- A resource with a content license approaching expiration is available.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Log in as a content creator.
2. Access a resource with a content license nearing expiration.
3. Monitor the system for any notifications related to license expiration.

**\*\*Postconditions:\*\***

- No postconditions.

**\*\*Expected Result:\*\*** The system sends a timely notification to the content creator about the impending license expiration, providing clear information about the license expiry date and options for renewal or extension.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 70**

**Project name -**

## Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Content Versioning

### **Feature description -**

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

### **Userstory title -**

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

### **Acceptance criteria -**

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.', 'Notifications are sent for significant content updates.']

### **Generated Test Cases:**

## Test Cases for Content Versioning Feature

**\*\*Project Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* Content Versioning**

**\*\*Test Case 1\*\***



**\*\*Test Case ID:\*\* CV-01**

**\*\*Test Case Title:\*\* Verify that Instructors can create and save multiple versions of content**

**\*\*Test Case Description:\*\* This test case checks the functionality of creating and saving different versions of course content. It ensures that instructors can create multiple versions of the same content, each with unique modifications and edits.**

**\*\*Test Suite:\*\* Content Versioning**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Instructor is logged in
- Course is created with existing content

**\*\*Test Data:\*\***

- Course content: A sample lecture video, a PDF document, and a quiz

**\*\*Test Steps:\*\***

1. Access the course content editor
2. Make changes to the lecture video (e.g., add new slides, modify existing ones).
3. Save the changes as a new version.
4. Repeat steps 2 and 3 for the PDF document and the quiz, creating a new version for each.

**\*\*Postconditions:\*\***

- Each content item now has multiple versions available.
- The version history for each content item is updated.

**\*\*Expected Result:\*\* The system successfully creates and saves multiple versions of each content item, each with its unique modifications.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

## **\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* CV-02**

**\*\*Test Case Title:\*\*** Verify that version history is maintained and accessible

**\*\*Test Case Description:\*\*** This test case verifies the system's ability to maintain a history of all content versions and allows instructors to access this history. It checks if all versions are recorded and can be retrieved by the instructor.

**\*\*Test Suite:\*\*** Content Versioning

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor is logged in
- Course with content versions exists (created in previous test case)

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Access the course content editor.
2. Navigate to the version history for a content item (e.g., lecture video).
3. View the list of versions and their timestamps.
4. Select a specific version to view its details.

**\*\*Postconditions:\*\***

- The version history is displayed.
- The details of each version are accessible.

**\*\*Expected Result:\*\*** The system displays the version history of the content item with timestamps for each version, allowing the instructor to access the details of each version.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

### **\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* CV-03**

**\*\*Test Case Title:\*\* Verify that instructors can compare different versions of content**

**\*\*Test Case Description:\*\* This test case checks if instructors can compare different versions of the same content to identify changes made between them. It ensures that the system provides a clear and intuitive interface for side-by-side comparison.**

**\*\*Test Suite:\*\* Content Versioning**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- Instructor is logged in
- Course with content versions exists (created in previous test case)

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the course content editor.
2. Navigate to the version history for a content item (e.g., lecture video).
3. Select two different versions for comparison.
4. Use the comparison feature to identify changes between the selected versions.

**\*\*Postconditions:\*\***

- The system highlights differences between the versions.
- The instructor can easily identify changes made between versions.

**\*\*Expected Result:\*\* The system provides a clear side-by-side comparison of the selected versions, highlighting the differences in content, allowing the instructor to quickly identify changes made.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* CV-04**

**\*\*Test Case Title:\*\* Verify that previous versions can be restored if needed**

**\*\*Test Case Description:\*\* This test case checks the functionality of reverting to a previous version of course content. It ensures that instructors can easily restore an older version if they need to undo changes or revert to an earlier draft.**

**\*\*Test Suite:\*\* Content Versioning**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Instructor is logged in
- Course with content versions exists (created in previous test case)

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Access the course content editor.
2. Navigate to the version history for a content item (e.g., lecture video).
3. Select a previous version to restore.
4. Confirm the restoration of the selected version.

**\*\*Postconditions:\*\***

- The content item is restored to the selected version.
- The version history is updated to reflect the restoration.

**\*\*Expected Result:\*\* The system successfully restores the content item to the selected previous version, ensuring that the content reverts to the earlier state.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* CV-05**

**\*\*Test Case Title:\*\* Verify that notifications are sent for significant content updates**

**\*\*Test Case Description:\*\* This test case checks the functionality of sending notifications to students whenever there are significant content updates. It ensures that students are informed about changes that might affect their learning experience.**

**\*\*Test Suite:\*\* Content Versioning**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- Instructor is logged in
- Course with content versions exists (created in previous test case)
- Student is enrolled in the course

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. As an instructor, make significant changes to the content of the lecture video (e.g., add a new module, remove a section).
2. Save the changes as a new version.
3. Check the student's notifications.

**\*\*Postconditions:\*\***

- The student receives a notification about the content update.
- The notification provides information about the updated content.

**\*\*Expected Result:\*\* The system sends a notification to the enrolled student, informing them about the significant content update and providing details about the changes made.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

---

## **Data - 71**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

API Integration

### **Feature description -**

The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

### **Userstory title -**

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

### **Acceptance criteria -**

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

### **Generated Test Cases:**

## Test Cases for Online Learning Platform API Integration

**\*\*Project Context:\*\***

\* \*\*Project Name:\*\* Online Learning Platform

\* \*\*Project Description:\*\* An online learning platform offering courses, lectures, tutorials, and quizzes on various subjects and topics.

\* \*\*Feature Name:\*\* API Integration

\* \*\*Feature Description:\*\* The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities.

\* \*\*User Story 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:\*\*

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 Creation Template:\*\*

\*\*Test Case ID:\*\*

\*\*Test Case Title:\*\*

\*\*Test Case Description:\*\*

\*\*Test Suite:\*\* API Integration

\*\*Test Priority:\*\*

\*\*Preconditions:\*\*

\*\*Test Data:\*\*

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\*** API Testing

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_API\_01

**\*\*Test Case Title:\*\*** Verify that APIs are available for key platform functionalities.

**\*\*Test Case Description:\*\*** This test case verifies the availability of APIs for core platform features like user registration, course enrollment, and content access.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* The Online Learning Platform is running.

\* API endpoints are defined and accessible.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Using a REST client (e.g., Postman), attempt to access API endpoints for user registration, course enrollment, and content access.

2. Observe the HTTP response codes and ensure they are successful (e.g., 200 OK) for each API call.

3. Check the API documentation for details on available endpoints and parameters.

**\*\*Postconditions:\*\***

\* The API endpoints are accessible and functional.

**\*\*Expected Result:\*\*** The API endpoints for user registration, course enrollment, and content access respond with a successful HTTP response code (200 OK) and provide the requested data.



**\*\*Severity:\*\* Blocker**

**\*\*Type of Testing:\*\* API Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC\_API\_02**

**\*\*Test Case Title:\*\* Verify that comprehensive API documentation is provided.**

**\*\*Test Case Description:\*\*** This test case verifies that the platform provides clear and comprehensive documentation for all available APIs, including detailed descriptions, parameters, request/response formats, and usage examples.

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* The Online Learning Platform is running.

\* API documentation is published and accessible.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Access the API documentation provided by the platform.

2. Verify the presence of documentation for each API endpoint, including:

\* API endpoint URL and method.

\* Request parameters and their data types.

\* Response data format and structure.

\* Example requests and responses.

\* Authentication methods and required credentials.

\* Error codes and their descriptions.

**\*\*Postconditions:\*\***

\* The API documentation is complete and accurate.

**\*\*Expected Result:\*\*** The documentation provides comprehensive information on each API endpoint, including descriptions, parameters, request/response formats, usage examples, and error handling.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Documentation Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_API\_03

**\*\*Test Case Title:\*\*** Verify that secure access protocols are implemented.

**\*\*Test Case Description:\*\*** This test case verifies that the platform uses secure access protocols like HTTPS and OAuth2 to protect API communication and user data.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* The Online Learning Platform is running.

\* API endpoints are defined and accessible.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Attempt to access the API endpoints using HTTP instead of HTTPS.
2. Observe the HTTP response code and ensure it indicates an error or redirection to HTTPS.
3. Check if the API endpoints support OAuth2 authentication for secure access control.

**\*\*Postconditions:\*\***

\* API communication is protected using secure protocols.

**\*\*Expected Result:\*\*** The platform only accepts API requests through HTTPS and provides mechanisms for secure authentication using OAuth2 or similar methods.

**\*\*Severity:\*\*** Critical

**\*\*Type of Testing:\*\*** Security Testing

**\*\*Test Case Approach:\*\* Negative**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_API\_04**

**\*\*Test Case Title:\*\* Verify that third-party applications can authenticate and access APIs.**

**\*\*Test Case Description:\*\* This test case verifies that third-party applications can successfully authenticate and access the API endpoints using valid credentials.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

**\* A third-party application is developed and configured to access the API.**

**\* Valid API credentials are provided for authentication.**

**\*\*Test Data:\*\* API credentials, test data for API requests.**

**\*\*Test Steps:\*\***

1. Using the third-party application, attempt to authenticate with the platform's API using provided credentials.
2. After successful authentication, access an API endpoint to retrieve data or perform an action.
3. Verify that the API request is processed successfully and the response data is returned correctly.

**\*\*Postconditions:\*\***

**\* The third-party application is authenticated and has access to the API.**

**\*\*Expected Result:\*\* The third-party application authenticates successfully using provided credentials and accesses the API endpoint, returning the expected data or performing the requested action.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Integration Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC\_API\_05**

**\*\*Test Case Title:\*\* Verify that rate limiting and monitoring features are in place.**

**\*\*Test Case Description:\*\* This test case verifies the implementation of rate limiting and monitoring features to prevent API abuse and ensure system stability.**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

\* The Online Learning Platform is running.

\* API endpoints are defined and accessible.

\* A tool for generating API requests is available (e.g., Postman, curl).

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Send a large number of API requests in a short period to a specific API endpoint.
2. Observe the response codes and error messages returned by the API.
3. Verify that the API implements rate limiting mechanisms, preventing excessive requests from a single source.
4. Check if the API provides monitoring capabilities to track API usage and identify potential abuse patterns.

**\*\*Postconditions:\*\***

\* The API effectively limits excessive API requests.

**\*\*Expected Result:\*\* The API implements rate limiting to prevent abuse and provides monitoring features to track API usage and identify potential issues.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Performance Testing**

**\*\*Test Case Approach:\*\* Negative**

-----

**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 Online Learning Platform Data Analytics and Insights

**\*\*Project Name:\*\*** Online Learning Platform

**\*\*Project Description:\*\*** An online learning platform offering courses, lectures, tutorials, and quizzes on various subjects and topics.

**\*\*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.

**\*\*User Story 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:\*\***

- 1. Data visualization tools are available for analysis.
- 2. Reports can be generated for learner performance metrics.
- 3. Real-time analytics provide insights into engagement and outcomes.
- 4. Data can be exported for further analysis.
- 5. Notifications are sent for significant data trends or anomalies.

**\*\*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	Type of Testing	Test Case Approach
---	---	---	---	---	---	---	---	---	---	---	---	---
TC_DA_01	Verify that data visualization tools are available for analysis	This test case verifies the availability and functionality of data visualization tools within the platform, enabling administrators to analyze learner data effectively.	Data Analytics & Insights	High	Administrator is logged in	No test data needed	1. Navigate to the data analytics section. 2. Verify the presence of various visualization tools like charts, graphs, and dashboards. 3. Interact with different visualization tools and verify their responsiveness and data display.	Visualization tools are displayed and interactive	Data visualization tools are available and function correctly, providing clear and insightful data representations.	Major	Functional Testing	Positive
TC_DA_02	Verify that reports can be generated for learner performance metrics	This test case ensures that the platform generates reports containing relevant learner performance metrics, allowing administrators to assess student progress and identify areas for improvement.	Data Analytics & Insights	High	Administrator is logged in	Course data (e.g., student scores, completion rates)	1. Access the data analytics section. 2.					

Select the desired course and reporting period. 3. Generate a report for specific learner performance metrics (e.g., average scores, completion rates, quiz scores). 4. Verify the report contains accurate and relevant data. | Report is generated and contains data | The system generates reports containing accurate and comprehensive learner performance data, helping administrators analyze and evaluate student progress. | Major | Functional Testing | Positive |

| TC\_DA\_03 | Verify that real-time analytics provide insights into engagement and outcomes | This test case ensures the platform provides real-time analytics that offer insights into learner engagement and course outcomes, allowing administrators to monitor student activity and make data-driven decisions. | Data Analytics & Insights | High | Administrator is logged in, course with active learners | Course data (e.g., active users, video view duration, forum interactions) | 1. Access the data analytics section. 2. Monitor real-time analytics for the selected course. 3. Verify the data updates in real-time, providing insights into user activity and engagement levels. | Analytics update in real-time | The platform provides real-time data analytics that update dynamically, offering insights into learner engagement and course outcomes. | Major | Functional Testing | Positive |

| TC\_DA\_04 | Verify that data can be exported for further analysis | This test case ensures administrators can export data from the platform for further analysis using external tools, enhancing the flexibility of data analysis. | Data Analytics & Insights | Medium | Administrator is logged in | No test data needed | 1. Access the data analytics section. 2. Select the desired data to export. 3. Choose the desired export format (e.g., CSV, Excel). 4. Verify the exported file contains the correct data in the specified format. | Data is exported successfully | Data can be exported in various formats for further analysis, providing flexibility to administrators. | Minor | Functional Testing | Positive |

| TC\_DA\_05 | Verify that notifications are sent for significant data trends or anomalies | This test case ensures that the platform notifies administrators of significant data trends or anomalies, alerting them to potential issues and enabling proactive interventions. | Data Analytics & Insights | Medium | Administrator is logged in, configured to receive notifications | Simulated data trends (e.g., sudden drop in engagement, high failure rates) | 1. Simulate a significant data trend or anomaly. 2. Verify that the administrator receives a notification. 3. Verify that the notification accurately describes the trend or anomaly. | Notification is sent to administrator |

The system sends notifications to administrators regarding significant data trends or anomalies, enabling them to take timely action. | Minor | System 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 Name:\*\* Online Learning Platform**

**\*\*Feature Name:\*\* User Roles and Permissions**

**\*\*User Story Title:\*\* 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:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\* User Roles and Permissions**

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\***

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* TC-URP-01**

**\*\*Test Case Title:\*\* Verify that administrators can create new user roles.**

**\*\*Test Case Description:\*\* Ensure that the system allows administrators to create new user roles with unique names and descriptions.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Administrator is logged in.

**\*\*Test Data:\*\***

- New role name: "Content Creator"
- New role description: "User with permission to create and upload course content."

**\*\*Test Steps:\*\***

1. Navigate to the User Roles management section.
2. Click on the "Create New Role" button.
3. Enter the new role name and description.
4. Click on the "Save" button.

**\*\*Postconditions:\*\***

- The new role is created and displayed in the list of roles.

**\*\*Expected Result:\*\* The system successfully creates the new user role with the specified name and description.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC-URP-02**

**\*\*Test Case Title:\*\* Verify that permissions are customizable for each role.**

**\*\*Test Case Description:\*\* Ensure that administrators can assign and remove specific permissions to different user roles.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- Administrator is logged in.
- A user role exists (e.g., "Content Creator").

**\*\*Test Data:\*\***

- Permission: "Upload Course Content"

**\*\*Test Steps:\*\***

1. Navigate to the User Roles management section.
2. Select the "Content Creator" role.
3. Go to the "Permissions" tab.
4. Enable the "Upload Course Content" permission.
5. Click on "Save Changes".

**\*\*Postconditions:\*\***

- The "Upload Course Content" permission is assigned to the "Content Creator" role.

**\*\*Expected Result:\*\* The system successfully assigns the "Upload Course Content" permission to the "Content Creator" role.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC-URP-03**

**\*\*Test Case Title:\*\* Verify that users are restricted to functionalities based on their assigned roles.**

**\*\*Test Case Description:\*\* Ensure that users with different roles have access to only the features and functionalities permitted for their assigned roles.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- A user is logged in with the "Content Creator" role.
- A user is logged in with the "Student" role (without "Upload Course Content" permission).

**\*\*Test Data:\*\***

- No specific data required.

**\*\*Test Steps:\*\***

1. As the "Content Creator" user, attempt to access and perform actions related to creating and uploading course content.
2. As the "Student" user, attempt to access and perform actions related to creating and uploading course content.

**\*\*Postconditions:\*\***

- The "Content Creator" user should be able to access and perform actions related to course content.
- The "Student" user should be restricted from accessing or performing actions related to course content.

**\*\*Expected Result:\*\* The system restricts users based on their assigned roles and permissions.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Negative**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC-URP-04

**\*\*Test Case Title:\*\*** Verify that changes to roles and permissions are logged.

**\*\*Test Case Description:\*\*** Ensure that the system records all changes made to user roles and permissions, including date, time, user who made the change, and details of the changes.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- Administrator is logged in.
- A user role exists (e.g., "Content Creator").

**\*\*Test Data:\*\***

- No specific data required.

**\*\*Test Steps:\*\***

1. Modify an existing user role by changing its description or permissions.
2. View the audit log or history of changes related to user roles.

**\*\*Postconditions:\*\***

- The audit log should display the changes made to the user role, including the date, time, user who made the change, and the details of the modifications.

**\*\*Expected Result:\*\*** The system successfully logs all changes to roles and permissions, providing a complete audit trail.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Security

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC-URP-05**

**\*\*Test Case Title:\*\* Verify that users are notified of their assigned roles and permissions.**

**\*\*Test Case Description:\*\* Ensure that users receive notifications when they are assigned a new role or when their permissions are changed.**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- A user is registered on the platform.

**\*\*Test Data:\*\***

- No specific data required.

**\*\*Test Steps:\*\***

1. As an administrator, assign a new role to the user.
2. As an administrator, change the permissions of the user's existing role.
3. Check the user's notification settings and email inbox for notifications regarding role and permission changes.

**\*\*Postconditions:\*\***

- The user should receive notifications about the assigned role or changes in permissions via email or within the platform interface.

**\*\*Expected Result:\*\* The system successfully notifies users of their assigned roles and permissions.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*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 Online Learning Platform - User Feedback and Rating System

**\*\*Project Context:\*\***

**\* \*\*Project Name:\*\*** Online Learning Platform

**\* \*\*Project Description:\*\*** An online learning platform offering courses, lectures, tutorials, and quizzes on various subjects.

**\* \*\*Feature Name:\*\*** User Feedback and Rating System

\* \*\*Feature Description:\*\* Allows learners to rate courses, instructors, and learning materials.

\* \*\*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:\*\* User Feedback and Rating System

\*\*Test Priority:\*\*

\*\*Preconditions:\*\*

\*\*Test Data:\*\*

\*\*Test Steps:\*\*

\*\*Postconditions:\*\*

\*\*Expected Result:\*\*

\*\*Severity:\*\*

\*\*Type of Testing:\*\*

\*\*Test Case Approach:\*\*



**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC-001

**\*\*Test Case Title:\*\*** Verify that learners can rate courses and instructors on a scale.

**\*\*Test Case Description:\*\*** This test case verifies that learners can provide ratings for courses and instructors on a predefined scale.

**\*\*Test Suite:\*\*** User Feedback and Rating System

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- User is enrolled in a course.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course page.
2. Locate the rating section for the course.
3. Click on a rating option (e.g., 1 to 5 stars).
4. Repeat steps 2-3 to rate the instructor.

**\*\*Postconditions:\*\***

- The ratings are submitted and displayed on the course page.

**\*\*Expected Result:\*\*** The system successfully records and displays the user's ratings for the course and instructor.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC-002**

**\*\*Test Case Title:\*\* Verify that feedback forms are available for detailed comments.**

**\*\*Test Case Description:\*\* This test case verifies that learners can provide detailed feedback in addition to ratings.**

**\*\*Test Suite:\*\* User Feedback and Rating System**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.
- User is enrolled in a course.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the course page.
2. Locate the feedback form section.
3. Enter detailed feedback in the text box.
4. Submit the feedback form.

**\*\*Postconditions:\*\***

- The feedback is submitted and stored by the system.

**\*\*Expected Result:\*\* The system successfully stores the learner's detailed feedback.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC-003**

**\*\*Test Case Title:\*\* Verify that instructors can view and respond to feedback.**

**\*\*Test Case Description:\*\*** This test case verifies that instructors can access and respond to feedback provided by learners.

**\*\*Test Suite:\*\*** User Feedback and Rating System

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- Instructor is logged in.
- Instructor has created and is teaching a course.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course dashboard.
2. Locate the feedback section for the course.
3. View feedback provided by learners.
4. Compose and submit a response to a feedback.

**\*\*Postconditions:\*\***

- The response is submitted and displayed alongside the feedback.

**\*\*Expected Result:\*\*** The system successfully allows the instructor to view, read, and respond to feedback provided by learners.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC-004

**\*\*Test Case Title:\*\*** Verify that aggregated ratings are displayed publicly.

**\*\*Test Case Description:\*\*** This test case verifies that aggregated ratings for courses and instructors are displayed publicly on the platform.

**\*\*Test Suite:\*\*** User Feedback and Rating System

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- User is viewing a course page.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course page.
2. Locate the rating section.
3. Verify that the average rating is displayed.

**\*\*Postconditions:\*\***

- Aggregated rating is displayed on the course page.

**\*\*Expected Result:\*\*** The system successfully calculates and displays the average rating for the course based on all learner ratings.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC-005

**\*\*Test Case Title:\*\*** Verify that anonymous feedback options are available.

**\*\*Test Case Description:\*\*** This test case verifies that learners can choose to provide feedback anonymously.

**\*\*Test Suite:\*\*** User Feedback and Rating System

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.

- User is enrolled in a course.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the course page.
2. Locate the feedback form.
3. Select the "Anonymous Feedback" option.
4. Submit feedback.

**\*\*Postconditions:\*\***

- Feedback is submitted anonymously.

**\*\*Expected Result:\*\*** The system successfully records feedback without associating it with the user's profile.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 75**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Social Media Integration

### **Feature description -**

The platform supports social media integration, allowing learners and instructors to share their achievements,

courses, and updates on social media platforms. This feature promotes engagement and community building, enhancing the visibility of the platform's offerings.

### **Userstory title -**

As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

### **Acceptance criteria -**

['Learners can connect their social media accounts to the platform.', 'Achievements and course completions can be shared on social media.', 'Privacy settings control the visibility of shared content.', 'Instructors can promote courses and updates through social media.', 'Analytics track the reach and engagement of shared content.']

### **Generated Test Cases:**

## Test Cases for Online Learning Platform - Social Media Integration

**\*\*Project Context:\*\***

**\* \*\*Project Name:\*\*** Online Learning Platform

**\* \*\*Project Description:\*\*** An online learning platform offering courses, lectures, tutorials, and quizzes.

**\* \*\*Feature Name:\*\*** Social Media Integration

**\* \*\*Feature Description:\*\*** Allows users to share achievements, courses, and updates on social media.

**\* \*\*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.

**\*\*Acceptance Criteria:\*\***

1. Learners can connect their social media accounts to the platform.
2. Achievements and course completions can be shared on social media.

- 3. Privacy settings control the visibility of shared content.
- 4. Instructors can promote courses and updates through social media.
- 5. Analytics track the reach and engagement of shared content.

**\*\*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:\*\* | \*\*Type of Testing:\*\* | \*\*Test Case Approach:\*\***

----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | ----- | -----

**\*\*TC\_SMI\_01\*\*** | Verify that learners can connect their social media accounts to the platform | This test case checks if the platform allows learners to connect their social media accounts, ensuring a smooth integration process. | Social Media Integration | High | Learner is logged in | Learner's social media account details | 1. Navigate to the user profile settings. 2. Select "Connect Social Media". 3. Choose a social media platform (e.g., Facebook, Twitter). 4. Authorize the platform to access user data. 5. Verify successful connection. | User account is connected to social media | The platform successfully connects the learner's social media account and displays a confirmation message. | Major | Functional | Positive

**\*\*TC\_SMI\_02\*\*** | Verify that achievements and course completions can be shared on social media | This test case ensures the platform allows learners to share their achievements and course completions to their connected social media platforms. | Social Media Integration | High | Learner is logged in, Course completion status is available, Social media account is connected | No test data needed | 1. Access the learner's profile page. 2. View course completion achievements. 3. Select "Share on Social Media". 4. Choose the desired social media platform. 5. Verify the post is shared on the selected platform. | Achievement/completion is shared on social media | The platform successfully shares the selected achievement/completion to the designated social media platform with relevant details and a link back to the platform. | Major | Functional | Positive

**\*\*TC\_SMI\_03\*\*** | Verify that privacy settings control the visibility of shared content | This test case checks if

the platform provides options to control the visibility of shared achievements and course completions on social media, ensuring user privacy. | Social Media Integration | Medium | Learner is logged in, Course completion status is available, Social media account is connected | No test data needed | 1. Navigate to the privacy settings section. 2. Adjust the visibility settings for social media shares (e.g., public, friends only). 3. Share a course completion on social media. 4. Verify the post's visibility matches the selected privacy settings. | Shared content visibility is restricted as per settings | The platform respects the chosen privacy settings and displays the shared content accordingly on social media. | Major | Security | Positive

**\*\*TC\_SMI\_04\*\*** | Verify that instructors can promote courses and updates through social media | This test case ensures the platform enables instructors to promote courses and share updates on social media, promoting engagement and increasing platform visibility. | Social Media Integration | High | Instructor is logged in, Course details are available, Social media account is connected | Course details, Updates, Social media account details | 1. Access the instructor's course management panel. 2. Select "Promote Course" or "Share Update". 3. Choose the social media platform for sharing. 4. Compose and schedule the promotional post or update. 5. Verify the post/update is shared on the selected platform. | Course promotion/update is shared on social media | The platform successfully shares the chosen promotion or update to the selected social media platform with relevant content and a link back to the platform. | Major | Functional | Positive

**\*\*TC\_SMI\_05\*\*** | Verify that analytics track the reach and engagement of shared content | This test case verifies the platform tracks the reach and engagement of shared content on social media, providing valuable insights for optimization and improvement. | Social Media Integration | Medium | Learner is logged in, Course completion status is available, Social media account is connected, Shared content is available | No test data needed | 1. Access the analytics dashboard or reporting section. 2. Select the "Social Media" filter. 3. View data on reach, impressions, clicks, shares, and other engagement metrics for shared content. | Analytics data is accurate and accessible | The platform provides accurate and comprehensive analytics data on the reach and engagement of shared content on social media platforms. | Major | Non-Functional | Positive

---



**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

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 Context:\*\***

Project Name: Fitness Tracker App

Project Description: 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.

User Story Title: As a user, I want to track my activities so that I can monitor my daily physical activity levels.

Acceptance Criteria:

1. The app records data on distance, duration, pace, and calories burned.
2. Users can view activity summaries.
3. Data is stored for historical analysis.
4. Activity tracking can be paused and resumed.
5. Users receive notifications for milestones.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module)

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*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 records distance, duration, pace, and calories burned for a running activity.

**\*\*Test Suite:\*\*** Activity Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in

\* GPS is enabled on the device

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Start tracking a running activity.
2. Run for a predetermined distance, duration, and pace.
3. End the activity.
4. Review the recorded data for distance, duration, pace, and calories burned.

**\*\*Postconditions:\*\*** The activity data is saved for later viewing.

**\*\*Expected Result:\*\*** The recorded data for distance, duration, pace, and calories burned should match the actual values during the activity.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*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 verifies that users can access a summary of their completed activities, including key metrics and a visual representation of their performance.

**\*\*Test Suite:\*\*** Activity Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in

\* At least one activity has been recorded

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the activity history section.
2. Select a previously recorded activity.
3. View the activity summary, which should include distance, duration, pace, and calories burned.

**\*\*Postconditions:\*\*** The activity summary is displayed correctly.

**\*\*Expected Result:\*\*** The system displays a summary of the selected activity, including all relevant data.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*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 verifies that the app stores activity data for historical analysis, allowing users to track their progress and identify trends over time.

**\*\*Test Suite:\*\*** Activity Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

\* User is logged in

\* Multiple activities have been recorded over a period of time

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the activity history section.
2. View the activity log, which should display all previously recorded activities.
3. Filter the activity log by date, activity type, or other criteria.
4. Analyze historical data to identify patterns and trends in activity levels.

**\*\*Postconditions:\*\*** Data is retrieved and displayed without errors.

**\*\*Expected Result:\*\*** The system displays a historical record of all recorded activities, allowing for easy analysis and tracking of progress.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*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 verifies that users can pause and resume activity tracking mid-session without losing data.**

**\*\*Test Suite:\*\* Activity Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

\* User is logged in

\* An activity is being tracked

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Start tracking an activity.
2. Pause the activity tracking.
3. Resume the activity tracking after a predetermined period.
4. End the activity.
5. Review the recorded data for distance, duration, pace, and calories burned.

**\*\*Postconditions:\*\* The activity data is saved correctly, including the pause and resume periods.**

**\*\*Expected Result:\*\* The system accurately records the total distance, duration, pace, and calories burned, even with pauses and resumes during the activity.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*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 verifies that the app sends notifications to users when they reach predetermined milestones during their activity tracking.

**\*\*Test Suite:\*\*** Activity Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

\* User is logged in

\* User has set a milestone within the app

**\*\*Test Data:\*\*** User-defined milestone (e.g., run 5 kilometers)

**\*\*Test Steps:\*\***

1. Start tracking a running activity.
2. Run until the defined milestone is reached.
3. Verify that a notification is received on the device.

**\*\*Postconditions:\*\*** The notification is received and acknowledged.

**\*\*Expected Result:\*\*** The system sends a notification to the user when they achieve the pre-defined milestone within the activity.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 77**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

GPS Tracking

**Feature description -**

Integrated GPS tracking allows users to accurately track outdoor activities and routes, providing detailed maps, elevation profiles, and location-based metrics. Users can visualize their workout routes, analyze performance data, and explore new routes for added variety and challenge.

**Userstory title -**

As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

**Acceptance criteria -**

['GPS tracking accurately records route data.', 'Users can view detailed maps and elevation profiles.', 'Performance data includes location-based metrics.', 'Routes can be saved and revisited.', 'Notifications are provided for deviations from planned routes.']

**Generated Test Cases:**

## Test Cases for Fitness Tracker App - GPS Tracking Feature

**Project Name:** Fitness Tracker App

**Feature Name:** GPS Tracking

**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:**

- GPS tracking accurately records route data.**
- 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:** (Provide a unique identifier for each test case)

**Test Case Title:** (Must start with "Verify that..." and clearly state the feature/function being tested)

**Test Case Description:** (A brief description of the test case, between 70-100 words)

**Test Suite:** GPS Tracking

**Test Priority:** (Priority level: High, Medium, Low)

**Preconditions:**

- User is logged in and has granted GPS permissions.
- GPS is enabled on the device.

**Test Data:** (Specify data required for execution; if none, state "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, in unordered list)

**Expected Result:** (Overall expected outcome of the test case)

**Severity:** (Blocker, Critical, Major, Minor, Trivial)

**Type of Testing:** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**Test Case Approach:** (Positive, Negative, Destructive)

**Test Case 1:**

**\*\*Test Case ID:\*\* TC\_GPS\_01**

**\*\*Test Case Title:\*\* Verify that GPS tracking accurately records route data.**

**\*\*Test Case Description:\*\* This test case verifies that the GPS tracking feature accurately records the user's route during an outdoor activity, including start and end points, distance traveled, and time taken.**

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Initiate a workout session using the GPS tracking feature.
2. Walk or run a predefined route for a specified duration.
3. End the workout session.

**\*\*Postconditions:\*\***

- The recorded route data is saved in the app.
- The user can view the recorded route on a map within the app.

**\*\*Expected Result:\*\* The GPS tracking feature accurately records the route data, including start and end points, distance traveled, and time taken. The recorded route is displayed accurately on the map within the app.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC\_GPS\_02**

**\*\*Test Case Title:\*\* Verify that users can view detailed maps and elevation profiles.**

**\*\*Test Case Description:\*\* This test case verifies that users can view detailed maps of their recorded routes, including elevation changes and gradients.**

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\* A previously recorded route with elevation changes.**

**\*\*Test Steps:\*\***

1. View a previously recorded workout session with GPS tracking.
2. Access the map view of the recorded route.
3. Verify that the map displays elevation changes and gradients along the route.

**\*\*Postconditions:\*\***

- User can zoom in and out of the map.
- User can view elevation profiles in different formats (e.g., graph, table).

**\*\*Expected Result:\*\* The app displays a detailed map of the recorded route, including elevation changes and gradients.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC\_GPS\_03**

**\*\*Test Case Title:\*\* Verify that performance data includes location-based metrics.**

**\*\*Test Case Description:\*\* This test case verifies that the GPS tracking feature records location-based metrics, including speed, pace, altitude, and distance traveled.**

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Initiate a workout session using the GPS tracking feature.
2. Walk or run a predefined route.
3. End the workout session.

**\*\*Postconditions:\*\***

- The recorded performance data is saved in the app.
- The user can view the performance data in detail.

**\*\*Expected Result:\*\*** The GPS tracking feature records location-based metrics such as speed, pace, altitude, and distance traveled accurately.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_GPS\_04

**\*\*Test Case Title:\*\*** Verify that routes can be saved and revisited.

**\*\*Test Case Description:\*\*** This test case verifies that the app allows users to save their recorded routes and revisit them later for future reference or navigation.

**\*\*Test Priority:\*\*** Medium

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Initiate a workout session using the GPS tracking feature.
2. Walk or run a predefined route.
3. End the workout session.
4. Save the recorded route.
5. Access the saved routes list.
6. Select the saved route and view it on the map.

**\*\*Postconditions:\*\***

- The saved route is added to the list of saved routes.
- The user can access and view the saved route on the map.

**\*\*Expected Result:\*\*** The app allows users to save and revisit their recorded routes. Saved routes are easily accessible and can be viewed on the map for future reference or navigation.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_GPS\_05

**\*\*Test Case Title:\*\*** Verify that notifications are provided for deviations from planned routes.

**\*\*Test Case Description:\*\*** This test case verifies that the app provides notifications to users when they deviate significantly from a planned route during a workout session.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** A planned route.

**\*\*Test Steps:\*\***

1. Initiate a workout session using the GPS tracking feature.
2. Set a planned route.
3. Intentionally deviate from the planned route.

**\*\*Postconditions:\*\***

- The app provides a notification about the deviation from the planned route.
- The user can view the updated route on the map.

**\*\*Expected Result:\*\*** The app provides timely notifications when the user deviates from the planned route, keeping them informed and allowing them to adjust their course if necessary.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Negative

---

## **Data - 78**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Heart Rate Monitoring

### **Feature description -**

The app monitors heart rate in real-time using built-in sensors or compatible wearable devices, providing insights into exercise intensity, recovery, and cardiovascular health. Heart rate data helps users optimize their workouts, maintain target heart rate zones, and track improvements in fitness over time.

### **Userstory title -**

As a user, I want to monitor my heart rate so that I can optimize my workouts and track my cardiovascular health.

### **Acceptance criteria -**

['Heart rate data is recorded in real-time.', 'Users can set and monitor target heart rate zones.', 'Historical heart rate data is stored and viewable.', 'Alerts are provided for abnormal heart rate readings.', 'Data is compatible with various wearable devices.']

### **Generated Test Cases:**

## Test Cases for Heart Rate Monitoring Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Heart Rate Monitoring**

**\*\*User Story 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:\*\***

1. Heart rate data is recorded in real-time.
2. Users can set and monitor target heart rate zones.
3. Historical heart rate data is stored and viewable.
4. Alerts are provided for abnormal heart rate readings.
5. Data is compatible with various wearable devices.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\* (Provide a unique identifier for each test case)**

**\*\*Test Case Title:\*\* (Must start with "Verify that..." and clearly state the feature/function being tested)**

**\*\*Test Case Description:\*\* (A brief description of the test case, between 70-100 words)**

**\*\*Test Suite:\*\* (Name of the test suite or module)**

**\*\*Test Priority:\*\* (Priority level: High, Medium, Low)**

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** HR-01

**\*\*Test Case Title:\*\*** Verify that heart rate data is recorded in real-time.

**\*\*Test Case Description:\*\*** This test case verifies that the app accurately captures and displays heart rate data



in real-time while a user is engaging in physical activity.

**\*\*Test Suite:\*\* Heart Rate Monitoring**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.
- App is connected to a compatible wearable device or built-in sensor.
- User initiates a workout session.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Start a workout session on the app.
2. Observe the heart rate display in real-time as the user performs an activity.
3. Note the heart rate value every 5 seconds.
4. Verify that the heart rate data on the app screen matches the heart rate readings from the wearable device (if applicable) or built-in sensor.

**\*\*Postconditions:\*\* End the workout session on the app.**

**\*\*Expected Result:\*\* The heart rate data on the app screen should closely match the actual heart rate readings from the wearable device or built-in sensor, updating in real-time as the user's heart rate fluctuates.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* HR-02**

**\*\*Test Case Title:\*\* Verify that users can set and monitor target heart rate zones.**

**\*\*Test Case Description:\*\* This test case validates that users can define custom target heart rate zones within the app and track their progress within those zones during exercise.**

**\*\*Test Suite:\*\* Heart Rate Monitoring**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.

**\*\*Test Data:\*\* Target heart rate zones (e.g., Fat Burn, Cardio, Peak).**

**\*\*Test Steps:\*\***

1. Navigate to the heart rate settings in the app.
2. Set target heart rate zones by entering the minimum and maximum heart rate values for each zone.
3. Start a workout session.
4. Observe the heart rate display during the workout and note if the current heart rate falls within the set target

zone.

5. Verify that the app provides visual indicators or notifications when the heart rate exceeds or falls below the target zone boundaries.

**\*\*Postconditions:\*\*** End the workout session on the app.

**\*\*Expected Result:\*\*** The app should accurately display the user's heart rate in relation to the defined target zones. Visual cues or notifications should inform the user when their heart rate goes beyond the set boundaries.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** 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 heart rate data from past workouts and provides users with a comprehensive history of their heart rate readings.

**\*\*Test Suite:\*\*** Heart Rate Monitoring

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in.
- User has completed at least one workout session with heart rate data recorded.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the heart rate history section in the app.
2. Review the stored heart rate data from previous workouts.
3. Verify that the date, time, and duration of each workout session are displayed correctly.
4. Check that the heart rate data points are displayed visually, allowing for analysis of trends and progress over time.

**\*\*Postconditions:\*\* No postconditions needed.**

**\*\*Expected Result:\*\*** The app should successfully display the user's historical heart rate data for each recorded workout session, including date, time, duration, and a clear visual representation of the heart rate readings.

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** HR-04

**\*\*Test Case Title:\*\*** Verify that alerts are provided for abnormal heart rate readings.

**\*\*Test Case Description:\*\*** This test case tests the app's ability to detect and alert the user to any abnormal heart rate readings that may indicate potential health issues.

**\*\*Test Suite:\*\*** Heart Rate Monitoring

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- App is connected to a compatible wearable device or built-in sensor.

**\*\*Test Data:\*\*** Simulated abnormal heart rate readings (e.g., extremely high or low heart rate).

**\*\*Test Steps:\*\***

1. Simulate an abnormal heart rate reading using a wearable device or by manually entering data into the app.
2. Monitor the app's response and observe if it triggers an alert notification.
3. Verify that the alert notification includes relevant information about the abnormal reading and suggests appropriate actions.

**\*\*Postconditions:\*\*** Disable the simulated abnormal heart rate readings.

**\*\*Expected Result:\*\*** The app should detect the simulated abnormal heart rate reading and issue a clear and timely alert notification to the user, providing relevant information and guidance.

**\*\*Severity:\*\*** Critical

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Negative

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** HR-05

**\*\*Test Case Title:\*\*** Verify that data is compatible with various wearable devices.

**\*\*Test Case Description:\*\*** This test case validates the app's compatibility with a variety of popular wearable fitness trackers and ensures seamless data synchronization and transfer between the app and the connected device.

**\*\*Test Suite:\*\*** Heart Rate Monitoring

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.
- App is installed on a compatible device (e.g., smartphone, tablet).
- Multiple compatible wearable devices are available (e.g., Fitbit, Apple Watch, Garmin).

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Connect the app to a compatible wearable device.
2. Start a workout session on the app and monitor the heart rate readings from the connected device.
3. Verify that the heart rate data is synchronized between the app and the wearable device in real-time.
4. Repeat steps 1-3 with different compatible wearable devices.

**\*\*Postconditions:\*\*** Disconnect all wearable devices from the app.

**\*\*Expected Result:\*\*** The app should successfully connect to and synchronize heart rate data with various compatible wearable devices, ensuring a seamless data flow between the app and the connected devices.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Compatibility

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 79**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Calorie Tracking

**Feature description -**

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

**Userstory title -**

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

**Acceptance criteria -**

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

**Generated Test Cases:**

## Fitness Tracker App - Calorie Tracking Test Cases

**\*\*Project Context:\*\***

**\* \*\*Project Name:\*\*** Fitness Tracker App

**\* \*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\* \*\*Feature Name:\*\*** Calorie Tracking

**\* \*\*Feature Description:\*\*** Users can log their daily food intake and track calorie consumption against calorie



expenditure from physical activities.

**User Story 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:**

- 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 Creation Template:**

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data |  
Test Steps | Postconditions | Expected Result | Severity | Type of Testing | Test Case Approach |

|---|---|---|---|---|---|---|---|---|---|---|---|

| TC\_CT\_01 | Verify that users can log daily food intake and calorie consumption | This test case ensures that the user can log their food intake and the corresponding calorie count for the day. | Calorie Tracking | High | User is logged in | Breakfast: Oatmeal (150 calories), Lunch: Chicken Salad (350 calories), Dinner: Salmon with Vegetables (400 calories) | 1. Go to the food log section. 2. Add a new food entry. 3. Enter the food name (e.g., Oatmeal). 4. Enter the calorie count (e.g., 150). 5. Save the entry. | Food log updated with the entered data | The system successfully logs the food entry and the associated calorie count. | Major | Functional | Positive |

| TC\_CT\_02 | Verify that the app calculates and displays total calories consumed and burned | This test case verifies the app's ability to accurately calculate and display total calories consumed and burned based on logged food intake and tracked physical activity. | Calorie Tracking | High | User is logged in, user has logged food intake and tracked exercise | Food log: 1500 calories, Exercise log: 500 calories burned | 1. Go to the calorie tracker overview. 2. Observe the displayed total calorie consumption and expenditure. | Total calories

consumed and burned are displayed | The system accurately calculates and displays the total calories consumed and burned based on the logged data. | Major | Functional | Positive |

| TC\_CT\_03 | Verify that users can set daily calorie goals and track progress | This test case checks the functionality of setting daily calorie goals and tracking progress towards those goals. | Calorie Tracking | High | User is logged in | Daily calorie goal: 1800 | 1. Go to the settings section. 2. Set a daily calorie goal (e.g., 1800 calories). 3. Observe the progress bar for the day. | Daily calorie goal is set and the progress bar reflects the current progress | The system successfully sets the daily calorie goal and displays the progress bar with accurate information. | Major | Functional | Positive |

| TC\_CT\_04 | Verify that food items and calorie information are searchable in the database | This test case ensures that users can efficiently search for food items and retrieve their corresponding calorie information from the database. | Calorie Tracking | Medium | User is logged in | Food item: Chicken Breast | 1. Go to the food search bar. 2. Enter a food item (e.g., "Chicken Breast"). 3. Observe the search results. | Search results display matching food items and their calorie information | The system displays a list of relevant food items with their calorie information based on the search query. | Minor | Functional | Positive |

| TC\_CT\_05 | Verify that users receive alerts for exceeding calorie goals | This test case checks the alert system for exceeding daily calorie goals, ensuring that users receive timely notifications to help them stay within their calorie limits. | Calorie Tracking | High | User is logged in, User has set a daily calorie goal, User has consumed more calories than the set goal | Daily calorie goal: 1500, Calories consumed: 1800 | 1. Consume calories exceeding the daily goal. 2. Observe the app's behavior. | An alert or notification is displayed notifying the user of exceeding their calorie goal. | Major | Functional | 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**

**\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\*\*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.

**\*\*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:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module) - Sleep Tracking

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing,

End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** ST-001

**\*\*Test Case Title:\*\*** Verify that sleep data is recorded including duration and quality.

**\*\*Test Case Description:\*\*** This test case verifies that the app accurately records sleep duration and quality based on accelerometer and heart rate sensor data.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in and has granted the app permission to access accelerometer and heart rate data.
- User has initiated sleep tracking mode.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Start sleep tracking mode in the app.
2. Sleep for a predetermined duration (e.g., 8 hours).
3. Stop sleep tracking mode.
4. Access the sleep data log.

**\*\*Postconditions:\*\***

- Sleep tracking is stopped.

**\*\*Expected Result:\*\*** The app records accurate sleep duration, and the sleep quality is displayed with details like sleep stages (light, deep, REM).

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* ST-002**

**\*\*Test Case Title:\*\* Verify that users can view detailed sleep stages and patterns.**

**\*\*Test Case Description:\*\* This test case validates the ability of the app to display detailed information about sleep stages and patterns over time.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in and has recorded sleep data.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the sleep tracking section of the app.
2. Access the sleep data log.
3. Select a specific sleep session.
4. View the sleep stages breakdown (e.g., light, deep, REM) and pattern visualization (graph or chart).

**\*\*Postconditions:\*\* None.**

**\*\*Expected Result:\*\* The app displays a detailed breakdown of sleep stages and patterns for the selected sleep session.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* ST-003**

**\*\*Test Case Title:\*\*** Verify that the app provides insights into sleep disturbances.

**\*\*Test Case Description:\*\*** This test case ensures that the app identifies and highlights potential sleep disturbances based on recorded data.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in and has recorded sleep data with potential disturbances (e.g., awakenings, movements, snoring).

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Go to the sleep tracking section of the app.
2. Access the sleep data log.
3. Select a specific sleep session.
4. Observe the sleep data for any identified disturbances and their descriptions (e.g., awakenings, movements, snoring).

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** The app identifies and highlights potential sleep disturbances within the sleep data, providing relevant information about their frequency, duration, and potential causes.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** ST-004

**\*\*Test Case Title:\*\*** Verify that users receive recommendations for improving sleep quality.

**\*\*Test Case Description:\*\*** This test case confirms that the app provides personalized recommendations based on recorded sleep data to enhance sleep quality.

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in and has recorded sleep data.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the sleep tracking section of the app.
2. Access the sleep data log.
3. View the insights and recommendations section for improving sleep quality.

**\*\*Postconditions:\*\* None.**

**\*\*Expected Result:\*\*** The app provides personalized recommendations based on user sleep data, such as adjusting bedtime routines, creating a relaxing sleep environment, or limiting caffeine intake before bed.

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* ST-005**

**\*\*Test Case Title:\*\* Verify that sleep data can be viewed in daily, weekly, and monthly summaries.**

**\*\*Test Case Description:\*\*** This test case checks that the app allows users to view aggregated sleep data in various timeframes.

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in and has recorded sleep data over an extended period.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Go to the sleep tracking section of the app.



2. Select the "Summary" or "Analytics" view.
3. Choose the desired timeframe (daily, weekly, or monthly).
4. View the aggregated sleep data summary.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** The app displays a summarized view of sleep data for the selected timeframe, showing average sleep duration, sleep quality, and potential trends.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 81**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Workout Logging

### **Feature description -**

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

### **Userstory title -**

As a user, I want to log my workouts so that I can track my progress and improve my performance.

**Acceptance criteria -**

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

**Generated Test Cases:**

**## Test Cases for Fitness Tracker App - Workout Logging**

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\*\*Feature Name:\*\* Workout Logging**

**\*\*Feature Description:\*\*** Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments.

**\*\*User Story Title:\*\*** As a user, I want to log my workouts so that I can track my progress and improve my performance.

**\*\*Acceptance Criteria:\*\***

1. Users can log various workout types with details.
2. Workout logs include duration, intensity, and notes.
3. Users can edit and delete workout entries.
4. Data is stored and viewable for tracking progress.
5. Users can filter logs by date and workout type.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Workout Logging

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

- User is logged in
- Workout logging feature is enabled

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "No test data needed")

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\***

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** FT-WL-TC-01

**\*\*Test Case Title:\*\*** Verify that users can log various workout types with details.

**\*\*Test Case Description:\*\*** This test case verifies that the user can log different workout types, specifying exercise type, duration, intensity, and additional notes.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\***

- Workout Type: Running, Cycling, Yoga, Strength Training
- Duration: 30 minutes, 1 hour, 45 minutes
- Intensity: Light, Moderate, High
- Notes: "Ran on treadmill", "Outdoor cycling", "Focused on core strength"

**\*\*Test Steps:\*\***

1. Navigate to the workout logging section.
2. Select a workout type from the available options (e.g., Running, Cycling, Yoga, Strength Training).
3. Enter the duration of the workout.
4. Select the intensity level.
5. Add optional notes describing the workout.
6. Save the workout log.

**\*\*Postconditions:\*\***

- Workout log is saved successfully.

**\*\*Expected Result:\*\*** The system should successfully create a new workout log entry with the specified details, including workout type, duration, intensity, and notes.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** FT-WL-TC-02

**\*\*Test Case Title:\*\*** Verify that workout logs include duration, intensity, and notes.

**\*\*Test Case Description:\*\*** This test case checks that all required information (duration, intensity, and notes) is captured and displayed for each workout log entry.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Log a workout with details for duration, intensity, and notes.
2. View the saved workout log entry.

**\*\*Postconditions:\*\***

- Workout log entry is displayed.

**\*\*Expected Result:\*\*** The workout log entry should display the duration, intensity, and notes entered by the user.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** FT-WL-TC-03

**\*\*Test Case Title:\*\*** Verify that users can edit and delete workout entries.

**\*\*Test Case Description:\*\*** This test case checks if users can modify or remove existing workout log entries.

**\*\*Test Priority:\*\*** Medium

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Log a workout.
2. Edit the workout log entry, changing the duration, intensity, or notes.
3. Save the changes.
4. Delete the workout log entry.

**\*\*Postconditions:\*\***

- Edited workout log entry is saved.
- Deleted workout log entry is removed.

**\*\*Expected Result:\*\*** The system should allow users to edit and delete workout log entries successfully, updating or removing the respective entries from the log.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* FT-WL-TC-04**

**\*\*Test Case Title:\*\* Verify that data is stored and viewable for tracking progress.**

**\*\*Test Case Description:\*\* This test case ensures that workout log data is persistently stored and can be accessed for progress tracking.**

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Log multiple workouts over several days.
2. View the workout log history, including past entries.

**\*\*Postconditions:\*\***

- Workout log history is displayed.

**\*\*Expected Result:\*\* The system should store all workout log entries and allow the user to access and view them for tracking progress over time.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* FT-WL-TC-05**

**\*\*Test Case Title:\*\* Verify that users can filter logs by date and workout type.**

**\*\*Test Case Description:\*\* This test case ensures that users can filter their workout log entries based on date and workout type, enabling easier analysis and tracking.**

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Log several workouts of different types across multiple days.
2. Filter the workout log by a specific date range (e.g., last week).
3. Filter the workout log by a specific workout type (e.g., Running).
4. Filter the workout log by both date and workout type.

**\*\*Postconditions:\*\***

- Filtered workout logs are displayed.

**\*\*Expected Result:\*\*** The system should successfully filter the workout log entries based on the selected date range, workout type, or both.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 82**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Goal Setting

**Feature description -**

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

#### **Userstory title -**

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

#### **Acceptance criteria -**

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

#### **Generated Test Cases:**

## Test Cases for Fitness Tracker App - Goal Setting Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\*\*Feature Name:\*\* Goal Setting**

**\*\*Feature Description:\*\*** Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives.

**\*\*User Story Title:\*\*** 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 Creation Template:\*\*

```

**Test Case ID:** | **Test Case Title:** | **Test Case Description:** | **Test Suite:** | **Test Priority:** |
**Preconditions:** | **Test Data:** | **Test Steps:** | **Postconditions:** | **Expected Result:** |
**Severity:** | **Type of Testing:** | **Test Case Approach:**

```

Test Case ID	Description	Preconditions	Test Data	Steps	Expected Results	Priority	Type	Status		
**TC-GS-01**	Verify that users can set goals for various fitness metrics.	This test case verifies that users can set goals for different fitness metrics like steps, calories burned, distance traveled, and heart rate.	Goal Setting	High	- User is logged in to the app - No test data needed	1. Navigate to the goal setting section. 2. Select a fitness metric (e.g., steps). 3. Enter a target value for the selected metric. 4. Save the goal.	- Goal is saved in the app. The system should successfully save the user's goal for the selected fitness metric and allow them to set goals for different metrics.	Major	Functional	Positive

**\*\*TC-GS-02\*\*** | Verify that progress tracking is available for each goal. | This test case ensures that progress tracking is available for each set goal, allowing users to monitor their progress towards achieving their goals. |

Goal Setting | High | - User is logged in to the app. - A goal is set for a fitness metric. | - No test data needed |

1. Navigate to the goal tracking section. 2. Select a set goal from the list. 3. Verify that the progress tracker displays the current progress towards the goal. | - Progress tracker shows current progress. | The system should display a clear and updated progress tracker for each goal, visually demonstrating the user's progress towards achieving their goals. | Major | Functional | Positive

**\*\*TC-GS-03\*\*** | Verify that users receive motivational feedback and reminders. | This test case validates that the app provides motivational feedback and reminders to users based on their progress and goal settings. |

Goal Setting | Medium | - User is logged in to the app. - A goal is set for a fitness metric. | - No test data needed | 1. Set a goal for a specific fitness metric. 2. Track progress towards the goal for a few days. 3. Observe the app's motivational messages and reminders. | - App provides motivational feedback and reminders. | The system should provide timely and encouraging feedback to users based on their progress, and

send reminders to keep them motivated to reach their goals. | Minor | Functional | Positive

**\*\*TC-GS-04\*\*** | Verify that goals can be adjusted or updated as needed. | This test case ensures that users can modify or update their set goals based on their evolving fitness needs and progress. | Goal Setting | Medium | - User is logged in to the app. - A goal is set for a fitness metric. | - No test data needed | 1. Navigate to the goal settings section. 2. Select an existing goal. 3. Change the target value or the selected fitness metric. 4. Save the updated goal. | - Goal is updated in the app. | The system should allow users to easily adjust or update their existing goals, enabling them to adapt their fitness plans as needed. | Minor | Functional | Positive

**\*\*TC-GS-05\*\*** | Verify that the app provides a summary of goal attainment. | This test case tests the functionality of providing a summary of the user's goal attainment, highlighting their accomplishments and progress. | Goal Setting | Low | - User is logged in to the app. - A goal is set for a fitness metric. - The goal is achieved. | - No test data needed | 1. Achieve a set goal for a specific fitness metric. 2. Navigate to the goal summary section. 3. Observe the summary of the achieved goal. | - App provides a summary of the achieved goal. | The system should provide a clear and concise summary of the user's goal attainment, showcasing their progress and accomplishments. | Minor | Functional | Positive

**\*\*Note:\*\*** These are just examples. You may need to add more test cases to cover all possible scenarios and edge cases.

---

**Data - 83**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

### **Feature name -**

Workout Plans

### **Feature description -**

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

### **Userstory title -**

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

### **Acceptance criteria -**

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

### **Generated Test Cases:**

## Test Cases for Fitness Tracker App - Workout Plans Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\*\*Feature Name:\*\* Workout Plans**

**\*\*Feature Description:\*\*** The app offers pre-designed workout plans tailored to different fitness levels, goals, and preferences.

**\*\*User Story Title:\*\*** 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:\*\*** Workout Plans

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the app

**\*\*Test Data:\*\***

- User profile with specified fitness level (e.g., Beginner, Intermediate, Advanced)

**\*\*Test Steps:\*\***

-

**\*\*Postconditions:\*\***

-

**\*\*Expected Result:\*\***

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_WP\_01

**\*\*Test Case Title:\*\*** Verify that users can select workout plans based on fitness levels.

**\*\*Test Case Description:\*\*** This test case verifies that the app allows users to choose workout plans based on their self-selected fitness level.

**\*\*Test Data:\*\*** User profile with fitness level set to "Beginner".

**\*\*Test Steps:\*\***

1. Navigate to the "Workout Plans" section.
2. Filter workout plans by "Beginner" fitness level.
3. Verify that only plans intended for beginners are displayed.

**\*\*Expected Result:\*\*** Only workout plans suitable for beginner fitness levels should be displayed in the results.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_WP\_02

**\*\*Test Case Title:\*\*** Verify that workout plans include detailed exercise routines and schedules.

**\*\*Test Case Description:\*\*** This test case ensures that selected workout plans contain detailed exercise routines and schedules, specifying exercises, sets, reps, and workout days.

**\*\*Test Data:\*\*** User profile with any fitness level and selected workout plan.

**\*\*Test Steps:\*\***

1. Select a workout plan from the list.
2. Verify that the selected plan displays a detailed schedule with specific exercises, sets, reps, and workout days.

3. Check that the exercise descriptions are clear and understandable.

**\*\*Expected Result:\*\*** The selected workout plan should provide a clear and detailed schedule with all necessary information for each workout session.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_WP\_03

**\*\*Test Case Title:\*\*** Verify that users can track adherence and progress within the plan.

**\*\*Test Case Description:\*\*** This test case ensures that the app tracks the user's adherence to the workout plan and allows them to monitor their progress.

**\*\*Test Data:\*\*** User profile with selected workout plan, completed workout sessions.

**\*\*Test Steps:\*\***

1. Complete a workout session as per the plan.
2. Verify that the app logs the completed workout and updates adherence metrics.
3. Check that progress metrics (e.g., days completed, exercises completed) are updated and displayed in the plan interface.

**\*\*Expected Result:\*\*** The app should record completed workouts, update progress metrics, and display adherence information in the plan interface.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_WP\_04

**\*\*Test Case Title:\*\*** Verify that the app provides notifications and reminders for workouts.

**\*\*Test Case Description:\*\*** This test case verifies that the app sends timely notifications and reminders to

users about their upcoming workout sessions.

**\*\*Test Data:\*\*** User profile with selected workout plan and enabled notifications.

**\*\*Test Steps:\*\***

1. Configure the app to send notifications for upcoming workouts.
2. Check that the user receives notifications at the scheduled time for each workout session.
3. Verify that the notifications provide relevant information about the workout, such as exercise type, duration, and plan name.

**\*\*Expected Result:\*\*** The app should send notifications or reminders to users about their upcoming workouts at the scheduled time.

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_WP\_05

**\*\*Test Case Title:\*\*** Verify that users can adjust plans based on progress and preferences.

**\*\*Test Case Description:\*\*** This test case verifies that users can modify their workout plans based on their progress and preferences, such as adjusting exercises, intensity, or schedule.

**\*\*Test Data:\*\*** User profile with selected workout plan and completed sessions.

**\*\*Test Steps:\*\***

1. Complete several sessions of the selected workout plan.
2. Based on progress and preferences, modify the plan by adding or removing exercises, changing the workout intensity, or adjusting the schedule.
3. Verify that the changes are reflected in the plan's schedule and progress tracking.

**\*\*Expected Result:\*\*** Users should be able to modify the workout plan according to their progress and preferences, and these changes should be reflected in the plan's schedule and progress tracking.

**\*\*Test Case Approach:\*\*** Positive

---

## Data - 84

### Project name -

Fitness Tracker App

### Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### Feature name -

Community Support

### Feature description -

Users can connect with a community of like-minded individuals, share progress updates, participate in challenges, and offer support and encouragement to fellow users. Community support fosters accountability, motivation, and a sense of camaraderie among users striving towards similar fitness goals.

### Userstory title -

As a user, I want to connect with a community so that I can share my progress and stay motivated.

### Acceptance criteria -

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.', 'The app facilitates participation in community challenges.', 'Users can offer and receive support and encouragement.', 'Community activity and engagement are tracked.']

### Generated Test Cases:

## Test Cases for Fitness Tracker App - Community Support Feature

\*\*Project Name:\*\* Fitness Tracker App



**\*\*Feature Name:\*\* Community Support**

**\*\*User Story Title:\*\* 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 | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data |  
Test Steps | Postconditions | Expected Result | Severity | Type of Testing | Test Case Approach |

|---|---|---|---|---|---|---|---|---|---|---|---|

| TC\_CS\_01 | Verify that users can join and participate in community groups | This test case aims to ensure that users can successfully join and participate in community groups within the app. | Community Support | High | - User is logged in. | - Sample community groups (e.g., "Running Enthusiasts", "Weight Loss Journey").  
| 1. Navigate to the "Community" section. <br> 2. Browse available groups. <br> 3. Select a group to join.  
<br> 4. Verify successful group membership. | - User is added to the group. <br> - User can view group posts and participate in discussions. | The system successfully adds the user to the chosen group, allowing them to access and interact with group content. | Major | Functional | Positive |

| TC\_CS\_02 | Verify that users can share progress updates and achievements | This test case validates the ability of users to share their progress updates and achievements within community groups. | Community

Support | High | - User is logged in. <br> - User is a member of a community group. | - Sample progress updates (e.g., "Completed a 5k run!", "Hit a new personal best in weightlifting"). | 1. Navigate to a community group. <br> 2. Access the "Share Progress" or "Post" option. <br> 3. Enter a progress update or achievement. <br> 4. Publish the update. | - Progress update is visible to other group members. | The system successfully posts the user's progress update or achievement within the group, making it visible to other members. | Major | Functional | Positive |

| TC\_CS\_03 | Verify that the app facilitates participation in community challenges | This test case examines the functionality of the app in facilitating participation in community challenges. | Community Support | High | - User is logged in. <br> - User is a member of a community group. | - Sample challenges (e.g., "Step Challenge", "Workout Streak"). | 1. Navigate to a community group. <br> 2. Access the "Challenges" section. <br> 3. Select a challenge to participate in. <br> 4. Accept the challenge. | - User is enrolled in the challenge. <br> - Challenge progress is tracked. | The system successfully enrolls the user in the chosen challenge, tracks their progress, and displays relevant challenge information. | Major | Functional | Positive |

| TC\_CS\_04 | Verify that users can offer and receive support and encouragement | This test case verifies the ability of users to offer and receive support and encouragement within community groups. | Community Support | Medium | - User is logged in. <br> - User is a member of a community group. | - Sample support messages (e.g., "Great work!", "Keep pushing!", "You've got this!"). | 1. Navigate to a community group. <br> 2. View other members' posts and progress updates. <br> 3. Offer support or encouragement through comments. <br> 4. Verify the support message is displayed. | - Support message is visible to the recipient. <br> - User can receive support messages from other members. | The system successfully allows users to exchange support and encouragement within the group, fostering a positive and motivating environment. | Minor | Functional | Positive |

| TC\_CS\_05 | Verify that community activity and engagement are tracked | This test case assesses the functionality of tracking community activity and engagement metrics. | Community Support | Medium | - User is logged in. <br> - User is a member of a community group. | - No specific data required. | 1. Navigate to a community group. <br> 2. Observe the "Activity" or "Engagement" section. <br> 3. Verify that the metrics are displayed. <br> 4. Check that the metrics reflect recent activity. | - Activity and engagement metrics are

accurately displayed. | The system successfully tracks and displays community activity and engagement metrics, providing insights into group participation and dynamics. | Minor | Functional | 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 - Integration with Wearable Devices

### \*\*Test Case 1:\*\*

\*\*Test Case ID:\*\* FT-WD-01

\*\*Test Case Title:\*\* Verify that the app supports integration with multiple wearable devices

\*\*Test Case Description:\*\* This test case verifies that the fitness tracker app can successfully connect and sync data with different popular wearable devices. The app should be compatible with at least two popular fitness trackers, one smartwatch, and one heart rate monitor.

\*\*Test Suite:\*\* Integration with Wearable Devices

\*\*Test Priority:\*\* High

#### \*\*Preconditions:\*\*

- \* The user has a compatible wearable device.
- \* The user has installed the Fitness Tracker app on their mobile device.

#### \*\*Test Data:\*\*

- \* Fitness tracker (e.g., Fitbit, Garmin)
- \* Smartwatch (e.g., Apple Watch, Samsung Galaxy Watch)
- \* Heart rate monitor (e.g., Polar, Wahoo)

#### \*\*Test Steps:\*\*

1. Open the Fitness Tracker app and access the device connection settings.
2. Attempt to connect the fitness tracker to the app.
3. Attempt to connect the smartwatch to the app.
4. Attempt to connect the heart rate monitor to the app.

#### \*\*Postconditions:\*\*

- \* The app successfully connects with each device.
- \* The app displays the connected device in the settings.

\*\*Expected Result:\*\* The app successfully establishes a connection and syncs data with all three wearable

devices.

**\*\*Severity:\*\*** Blocker

**\*\*Type of Testing:\*\*** Integration Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** FT-WD-02

**\*\*Test Case Title:\*\*** Verify that users can sync activity data from their wearables

**\*\*Test Case Description:\*\*** This test case verifies that the fitness tracker app can successfully retrieve and sync activity data from connected wearable devices. The app should sync data like steps, distance, calories burned, and heart rate.

**\*\*Test Suite:\*\*** Integration with Wearable Devices

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- \* The user has a compatible wearable device connected to the app.
- \* The user has performed physical activity while wearing the device.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker app.
2. Access the activity data section (e.g., steps, distance, calories burned, heart rate).
3. Compare the activity data displayed on the app with the data recorded on the wearable device.

**\*\*Postconditions:\*\***

- \* The app displays the activity data synced from the wearable device.

**\*\*Expected Result:\*\*** The app successfully retrieves and syncs the activity data from the connected wearable device, displaying it in the app.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* FT-WD-03**

**\*\*Test Case Title:\*\* Verify that the app provides real-time data updates from wearables**

**\*\*Test Case Description:\*\*** This test case verifies that the fitness tracker app can receive and display real-time data updates from connected wearable devices. The app should provide real-time updates for heart rate, steps, and other relevant metrics.

**\*\*Test Suite:\*\* Integration with Wearable Devices**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- \* The user has a compatible wearable device connected to the app.
- \* The user is actively wearing the wearable device.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker app and access the real-time data view (e.g., live heart rate monitor).
2. Engage in physical activity while wearing the wearable device.
3. Monitor the real-time data displayed in the app.

**\*\*Postconditions:\*\***

- \* The app displays the real-time data updates received from the wearable device.

**\*\*Expected Result:\*\*** The app continuously receives and displays real-time data updates from the wearable device.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* FT-WD-04**

**\*\*Test Case Title:\*\* Verify that wearable data includes steps, heart rate, and calories burned**

**\*\*Test Case Description:\*\* This test case verifies that the fitness tracker app successfully retrieves and syncs the necessary activity data from connected wearable devices. The data should include steps taken, heart rate, and calories burned.**

**\*\*Test Suite:\*\* Integration with Wearable Devices**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- \* The user has a compatible wearable device connected to the app.
- \* The user has performed physical activity while wearing the device.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker app and access the activity data section.
2. Verify that the app displays steps taken, heart rate, and calories burned.

**\*\*Postconditions:\*\***

- \* The app displays the relevant data from the connected wearable device.

**\*\*Expected Result:\*\* The app successfully syncs and displays steps taken, heart rate, and calories burned.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* FT-WD-05**

**\*\*Test Case Title:\*\*** Verify that users can troubleshoot connectivity issues with wearables

**\*\*Test Case Description:\*\*** This test case verifies that the fitness tracker app provides helpful troubleshooting options for users experiencing connectivity issues with their wearable devices. The app should guide users through steps to resolve common connectivity problems.

**\*\*Test Suite:\*\*** Integration with Wearable Devices

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- \* The user has a compatible wearable device.
- \* The user is experiencing connectivity issues with the wearable device.

**\*\*Test Data:\*\***

- \* No test data needed.

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker app and access the device connection settings.
2. Trigger a simulated connectivity issue (e.g., disconnect the device from Bluetooth).
3. Observe the app's response and troubleshoot suggestions.

**\*\*Postconditions:\*\***

- \* The app provides helpful troubleshooting options and steps to resolve the connectivity issues.

**\*\*Expected Result:\*\*** The app provides clear troubleshooting steps and instructions to help users resolve connectivity issues with their wearable devices.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Usability Testing

**\*\*Test Case Approach:\*\*** Negative

-----



**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:**

## Test Cases for Fitness Tracker App - Water Intake Tracking

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Feature Name:\*\* Water Intake Tracking**

**\*\*User Story Title:\*\*** 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:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module) - Water Intake Tracking

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* WT-01**

**\*\*Test Case Title:\*\* Verify that users can log their daily water intake.**

**\*\*Test Case Description:\*\* This test case aims to ensure that the app allows users to record their water consumption throughout the day. It tests the functionality of logging water intake by verifying the user can input water amounts and the app stores the data.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the app.

**\*\*Test Data:\*\***

- Water intake values (e.g., 250ml, 500ml, 750ml)

**\*\*Test Steps:\*\***

1. Access the "Water Intake" section of the app.
2. Input a water intake amount (e.g., 250ml).
3. Verify that the entered water intake is recorded in the app.
4. Repeat steps 2 and 3 with different water intake amounts.

**\*\*Postconditions:\*\***

- Water intake values are logged and displayed in the app.

**\*\*Expected Result:\*\* The app successfully records the user's water intake values, displaying them in the water intake log.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* WT-02**

**\*\*Test Case Title:\*\*** Verify that the app calculates total water consumption for the day.

**\*\*Test Case Description:\*\*** This test case verifies the app's ability to accurately calculate the total water consumed throughout the day based on user-logged entries. It assesses the correct computation of cumulative intake.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User has logged water intake values throughout the day.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Log multiple water intake entries throughout the day (e.g., 250ml, 500ml, 750ml).
2. Access the "Water Intake" section and view the daily total water consumption.
3. Manually calculate the total water intake based on the logged values.
4. Compare the manually calculated total with the total displayed in the app.

**\*\*Postconditions:\*\***

- Daily total water consumption is displayed in the app.

**\*\*Expected Result:\*\*** The app accurately calculates and displays the total water consumed for the day, matching the manual calculation based on the logged entries.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** WT-03

**\*\*Test Case Title:\*\*** Verify that users can set daily hydration goals.

**\*\*Test Case Description:\*\*** This test case ensures that the app allows users to set personalized hydration goals. It verifies the functionality of setting and saving daily hydration targets.

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in to the app.

**\*\*Test Data:\*\***

- Hydration goal values (e.g., 1.5L, 2L, 2.5L)

**\*\*Test Steps:\*\***

1. Access the "Water Intake" section and navigate to the "Goal" settings.
2. Set a daily hydration goal (e.g., 2L).
3. Verify that the set goal is saved and displayed in the app.
4. Repeat steps 2 and 3 with different goal values.

**\*\*Postconditions:\*\***

- Daily hydration goal is saved and displayed in the app.

**\*\*Expected Result:\*\*** The app allows users to set and save their daily hydration goals, displaying the chosen target.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** WT-04

**\*\*Test Case Title:\*\*** Verify that reminders are provided to encourage regular water intake.

**\*\*Test Case Description:\*\*** This test case evaluates the app's reminder system to encourage regular water consumption. It verifies the functionality of scheduling and receiving reminders throughout the day.

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User has set a daily hydration goal.

**\*\*Test Data:\*\***

- Reminder frequency (e.g., hourly, every 2 hours, custom intervals)

**\*\*Test Steps:\*\***

1. Set a daily hydration goal (e.g., 2L).
2. Configure reminder settings (e.g., hourly reminders).
3. Verify that the app sends reminders at the scheduled intervals.
4. Change the reminder frequency and repeat step 3.

**\*\*Postconditions:\*\***

- Reminders are received at the configured intervals.

**\*\*Expected Result:\*\*** The app successfully sends reminders at the chosen frequency to encourage users to drink water throughout the day.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* WT-05**

**\*\*Test Case Title:\*\* Verify that users can view hydration progress and trends.**

**\*\*Test Case Description:\*\*** This test case ensures the app allows users to track their hydration progress and analyze their water intake patterns. It verifies the display of hydration progress, visualization of trends, and availability of relevant data.

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User has logged water intake values for multiple days.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Access the "Water Intake" section and navigate to the "Progress" view.
2. Verify that the app displays the user's current hydration progress for the day.
3. Explore the trend graphs and charts to analyze water intake patterns over time (e.g., weekly, monthly trends).
4. Verify that the trends are displayed in a clear and easy-to-understand manner.

**\*\*Postconditions:\*\***

- Hydration progress and trends are visualized in the app.

**\*\*Expected Result:\*\*** The app displays the user's daily hydration progress, providing relevant data and visualizations of water intake trends over time, helping users analyze their hydration patterns and track their progress toward their hydration goals.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 87**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Nutrition Tracking

**Feature description -**

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

**Userstory title -**

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

**Acceptance criteria -**

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

**Generated Test Cases:**

## Test Cases for Fitness Tracker App: Nutrition Tracking

**Project Name:** Fitness Tracker App

**Project Description:** 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.

**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.

**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 Creation Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Nutrition Tracking

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\***

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\* TC\_NT\_01**

**\*\*Test Case Title:\*\* Verify that users can log their daily meals and snacks.**

**\*\*Test Case Description:\*\* This test case ensures that users can add meals and snacks to their food diary, including details like food type, quantity, and time of consumption.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged into the app.

**\*\*Test Data:\*\***

- Breakfast: Oatmeal, 1 cup
- Lunch: Chicken Salad Sandwich, 1 sandwich
- Snack: Apple, 1 medium
- Dinner: Grilled Salmon, 4 oz, Baked Potato, 1 medium

**\*\*Test Steps:\*\***

1. Navigate to the "Nutrition" or "Food Diary" section of the app.
2. Tap the "Add Meal" or "Log Food" button.
3. Select the meal type (breakfast, lunch, snack, dinner).
4. Search for and select the food items from the food database.
5. Enter the quantity consumed for each item.
6. Add notes or comments about the meal (optional).
7. Save the meal log.

**\*\*Postconditions:\*\***

- The meal and snack entries are saved in the user's food diary.

**\*\*Expected Result:\*\* The app successfully records the meal and snack entries with all details provided, and the entries appear in the user's food diary.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_NT\_02

**\*\*Test Case Title:\*\*** Verify that the app calculates macronutrient intake (carbs, fats, proteins).

**\*\*Test Case Description:\*\*** This test case validates that the app correctly calculates the total macronutrient intake for each meal and the entire day based on the logged food items.

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User has logged at least one meal.

**\*\*Test Data:\*\***

- See Test Case 1 (meal data)

**\*\*Test Steps:\*\***

1. Navigate to the "Nutrition" or "Food Diary" section of the app.
2. View the summary for a specific meal or the entire day.
3. Check the displayed macronutrient intake (carbs, fats, proteins) for the selected timeframe.

**\*\*Postconditions:\*\***

- No postconditions.

**\*\*Expected Result:\*\*** The app accurately displays the calculated macronutrient intake for the specified period, matching the nutritional values of the logged food items.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TC\_NT\_03**

**\*\*Test Case Title:\*\* Verify that users can set nutritional goals and track progress.**

**\*\*Test Case Description:\*\* This test case verifies that users can set personalized nutritional goals (e.g., daily calorie intake, macronutrient targets) and track their progress towards achieving those goals.**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged into the app.

**\*\*Test Data:\*\***

- Daily calorie goal: 2000 calories
- Protein target: 150 grams

**\*\*Test Steps:\*\***

1. Navigate to the "Nutrition" or "Goals" section of the app.
2. Select "Nutritional Goals."
3. Enter the desired calorie goal and macronutrient targets.
4. Save the goals.
5. View the progress tracker for the set goals.

**\*\*Postconditions:\*\***

- Nutritional goals are saved and reflected in the progress tracker.

**\*\*Expected Result:\*\* The app successfully saves the user's nutritional goals and displays their progress towards those goals in a clear and understandable format.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_NT\_04**

**\*\*Test Case Title:\*\* Verify that the app provides nutritional information and suggestions.**

**\*\*Test Case Description:\*\* This test case confirms that the app offers nutritional information and personalized suggestions based on the user's logged meals, goals, and other data.**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User has logged at least one meal.

**\*\*Test Data:\*\***

- High-sodium meals logged.

**\*\*Test Steps:\*\***

1. Navigate to the "Nutrition" or "Insights" section of the app.
2. View nutritional information and suggestions, such as:
  - Overall macronutrient balance.
  - Potential nutrient deficiencies or excesses.
  - Recommendations for healthier food choices.
  - Information about specific foods logged.

**\*\*Postconditions:\*\***

- No postconditions.

**\*\*Expected Result:\*\* The app provides accurate and relevant nutritional information and personalized suggestions based on the user's data.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC\_NT\_05**

**\*\*Test Case Title:\*\* Verify that users can view their nutritional history and trends.**

**\*\*Test Case Description:\*\* This test case ensures that users can view their past nutritional data and analyze trends over time, such as average daily calorie intake, macronutrient intake patterns, and progress towards their goals.**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User has logged meals over a period of time.

**\*\*Test Data:\*\***

- No specific data required.

**\*\*Test Steps:\*\***

1. Navigate to the "Nutrition" or "History" section of the app.
2. Select a timeframe (e.g., daily, weekly, monthly).
3. View graphs, charts, or tables displaying nutritional data for the selected period.
4. Analyze the trends in calorie intake, macronutrients, and progress towards goals.

**\*\*Postconditions:\*\***

- No postconditions.

**\*\*Expected Result:\*\* The app displays clear and accurate visualizations of the user's nutritional history and identifies trends in their data.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

-----

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

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 Context:\*\***

**\* \*\*Project Name:\*\* Fitness Tracker App**

**\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\*\*Feature Name:\*\*** Progress Reports

**\*\*Feature Description:\*\*** Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment.

**\*\*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:\*\*** Progress Reports

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

\* User is logged in.

\* User has completed some activities within the app.

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***



**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\*** TC\_PR\_01

**\*\*Test Case Title:\*\*** Verify that users receive weekly and monthly progress reports.

**\*\*Test Case Description:\*\*** This test case verifies that the app generates and delivers both weekly and monthly progress reports to users.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Access the Progress Reports section of the app.
2. Verify the availability of both weekly and monthly reports.
3. Check if the app displays the latest weekly and monthly reports.

**\*\*Postconditions:\*\*** User is still logged in and can access the app.

**\*\*Expected Result:\*\*** The app successfully generates and displays both weekly and monthly progress reports.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** TC\_PR\_02

**\*\*Test Case Title:\*\*** Verify that reports include activity levels, workout performance, and goal attainment.

**\*\*Test Case Description:\*\*** This test case verifies that the generated progress reports include essential metrics

like activity levels, workout performance, and goal attainment.

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Open a weekly or monthly progress report.
2. Check if the report displays the user's total activity duration (e.g., steps, calories burned, distance).
3. Verify the report includes details about completed workouts (e.g., types, duration, intensity).
4. Verify the report shows the progress towards any set goals (e.g., weight loss, steps per day).

**\*\*Postconditions:\*\* User is still logged in and can access the app.**

**\*\*Expected Result:\*\* The progress reports display the user's activity levels, workout performance, and goal attainment.**

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* TC\_PR\_03**

**\*\*Test Case Title:\*\* Verify that users can view detailed breakdowns of achievements and trends.**

**\*\*Test Case Description:\*\* This test case checks if users can access detailed breakdowns of their achievements and identify trends in their fitness data.**

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Open a weekly or monthly progress report.
2. Look for a "Detailed Breakdown" section or a feature allowing drill-down into specific metrics.
3. Verify if the report allows viewing data trends over time (e.g., weekly step counts, monthly calorie intake).

4. Check if the report identifies any significant achievements or milestones.

**\*\*Postconditions:\*\*** User is still logged in and can access the app.

**\*\*Expected Result:\*\*** The app presents detailed breakdowns of achievements, trends, and allows users to analyze their data.

**\*\*Severity:\*\*** Minor

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TC\_PR\_04

**\*\*Test Case Title:\*\*** Verify that the app provides recommendations based on report findings.

**\*\*Test Case Description:\*\*** This test case assesses if the app suggests personalized recommendations based on the data displayed in the progress reports.

**\*\*Test Priority:\*\*** Medium

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Open a weekly or monthly progress report.
2. Look for a section labeled "Recommendations" or similar.
3. Verify if the report provides suggestions based on the user's activity data, workout performance, and goal progress (e.g., increase workout frequency, adjust calorie intake).

**\*\*Postconditions:\*\*** User is still logged in and can access the app.

**\*\*Expected Result:\*\*** The app provides relevant recommendations based on the user's progress report data.

**\*\*Severity:\*\*** Minor

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC\_PR\_05**

**\*\*Test Case Title:\*\* Verify that users can customize the frequency and content of reports.**

**\*\*Test Case Description:\*\* This test case checks if users can personalize the frequency and content of their progress reports to match their preferences.**

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Access the Settings section within the app.
2. Navigate to Progress Reports settings.
3. Verify the option to customize report frequency (e.g., weekly, bi-weekly, monthly).
4. Check if users can choose the specific metrics or information displayed in the reports.

**\*\*Postconditions:\*\* User is still logged in and can access the app.**

**\*\*Expected Result:\*\* The app allows users to adjust the frequency and content of their progress reports.**

**\*\*Severity:\*\* Minor**

**\*\*Test Case Approach:\*\* Positive**

-----

**Data - 89**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

## In-App Challenges

### Feature description -

The app offers in-app challenges and competitions, encouraging users to set personal records, compete with friends, and earn rewards for achieving fitness milestones. In-app challenges add an element of fun, motivation, and friendly competition to the fitness tracking experience.

### Userstory title -

As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals.

### Acceptance criteria -

['Users can join various in-app challenges.', 'Challenges have clear objectives and rules.', 'Users can track their progress within the challenge.', 'Rewards are provided for achieving challenge milestones.', 'Users can compete with friends and view leaderboards.']

### Generated Test Cases:

#### ## Test Cases for Fitness Tracker App - In-App Challenges

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\*\*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.

**\*\*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:\*\* In-App Challenges**

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

- User is logged in to the app.
- User has opted-in to participate in challenges.

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\* Functional Testing**

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC-CHAL-01**

**\*\*Test Case Title:\*\* Verify that users can join various in-app challenges.**

**\*\*Test Case Description:\*\* Ensure that the app displays a variety of challenges for users to join, with different durations, difficulty levels, and objectives.**

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the "Challenges" section of the app.
2. Verify that multiple challenges are listed with clear titles, descriptions, and durations.
3. Select a challenge and click the "Join" button.
4. Verify that the challenge is added to the user's "My Challenges" list.

**\*\*Postconditions:\*\* The user is now a participant in the chosen challenge.**

**\*\*Expected Result:\*\* The app should display a list of available challenges, allowing users to select and join them easily.**

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* TC-CHAL-02**

**\*\*Test Case Title:\*\* Verify that challenges have clear objectives and rules.**

**\*\*Test Case Description:\*\* Ensure that each challenge has a clear objective and specific rules, outlining how users can participate and win.**

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Choose a challenge from the list.
2. Read the "Challenge Description" section.
3. Verify that the description clearly outlines the challenge objective (e.g., "Walk 10,000 steps per day for a week").
4. Verify that the "Rules" section specifies the challenge duration, required actions, and any specific

limitations.

**\*\*Postconditions:\*\*** The user understands the challenge objective and rules.

**\*\*Expected Result:\*\*** The challenge description and rules should be clear and easy to understand, providing users with a comprehensive overview of the challenge.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TC-CHAL-03

**\*\*Test Case Title:\*\*** Verify that users can track their progress within the challenge.

**\*\*Test Case Description:\*\*** Ensure that the app provides a progress tracker for each challenge, allowing users to monitor their performance against the challenge objective.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. After joining a challenge, navigate to "My Challenges".
2. Select the active challenge.
3. Verify that a progress tracker is displayed.
4. Track your progress through the challenge (e.g., step count, distance covered).
5. Verify that the progress tracker updates in real-time.

**\*\*Postconditions:\*\*** The user can monitor their progress in the challenge.

**\*\*Expected Result:\*\*** The app should provide a clear and dynamic progress tracker, allowing users to see their current progress and how it relates to the challenge objective.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive



## **\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* TC-CHAL-04**

**\*\*Test Case Title:\*\* Verify that rewards are provided for achieving challenge milestones.**

**\*\*Test Case Description:\*\* Ensure that the app offers rewards for users who successfully complete challenge milestones or achieve certain targets.**

**\*\*Test Priority:\*\* High**

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Participate in a challenge with predefined milestones.
2. Achieve a specific milestone (e.g., complete 3 workouts).
3. Verify that the app acknowledges the milestone achievement.
4. Verify that the corresponding reward is unlocked and displayed (e.g., achievement badge, virtual currency).

**\*\*Postconditions:\*\* The user receives a reward for completing the milestone.**

**\*\*Expected Result:\*\* The app should clearly display milestones and the associated rewards, motivating users to achieve their goals.**

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

## **\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC-CHAL-05**

**\*\*Test Case Title:\*\* Verify that users can compete with friends and view leaderboards.**

**\*\*Test Case Description:\*\* Ensure that users can connect with friends to participate in challenges together, track their standings on leaderboards, and compete for higher rankings.**

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\*** Multiple user accounts with friend connections.

**\*\*Test Steps:\*\***

1. Invite a friend to participate in a challenge.
2. Verify that the friend receives the invitation.
3. Both users join the challenge.
4. Track the progress of both users on the leaderboard.
5. Verify that the leaderboard updates in real-time, reflecting the latest progress.

**\*\*Postconditions:\*\*** Both users are now competing with each other within the challenge.

**\*\*Expected Result:\*\*** The app should allow users to connect with friends, participate in challenges together, and view a live leaderboard showcasing their progress and rankings.

**\*\*Severity:\*\*** Minor

**\*\*Test Case Approach:\*\*** Positive

-----

## **Data - 90**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Customizable Dashboards

### **Feature description -**

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based

on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

**Userstory title -**

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

**Acceptance criteria -**

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

**Generated Test Cases:**

## Test Cases for Customizable Dashboards Feature:

**\*\*Test Case 1\*\***

**\*\*Test Case ID:\*\* TC\_DASH\_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 and unchoose specific fitness metrics to display on their dashboard, ensuring that the dashboard reflects their personalized preferences.

**\*\*Test Suite:\*\*** Customizable Dashboards

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the Fitness Tracker App
- User has access to the Dashboard settings

**\*\*Test Data:\*\***

- List of available metrics: steps, calories burned, distance, heart rate, sleep duration, water intake, weight

**\*\*Test Steps:\*\***

1. Access the Dashboard settings.
2. Navigate to the "Metrics" section.
3. Select "Steps", "Calories Burned", and "Distance" metrics to display.
4. Deselect "Sleep Duration" and "Water Intake" metrics.
5. Save the dashboard settings.
6. Verify that the selected metrics (Steps, Calories Burned, Distance) are displayed on the dashboard, and the deselected metrics (Sleep Duration, Water Intake) are not visible.

**\*\*Postconditions:\*\***

- Dashboard settings are updated to reflect the chosen metrics.

**\*\*Expected Result:\*\*** The system displays the selected metrics on the dashboard and excludes the deselected ones.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** TC\_DASH\_02

**\*\*Test Case Title:\*\*** Verify that the app allows customization of dashboard layout and theme.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of customizing the layout and theme of the dashboard, allowing users to personalize the visual appearance of their dashboard.

**\*\*Test Suite:\*\*** Customizable Dashboards

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in to the Fitness Tracker App
- User has access to the Dashboard settings

**\*\*Test Data:\*\***

- List of available dashboard layouts: Grid, List, Compact

- List of available themes: Light, Dark, Classic

**\*\*Test Steps:\*\***

1. Access the Dashboard settings.
2. Navigate to the "Layout" section.
3. Choose "List" layout.
4. Navigate to the "Theme" section.
5. Select "Dark" theme.
6. Save the dashboard settings.
7. Verify that the dashboard layout is "List" and the theme is "Dark".

**\*\*Postconditions:\*\***

- Dashboard settings are updated to reflect the chosen layout and theme.

**\*\*Expected Result:\*\*** The dashboard displays the selected layout and theme.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TC\_DASH\_03

**\*\*Test Case Title:\*\*** Verify that users can save and update their dashboard settings.

**\*\*Test Case Description:\*\*** This test case verifies the ability of users to save their customized dashboard settings and update them later, allowing for easy customization and personalization.

**\*\*Test Suite:\*\*** Customizable Dashboards

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the Fitness Tracker App

- User has access to the Dashboard settings

**\*\*Test Data:\*\***

- Existing dashboard settings with specific metrics, layout, and theme.

**\*\*Test Steps:\*\***

1. Access the Dashboard settings.
2. Modify existing dashboard settings: change the layout to "Grid", add "Heart Rate" metric, and change the theme to "Classic".
3. Save the dashboard settings.
4. Verify that the saved settings are applied to the dashboard.
5. Access the Dashboard settings again.
6. Update the settings: change the theme to "Dark" and remove "Heart Rate" metric.
7. Save the updated settings.
8. Verify that the updated settings are applied to the dashboard.

**\*\*Postconditions:\*\***

- Dashboard settings are updated to reflect the saved and updated changes.

**\*\*Expected Result:\*\*** The system saves and updates the dashboard settings, applying the changes to the dashboard interface.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TC\_DASH\_04

**\*\*Test Case Title:\*\*** Verify that dashboard displays are dynamic and update in real-time.

**\*\*Test Case Description:\*\*** This test case verifies that the dashboard displays updated information in real-time, ensuring that the user has access to the most recent data and progress.

**\*\*Test Suite:\*\* Customizable Dashboards**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the Fitness Tracker App
- User has access to the Dashboard settings
- User has activity data recorded in the app (steps, calories, etc.)

**\*\*Test Data:\*\***

- Existing dashboard settings with selected metrics.

**\*\*Test Steps:\*\***

1. Access the Dashboard.
2. Record a short workout activity (e.g., walk for 10 minutes).
3. Verify that the selected metrics on the dashboard update in real-time to reflect the new activity data.
4. Observe if the dashboard refreshes automatically after the activity is completed or if a manual refresh is required.

**\*\*Postconditions:\*\***

- Dashboard displays updated data based on the latest recorded activity.

**\*\*Expected Result:\*\*** The dashboard displays dynamic and updated metrics in real-time, reflecting the user's latest activity data.

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Performance Testing**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* TC\_DASH\_05**

**\*\*Test Case Title:\*\* Verify that users can reset the dashboard to default settings.**

**\*\*Test Case Description:\*\* This test case verifies the ability of users to reset their dashboard to its default**

settings, providing a quick and easy way to restore the original configuration if needed.

**\*\*Test Suite:\*\*** Customizable Dashboards

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in to the Fitness Tracker App
- User has access to the Dashboard settings
- User has customized their dashboard settings (metrics, layout, theme)

**\*\*Test Data:\*\***

- Existing dashboard settings.

**\*\*Test Steps:\*\***

1. Access the Dashboard settings.
2. Locate the "Reset to Defaults" option.
3. Click on "Reset to Defaults".
4. Confirm the reset action.
5. Verify that the dashboard is reset to its default configuration: the default metrics, layout, and theme are applied.

**\*\*Postconditions:\*\***

- Dashboard settings are reset to the default configuration.

**\*\*Expected Result:\*\*** The system resets the dashboard to its default settings, including the default metrics, layout, and theme.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

-----



**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

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:**

## Test Cases for Fitness Tracker App - Social Sharing Feature

**\*\*Project Context:\*\***

**\* \*\*Project Name:\*\* Fitness Tracker App**

\* \*\*Project Description:\*\* A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

\* \*\*Feature Name:\*\* Social Sharing

\* \*\*Feature Description:\*\* Users can share workout summaries, achievements, and progress updates on social media platforms.

\* \*\*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	Type of Testing	Test Case Approach
TC-SS-01	Verify that users can share workout summaries	Ensure that users can share summaries of their completed workouts on social media platforms.	Social Sharing	High	- User is logged in. - A workout is completed and saved in the app.	No test data needed	1. Navigate to the workout summary page. 2. Tap/click on the share button. 3. Select a social media platform. 4. Share the workout summary.	- The workout summary is shared on the selected social media platform.	The workout summary is successfully shared with the chosen social media platform.	Major	Functional Testing	Positive

| TC-SS-02 | Verify that the app supports sharing on multiple social media platforms | Ensure that users can share their fitness data on different social media platforms. | Social Sharing | High | - User is logged in. | - Facebook, Twitter, Instagram accounts linked to the app. | 1. Navigate to the workout summary page. 2. Tap/click on the share button. 3. Observe available social media platforms. 4. Select different social media platforms and attempt to share. | - The workout summary is shared on the selected social media platform. | The app supports sharing on multiple social media platforms like Facebook, Twitter, Instagram, etc. | Major | Functional Testing | Positive |

| TC-SS-03 | Verify that users can customize the content and format of shared posts | Ensure that users can edit the content and format of their shared posts. | Social Sharing | Medium | - User is logged in. - A workout is completed and saved. | No test data needed | 1. Navigate to the workout summary page. 2. Tap/click on the share button. 3. Add custom text to the share post. 4. Modify the format (e.g., adding emojis, hashtags). 5. Share the post. | - The shared post includes the customized text and format. | The user can customize the content and format of the shared posts, including adding text, emojis, and hashtags. | Minor | Functional Testing | Positive |

| TC-SS-04 | Verify that sharing features are seamlessly integrated with social media | Ensure that the app's sharing features smoothly integrate with the user's social media accounts. | Social Sharing | High | - User is logged in. - Social media accounts are linked to the app. | No test data needed | 1. Navigate to the workout summary page. 2. Tap/click on the share button. 3. Select a social media platform. 4. Observe the sharing process. 5. Verify if the app authenticates with the selected social media platform. | - The sharing process is smooth and user-friendly. - The app seamlessly integrates with the social media account. | Major | Integration Testing | Positive |

| TC-SS-05 | Verify that users can track the engagement of their shared posts | Ensure that users can view and track the likes, comments, and shares of their shared posts. | Social Sharing | Medium | - User is logged in. - A workout summary is shared on a social media platform. | No test data needed | 1. Navigate to the social media platform where the post was shared. 2. Observe the post engagement metrics (likes, comments, shares). 3. Check if the app displays the engagement data in the post history section. | - The app displays the engagement metrics for the shared post. | Users can view and track the engagement of their shared posts, including likes,

**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

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** FT-RRT-01

**\*\*Test Case Title:\*\*** Verify that users can log rest days and recovery activities.

**\*\*Test Case Description:\*\*** This test case verifies the functionality for users to log rest days and recovery activities within the Fitness Tracker App. It ensures that the app provides options for users to indicate rest days and to log various types of recovery activities, including sleep, stretching, massage, or other activities.

**\*\*Test Suite:\*\*** Rest and Recovery Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the app.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Rest & Recovery" section of the app.
2. Select "Log Rest Day" or "Log Recovery Activity".
3. Choose the type of recovery activity from a predefined list.
4. Enter details about the activity, such as duration, time, and additional notes.
5. Save the logged activity.

**\*\*Postconditions:\*\***

- The logged rest day or recovery activity is saved in the user's activity history.

**\*\*Expected Result:\*\*** The app successfully records the rest day or recovery activity, including the specified details, and displays it in the user's activity history.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* FT-RRT-02**

**\*\*Test Case Title:\*\*** Verify that the app tracks resting heart rate and sleep quality.

**\*\*Test Case Description:\*\*** This test case checks the app's ability to track resting heart rate and sleep quality.

It ensures that the app has functionality to record and store these metrics for the user.

**\*\*Test Suite:\*\*** Rest and Recovery Tracking

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the app.
- User has enabled heart rate tracking and sleep tracking.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Rest & Recovery" section of the app.
2. Check if the app displays the user's resting heart rate data.
3. Check if the app displays the user's sleep quality data.

**\*\*Postconditions:\*\***

- The app displays the user's resting heart rate and sleep quality data.

**\*\*Expected Result:\*\*** The app successfully displays the user's recent resting heart rate and sleep quality data, either automatically tracked or manually entered by the user.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* FT-RRT-03**

**\*\*Test Case Title:\*\* Verify that users receive recommendations for recovery based on their activities.**

**\*\*Test Case Description:\*\* This test case assesses whether the app provides personalized recovery recommendations based on the user's recent activity data. It ensures that the app analyzes the user's activity history and provides appropriate recovery suggestions.**

**\*\*Test Suite:\*\* Rest and Recovery Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the app.
- User has logged at least 3 days of activity, including both exercise and rest days.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the "Rest & Recovery" section of the app.
2. Check for any recovery recommendations provided by the app.
3. Observe the type of recommendations, such as recommended rest time, recovery activities, or adjustments to workout schedule.

**\*\*Postconditions:\*\***

- The app provides recovery recommendations based on the user's activity data.

**\*\*Expected Result:\*\* The app provides personalized recovery recommendations, considering the intensity, duration, and type of the user's recent activities, recommending suitable rest periods, recovery activities, or other adjustments for optimal recovery.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* FT-RRT-04**

**\*\*Test Case Title:\*\* Verify that the app provides alerts for insufficient recovery time.**

**\*\*Test Case Description:\*\* This test case verifies the app's functionality to alert users when they have not had sufficient recovery time between workouts or after a period of intense activity.**

**\*\*Test Suite:\*\* Rest and Recovery Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the app.
- User has enabled notifications.
- User has logged a recent intense workout with minimal rest time afterward.

**\*\*Test Data:\*\* No test data needed.**

**\*\*Test Steps:\*\***

1. Navigate to the "Rest & Recovery" section of the app.
2. Check if the app provides an alert or notification regarding insufficient recovery time.
3. Observe the alert content, including the type of recovery recommended.

**\*\*Postconditions:\*\***

- The app displays an alert for insufficient recovery time.

**\*\*Expected Result:\*\* The app accurately assesses the user's recovery time based on their recent activities and provides a timely alert if they have not had sufficient recovery, suggesting appropriate recovery actions.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* FT-RRT-05**

**\*\*Test Case Title:\*\* Verify that users can view trends and history of their recovery data.**



**\*\*Test Case Description:\*\*** This test case checks the app's ability to display trends and historical data related to the user's rest and recovery. It ensures that the user can visualize their recovery progress over time.

**\*\*Test Suite:\*\*** Rest and Recovery Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in to the app.
- User has logged at least 1 week of rest and recovery data.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Rest & Recovery" section of the app.
2. Look for any available options to view historical data or trends.
3. Select options to view charts or graphs displaying resting heart rate, sleep quality, recovery activities, or other recovery metrics over time.

**\*\*Postconditions:\*\***

- The app displays historical data and trends related to recovery metrics.

**\*\*Expected Result:\*\*** The app allows users to access their historical data and visualize trends over time for resting heart rate, sleep quality, recovery activities, and other metrics, enabling them to track their overall recovery progress.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 93**

**Project name -**

Fitness Tracker App

## **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

## **Feature name -**

Audio Coaching

## **Feature description -**

Users can access audio-guided workouts, coaching cues, and motivational messages during their workouts, enhancing engagement and adherence. Audio coaching provides real-time feedback, encouragement, and guidance, simulating the experience of having a personal trainer or workout buddy.

## **Userstory title -**

As a user, I want to access audio coaching during workouts so that I can stay motivated and receive real-time guidance.

## **Acceptance criteria -**

['Users can access a variety of audio-guided workouts.', 'Audio coaching includes motivational messages and coaching cues.', 'Users can choose coaching based on workout type and intensity.', 'Audio coaching integrates seamlessly with other app features.', 'Users can control audio settings and preferences.']

## **Generated Test Cases:**

## Test Cases for Fitness Tracker App - Audio Coaching Feature

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

**\*\*Feature Name:\*\* Audio Coaching**

**\*\*Feature Description:\*\*** Users can access audio-guided workouts, coaching cues, and motivational messages

during their workouts.

**\*\*User Story 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:\*\***

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 1\*\***

**\*\*Test Case ID:\*\*** TC\_AC1

**\*\*Test Case Title:\*\*** Verify that users can access a variety of audio-guided workouts.

**\*\*Test Case Description:\*\*** Ensure that the app offers a diverse selection of audio-guided workouts for different fitness levels and workout types.

**\*\*Test Suite:\*\*** Audio Coaching

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- App has internet connection.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Workouts" section of the app.
2. Select the "Audio Coaching" tab.
3. Explore the available workout categories (e.g., cardio, strength training, yoga).
4. Choose a specific workout category.
5. Verify that a list of audio-guided workouts within that category appears.
6. Check the duration, intensity level, and workout type for each available workout.

**\*\*Postconditions:\*\*** The app displays a list of audio-guided workouts with varied durations, intensity levels, and workout types.

**\*\*Expected Result:\*\*** The system presents a diverse range of audio-guided workouts across different categories and intensity levels.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\*** TC\_AC2

**\*\*Test Case Title:\*\*** Verify that audio coaching includes motivational messages and coaching cues.

**\*\*Test Case Description:\*\*** Ensure that the audio coaching feature provides motivational messages and specific coaching cues throughout the workout.

**\*\*Test Suite:\*\*** Audio Coaching

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- Audio coaching workout is selected.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Select an audio-guided workout.
2. Start the workout and listen to the audio coaching.
3. Note the presence of motivational messages during the workout.
4. Verify that coaching cues are given at appropriate times during the workout (e.g., reminders about form, breathing, timing).

**\*\*Postconditions:\*\*** The audio coaching provides motivational messages and coaching cues during the workout.

**\*\*Expected Result:\*\*** The audio coaching delivers motivational messages and coaching cues throughout the workout, enhancing the user's experience.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\*** TC\_AC3

**\*\*Test Case Title:\*\*** Verify that users can choose coaching based on workout type and intensity.

**\*\*Test Case Description:\*\*** Ensure that users can select audio coaching based on their desired workout type and intensity level.

**\*\*Test Suite:\*\*** Audio Coaching

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Navigate to the "Workouts" section.
2. Choose a specific workout category (e.g., cardio).
3. Filter the workout list by workout type (e.g., running, cycling).
4. Filter the workout list by intensity level (e.g., beginner, intermediate, advanced).
5. Verify that the available audio coaching workouts are filtered according to the selected criteria.

**\*\*Postconditions:\*\*** The app filters the audio coaching workouts based on the user's preferences for workout type and intensity level.

**\*\*Expected Result:\*\*** The system filters audio coaching workouts based on the selected workout type and intensity level, providing relevant choices for users.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\*** TC\_AC4

**\*\*Test Case Title:\*\*** Verify that audio coaching integrates seamlessly with other app features.

**\*\*Test Case Description:\*\*** Ensure that the audio coaching feature seamlessly integrates with other app functionalities, such as workout tracking and progress tracking.

**\*\*Test Suite:\*\*** Audio Coaching

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.
- Audio coaching workout is selected.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Start an audio-guided workout.
2. Verify that the app automatically tracks the workout duration, calories burned, and other relevant metrics.
3. Pause the workout and check the progress bar.
4. Resume the workout and continue listening to audio coaching.
5. After completing the workout, ensure that the app records the workout data in the user's history.

**\*\*Postconditions:\*\*** The audio coaching feature seamlessly integrates with workout tracking and progress tracking functionalities.

**\*\*Expected Result:\*\*** The app smoothly integrates audio coaching with workout tracking and progress tracking features, providing a consistent user experience.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Integration Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\*** TC\_AC5

**\*\*Test Case Title:\*\*** Verify that users can control audio settings and preferences.

**\*\*Test Case Description:\*\*** Ensure that users can adjust audio settings such as volume, playback speed, and coaching cues based on their preferences.

**\*\*Test Suite:\*\*** Audio Coaching

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.
- Audio coaching workout is selected.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Start an audio-guided workout.
2. Access the audio settings menu within the app.
3. Adjust the volume of the audio coaching.
4. Change the playback speed of the audio.
5. Enable/disable specific types of coaching cues (e.g., form correction, motivational messages).
6. Verify that the audio settings changes are applied and saved for future workouts.

**\*\*Postconditions:\*\*** The app allows users to adjust audio settings and save their preferences for future workouts.

**\*\*Expected Result:\*\*** The system provides users with control over audio settings, allowing them to customize their audio coaching experience.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Usability Testing

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 94**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Integration with Health Apps



## **Feature description -**

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

## **Userstory title -**

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

## **Acceptance criteria -**

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

## **Generated Test Cases:**

## Test Cases for Fitness Tracker App Integration with Health Apps

**\*\*Project Name:\*\* Fitness Tracker App**

**\*\*Project Description:\*\*** 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.

**\*\*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.

**\*\*User Story 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.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\***

**\*\*Test Case Title:\*\***

**\*\*Test Case Description:\*\***

**\*\*Test Suite:\*\*** Integration with Health Apps

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged into the Fitness Tracker App
- User has an account with a compatible health app (e.g., MyFitnessPal, Headspace)
- Both apps are installed on the same device

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Integration Testing

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TCI-01

**\*\*Test Case Title:\*\*** Verify that the app integrates with popular health and wellness apps.

**\*\*Test Case Description:\*\*** This test case verifies that the Fitness Tracker app supports integration with

popular health and wellness apps such as MyFitnessPal, Headspace, and others.

**\*\*Test Steps:\*\***

1. Navigate to the app settings section.
2. Locate the "Health App Integration" section.
3. Check for a list of supported health apps.
4. Verify that the list includes popular and commonly used health and wellness apps.

**\*\*Postconditions:\*\*** The list of supported health apps is displayed in the app settings.

**\*\*Expected Result:\*\*** The app supports integration with popular health and wellness apps, displaying a list of compatible apps within the settings.

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TCI-02

**\*\*Test Case Title:\*\*** Verify that users can sync data from connected health apps.

**\*\*Test Case Description:\*\*** This test case verifies that the Fitness Tracker app can successfully sync data from connected health apps, allowing users to access a comprehensive view of their health data.

**\*\*Test Steps:\*\***

1. Select a supported health app from the list.
2. Grant the Fitness Tracker app permission to access data from the selected health app.
3. Initiate data sync from the connected health app to the Fitness Tracker app.
4. Monitor the sync progress.
5. Verify that data from the connected health app is successfully imported into the Fitness Tracker app.

**\*\*Postconditions:\*\*** Data is synced from the connected health app to the Fitness Tracker app.

**\*\*Expected Result:\*\*** The app successfully syncs data from connected health apps, displaying the imported data in the Fitness Tracker app.

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\* TCI-03**

**\*\*Test Case Title:\*\*** Verify that the app provides a consolidated view of health metrics.

**\*\*Test Case Description:\*\*** This test case verifies that the Fitness Tracker app consolidates and displays health metrics from both the Fitness Tracker app and connected health apps in a unified view.

**\*\*Test Steps:\*\***

1. Navigate to the "Health Metrics" section of the Fitness Tracker app.
2. Verify that the section displays metrics from both the Fitness Tracker app and connected health apps.
3. Observe if the consolidated data is presented in an organized and user-friendly manner.
4. Verify that the user can easily switch between different data sources within the "Health Metrics" section.

**\*\*Postconditions:\*\*** Health metrics are displayed in a unified view, combining data from different apps.

**\*\*Expected Result:\*\*** The app provides a consolidated view of health metrics, presenting a unified view of data from the Fitness Tracker app and connected health apps.

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TCI-04**

**\*\*Test Case Title:\*\*** Verify that users receive insights based on combined data from multiple apps.

**\*\*Test Case Description:\*\*** This test case verifies that the Fitness Tracker app can generate insights based on the combined data from both the Fitness Tracker app and connected health apps.

**\*\*Test Steps:\*\***

1. Ensure data has been synced from connected health apps to the Fitness Tracker app.
2. Navigate to the "Insights" or "Recommendations" section of the Fitness Tracker app.
3. Verify that the app provides insights based on the combined data from all connected health apps and the Fitness Tracker app itself.
4. Observe if the insights are relevant and personalized based on the user's overall health data.

**\*\*Postconditions:\*\*** The app generates relevant insights based on combined data from multiple apps.

**\*\*Expected Result:\*\*** The app provides personalized insights and recommendations based on the combined health data from multiple apps.

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TCI-05

**\*\*Test Case Title:\*\*** Verify that integration settings are easy to manage and configure.

**\*\*Test Case Description:\*\*** This test case verifies that the user can easily manage and configure integration settings for connected health apps.

**\*\*Test Steps:\*\***

1. Navigate to the "Health App Integration" section of the app settings.
2. Verify that the user can add, remove, or modify connected health apps.
3. Observe if the user can adjust data sync frequency and specific data types to be synced from each connected health app.
4. Verify that the user can easily revoke permissions granted to the Fitness Tracker app for accessing data from connected health apps.

**\*\*Postconditions:\*\*** Users can easily manage and configure integration settings.

**\*\*Expected Result:\*\*** The app provides a user-friendly interface for managing integration settings, allowing users to add, remove, modify, and revoke permissions for connected health apps.

**Data - 95**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Body Composition Tracking

**Feature description -**

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and the effectiveness of exercise and nutrition interventions.

**Userstory title -**

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed decisions about my training and nutrition.

**Acceptance criteria -**

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set goals for body composition metrics.']

**Generated Test Cases:**

## Test Cases for Fitness Tracker App - Body Composition Tracking

**\*\*Project Context:\*\***

\* **\*\*Project Name:\*\*** Fitness Tracker App

\* **\*\*Project Description:\*\*** A fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics.

\* **\*\*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.

\* \*\*User Story 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:\*\*

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 Creation Template:\*\*

\*\*Test Case ID:\*\*

\*\*Test Case Title:\*\*

\*\*Test Case Description:\*\*

\*\*Test Suite:\*\* Body Composition Tracking

\*\*Test Priority:\*\*

\*\*Preconditions:\*\*

\* User is logged in

\* User has a profile set up

\*\*Test Data:\*\*

\*\*Test Steps:\*\*

\*\*Postconditions:\*\*

\*\*Expected Result:\*\*

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_001

**\*\*Test Case Title:\*\*** Verify that users can log body composition metrics such as weight, body fat percentage, and muscle mass.

**\*\*Test Case Description:\*\*** This test case ensures the system allows users to enter and save data for weight, body fat percentage, and muscle mass.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\***

\* Weight: 75 kg

\* Body Fat Percentage: 18%

\* Muscle Mass: 55 kg

**\*\*Test Steps:\*\***

1. Navigate to the Body Composition section of the app.
2. Tap on the "Add Entry" button.
3. Enter the test data values for weight, body fat percentage, and muscle mass.
4. Tap on the "Save" button.

**\*\*Postconditions:\*\***

\* The entered data is saved in the user's body composition history.

**\*\*Expected Result:\*\*** The system should successfully save the entered body composition metrics in the user's history.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive



**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_002

**\*\*Test Case Title:\*\*** Verify that the app calculates and displays BMI based on logged data.

**\*\*Test Case Description:\*\*** This test case checks if the app correctly calculates BMI using the entered weight and height data.

**\*\*Test Priority:\*\*** High

**\*\*Test Data:\*\***

\* Weight: 75 kg

\* Height: 1.80 meters

**\*\*Test Steps:\*\***

1. Navigate to the Body Composition section of the app.
2. View the "BMI" section.
3. Ensure the BMI value displayed matches the calculated BMI ( $75 / (1.80 * 1.80) = 23.15$ ).

**\*\*Postconditions:\*\***

\* The displayed BMI is accurate.

**\*\*Expected Result:\*\*** The app accurately calculates and displays the user's BMI based on the logged weight and height data.

**\*\*Severity:\*\*** Major

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_003

**\*\*Test Case Title:\*\*** Verify that users can view trends and history of body composition changes.

**\*\*Test Case Description:\*\*** This test case ensures that users can view their body composition data over time,

including graphical representations of changes.

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\***

\* Previous body composition entries for different dates.

**\*\*Test Steps:\*\***

1. Navigate to the Body Composition section of the app.
2. Access the "History" tab or view.
3. Verify that the app displays a timeline or history of all recorded body composition data.
4. Verify that the app provides options for viewing trends, including graphs or charts, showing changes in weight, body fat percentage, and muscle mass over time.

**\*\*Postconditions:\*\***

\* The user can view and analyze their body composition history.

**\*\*Expected Result:\*\*** The app successfully displays a history of body composition changes and allows users to visualize trends in the data.

**\*\*Severity:\*\* Major**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\* TC\_004**

**\*\*Test Case Title:\*\*** Verify that the app provides insights and recommendations based on body composition data.

**\*\*Test Case Description:\*\*** This test case verifies that the app analyzes the user's body composition data and provides personalized recommendations related to training, nutrition, or health.

**\*\*Test Priority:\*\* Medium**

**\*\*Test Data:\*\***

\* Body composition data indicative of specific conditions, such as high body fat percentage, low muscle mass,

or a sudden weight change.

**\*\*Test Steps:\*\***

1. Navigate to the Body Composition section of the app.
2. View the "Insights" or "Recommendations" section.
3. Verify that the app provides relevant recommendations based on the user's body composition data.
4. Check if the recommendations include suggestions for exercise, nutrition, or other health considerations.

**\*\*Postconditions:\*\***

\* The app provides actionable insights and recommendations based on body composition trends.

**\*\*Expected Result:\*\*** The app analyzes the user's body composition data and provides relevant and personalized insights and recommendations.

**\*\*Severity:\*\*** Minor

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_005

**\*\*Test Case Title:\*\*** Verify that users can set goals for body composition metrics.

**\*\*Test Case Description:\*\*** This test case ensures that users can set specific targets for weight, body fat percentage, and muscle mass.

**\*\*Test Priority:\*\*** Medium

**\*\*Test Data:\*\***

\* Target weight: 70 kg

\* Target body fat percentage: 15%

\* Target muscle mass: 60 kg

**\*\*Test Steps:\*\***

1. Navigate to the Body Composition section of the app.
2. Access the "Goals" or "Targets" section.

3. Set specific goals for weight, body fat percentage, and muscle mass.

4. Verify that the set goals are saved.

**\*\*Postconditions:\*\***

\* The app stores the set goals for the user.

**\*\*Expected Result:\*\*** The app allows users to set and save specific goals for their body composition metrics.

**\*\*Severity:\*\*** Major

**\*\*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 Context:\*\*

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.

User Story 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.

## **\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** (Provide a unique identifier for each test case)

**\*\*Test Case Title:\*\*** (Must start with "Verify that..." and clearly state the feature/function being tested)

**\*\*Test Case Description:\*\*** (A brief description of the test case, between 70-100 words)

**\*\*Test Suite:\*\*** (Name of the test suite or module)

**\*\*Test Priority:\*\*** (Priority level: High, Medium, Low)

**\*\*Preconditions:\*\*** (List any prerequisites before executing the test case, max 3 items, in unordered list)

**\*\*Test Data:\*\*** (Specify data required for execution; if none, state "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, in unordered list)

**\*\*Expected Result:\*\*** (Overall expected outcome of the test case)

**\*\*Severity:\*\*** (Blocker, Critical, Major, Minor, Trivial)

**\*\*Type of Testing:\*\*** (Unit Testing, Integration Testing, System Testing, Functional Testing, Non-Functional Testing, Regression Testing, Acceptance Testing, Performance Testing, Load Testing, Stress Testing, End-to-End Testing, Security Testing, Usability Testing, Compatibility Testing, Sanity Testing, Smoke Testing, Exploratory Testing, Ad-Hoc Testing, Data-Driven Testing, Cross-Browser Testing, API Testing, etc.)

**\*\*Test Case Approach:\*\*** (Positive, Negative, Destructive)

### **\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_WA\_01

**\*\*Test Case Title:\*\*** Verify that users can view detailed performance metrics for each workout

**\*\*Test Case Description:\*\*** This test case ensures that the app displays detailed performance metrics for each completed workout, including duration, distance, calories burned, heart rate, and average pace.

**\*\*Test Suite:\*\*** Workout Analysis

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- User has completed at least one workout

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Go to the "Workouts" section of the app.
2. Select a completed workout from the list.
3. Observe the displayed performance metrics for the selected workout.

**\*\*Postconditions:\*\* None**

**\*\*Expected Result:\*\* The app displays detailed performance metrics for the selected workout, including duration, distance, calories burned, heart rate, and average pace.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\* TC\_WA\_02**

**\*\*Test Case Title:\*\* Verify that the app provides trends and insights into workout intensity and duration**

**\*\*Test Case Description:\*\* This test case validates the app's ability to show trends and insights related to workout intensity and duration based on historical data, helping users understand their progress over time.**

**\*\*Test Suite:\*\* Workout Analysis**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- User has completed multiple workouts with varying intensity and duration

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Go to the "Workout Analysis" section of the app.
2. Observe the graphs and charts related to workout intensity and duration.
3. Analyze the trends and insights presented, such as average workout duration, intensity variations, and progress over time.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The app presents graphs and charts that display trends and insights into workout intensity and duration, highlighting changes over time and providing meaningful information about the user's training patterns.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_WA\_03

**\*\*Test Case Title:\*\*** Verify that users can track calorie burn and other health metrics

**\*\*Test Case Description:\*\*** This test case checks that the app accurately tracks calorie burn and other health metrics during workouts, ensuring the user can monitor their overall progress and health impact.

**\*\*Test Suite:\*\*** Workout Analysis

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in
- User has completed at least one workout

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***



1. Go to the "Workouts" section of the app.
2. Select a completed workout.
3. View the calorie burn and other health metrics associated with the workout, such as heart rate and steps taken.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The app displays accurate data for calorie burn and other health metrics related to the selected workout.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_WA\_04

**\*\*Test Case Title:\*\*** Verify that the app offers recommendations based on workout analysis

**\*\*Test Case Description:\*\*** This test case checks whether the app offers personalized recommendations based on the user's workout history, helping them optimize their training strategies and achieve their fitness goals.

**\*\*Test Suite:\*\*** Workout Analysis

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in
- User has completed multiple workouts with varying intensity and duration

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Go to the "Workout Analysis" section of the app.
2. Analyze the presented insights and trends.
3. Observe if the app provides personalized recommendations for workout adjustments, based on the

analyzed data.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The app provides personalized recommendations based on the user's workout history and analysis, suggesting adjustments for intensity, duration, or frequency to improve performance and achieve fitness goals.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\*** TC\_WA\_05

**\*\*Test Case Title:\*\*** Verify that users can compare current workouts with past performance

**\*\*Test Case Description:\*\*** This test case checks if the app allows users to compare their current workout performance with previous workouts, providing valuable insights into their progress and areas for improvement.

**\*\*Test Suite:\*\*** Workout Analysis

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in
- User has completed multiple workouts of the same type

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Go to the "Workouts" section of the app.
2. Select two workouts of the same type (e.g., running) completed on different dates.
3. Utilize any available comparison functionality within the app to analyze the difference in performance metrics between the two workouts.

**\*\*Postconditions:\*\*** None

**\*\*Expected Result:\*\*** The app allows users to compare selected workouts, displaying performance metrics side-by-side, allowing them to analyze progress, identify areas for improvement, and track their overall development.

**\*\*Severity:\*\*** Minor

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 97**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Weather Integration

**Feature description -**

Integrated weather forecasts and conditions allow users to plan outdoor workouts and activities based on current weather conditions and forecasts. Weather integration helps users make informed decisions, stay safe, and optimize their outdoor exercise routines according to weather conditions.

**Userstory title -**

As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

**Acceptance criteria -**

['The app provides current weather conditions and forecasts.', 'Users can view weather details for different locations.', 'Weather data is updated in real-time.', 'The app offers recommendations based on weather conditions.', 'Users can receive weather alerts for extreme conditions.']

## **Generated Test Cases:**

### **## Test Cases for Fitness Tracker App - Weather Integration**

**\*\*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.

**\*\*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:\*\*** Weather Integration

**\*\*Test Priority:\*\***

**\*\*Preconditions:\*\***

**\*\*Test Data:\*\***

**\*\*Test Steps:\*\***

**\*\*Postconditions:\*\***

**\*\*Expected Result:\*\***

**\*\*Severity:\*\***

**\*\*Type of Testing:\*\***

**\*\*Test Case Approach:\*\***

**\*\*Test Case 1:\*\***

**\*\*Test Case ID:\*\*** TC\_Weather\_01

**\*\*Test Case Title:\*\*** Verify that the app provides current weather conditions and forecasts.

**\*\*Test Case Description:\*\*** This test case verifies that the app displays the current weather conditions and forecasts for the user's current location.

**\*\*Test Suite:\*\*** Weather Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- The app has access to the user's location.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker App.
2. Navigate to the weather section.
3. Verify that the app displays the current weather conditions, such as temperature, humidity, wind speed, and precipitation.
4. Verify that the app displays a weather forecast for the next 24 hours or more.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** The app displays the current weather conditions and a forecast for the user's location.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 2:\*\***

**\*\*Test Case ID:\*\*** TC\_Weather\_02

**\*\*Test Case Title:\*\*** Verify that users can view weather details for different locations.

**\*\*Test Case Description:\*\*** This test case verifies that the app allows users to select a different location and view weather details for that location.

**\*\*Test Suite:\*\*** Weather Integration

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.

**\*\*Test Data:\*\*** Different locations (e.g., New York City, London, Tokyo).

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker App.
2. Navigate to the weather section.
3. Search for a different location (e.g., New York City).
4. Select the chosen location.

5. Verify that the app displays the current weather conditions and forecast for the selected location.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** The app displays the current weather conditions and forecast for the selected location.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 3:\*\***

**\*\*Test Case ID:\*\*** TC\_Weather\_03

**\*\*Test Case Title:\*\*** Verify that weather data is updated in real-time.

**\*\*Test Case Description:\*\*** This test case verifies that weather data updates automatically and reflects changes in weather conditions.

**\*\*Test Suite:\*\*** Weather Integration

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- The app has access to the user's location.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker App.
2. Navigate to the weather section.
3. Observe the weather conditions for the user's location.
4. Wait for a significant amount of time (e.g., 15-30 minutes).
5. Refresh the weather section.
6. Verify that the weather data has been updated to reflect any changes in weather conditions.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** The weather data updates automatically to reflect changes in weather conditions.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case 4:\*\***

**\*\*Test Case ID:\*\*** TC\_Weather\_04

**\*\*Test Case Title:\*\*** Verify that the app offers recommendations based on weather conditions.

**\*\*Test Case Description:\*\*** This test case verifies that the app provides recommendations or suggestions for outdoor workouts based on current weather conditions.

**\*\*Test Suite:\*\*** Weather Integration

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in.
- The app has access to the user's location.

**\*\*Test Data:\*\*** No test data needed.

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker App.
2. Navigate to the weather section.
3. Observe the weather conditions.
4. Verify that the app provides recommendations or suggestions for outdoor activities based on the current weather conditions (e.g., "It's a perfect day for running", "Consider indoor activities today due to rain").

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** The app provides recommendations or suggestions for outdoor workouts based on the current weather conditions.

**\*\*Severity:\*\*** Minor



**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5:\*\***

**\*\*Test Case ID:\*\* TC\_Weather\_05**

**\*\*Test Case Title:\*\* Verify that users can receive weather alerts for extreme conditions.**

**\*\*Test Case Description:\*\* This test case verifies that the app sends notifications or alerts to the user when there are extreme weather conditions (e.g., heavy rain, thunderstorms, heatwave) in their location.**

**\*\*Test Suite:\*\* Weather Integration**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.
- The app has access to the user's location.
- User has enabled notifications for weather alerts.

**\*\*Test Data:\*\* Simulated extreme weather conditions.**

**\*\*Test Steps:\*\***

1. Open the Fitness Tracker App.
2. Navigate to settings and enable weather alerts.
3. Simulate extreme weather conditions (e.g., using a weather simulation tool or waiting for real-time extreme weather).
4. Verify that the app sends notifications or alerts to the user about the extreme weather conditions.

**\*\*Postconditions:\*\* None.**

**\*\*Expected Result:\*\* The app sends notifications or alerts to the user when there are extreme weather conditions.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**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

### \*\*Test Case 1\*\*

\*\*Test Case ID:\*\* FT-MT-001

\*\*Test Case Title:\*\* Verify that users can log their mood, energy levels, and motivation.

\*\*Test Case Description:\*\* This test case ensures that the app allows users to input their mood, energy levels, and motivation before and after workouts. It covers the first acceptance criteria.

\*\*Test Suite:\*\* Mood Tracking

\*\*Test Priority:\*\* High

\*\*Preconditions:\*\*

- User is logged in
- Workout has been completed

\*\*Test Data:\*\* No test data needed

\*\*Test Steps:\*\*

1. Navigate to the "Mood Tracker" section after completing a workout.
2. Select a mood level from the provided scale (e.g., Very Happy, Happy, Neutral, Sad, Very Sad).
3. Select an energy level from the provided scale (e.g., High, Medium, Low).
4. Select a motivation level from the provided scale (e.g., Very Motivated, Motivated, Neutral, Unmotivated, Very Unmotivated).
5. Submit the mood entry.

\*\*Postconditions:\*\*

- Mood entry is successfully saved.
- User is redirected to the main workout log or mood tracker section.

\*\*Expected Result:\*\* The app successfully saves the user's mood, energy, and motivation levels for the specified workout session.

\*\*Severity:\*\* Major

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 2\*\***

**\*\*Test Case ID:\*\* FT-MT-002**

**\*\*Test Case Title:\*\* Verify that the app provides insights into mood and energy trends over time.**

**\*\*Test Case Description:\*\* This test case checks if the app displays visual representations of the user's mood and energy levels over time, verifying the second acceptance criteria.**

**\*\*Test Suite:\*\* Mood Tracking**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in
- User has logged multiple mood entries over a period of time.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the "Mood & Energy Trends" section of the app.
2. Observe the visual representation (e.g., graphs, charts) of mood and energy levels.
3. Verify that the trends are displayed for the selected timeframe (e.g., daily, weekly, monthly).

**\*\*Postconditions:\*\***

- The user is able to view and interact with the mood and energy trends visualization.

**\*\*Expected Result:\*\* The app displays a clear visual representation of the user's mood and energy trends over time, providing insights into patterns and variations.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

### **\*\*Test Case 3\*\***

**\*\*Test Case ID:\*\* FT-MT-003**

**\*\*Test Case Title:\*\* Verify that users receive recommendations based on their mood and energy data.**

**\*\*Test Case Description:\*\* This test case validates that the app provides personalized recommendations based on the user's mood and energy data, fulfilling the third acceptance criteria.**

**\*\*Test Suite:\*\* Mood Tracking**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in
- User has logged multiple mood entries with varied mood and energy levels.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the "Recommendations" section of the app.
2. Observe the recommendations based on the user's past mood and energy data.
3. Check that the recommendations are tailored to the user's individual needs and patterns (e.g., suggesting workouts for low energy days, calming activities for high stress days).

**\*\*Postconditions:\*\***

- The user can view and interact with the recommendations.

**\*\*Expected Result:\*\* The app provides personalized recommendations based on the user's mood and energy data, aiming to improve their overall well-being and fitness experience.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

### **\*\*Test Case 4\*\***

**\*\*Test Case ID:\*\* FT-MT-004**

**\*\*Test Case Title:\*\* Verify that the app supports journaling for personal reflections on mood and workouts.**

**\*\*Test Case Description:\*\* This test case confirms that the app allows users to write personalized journal entries about their mood and workout experiences, satisfying the fourth acceptance criteria.**

**\*\*Test Suite:\*\* Mood Tracking**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in
- User has completed a workout and logged their mood.

**\*\*Test Data:\*\* No test data needed**

**\*\*Test Steps:\*\***

1. Navigate to the "Journal" section of the app.
2. Select the workout session for which the user wants to write a journal entry.
3. Enter a journal entry describing their mood, workout experience, and any related thoughts or reflections.
4. Save the journal entry.

**\*\*Postconditions:\*\***

- The journal entry is saved and associated with the specific workout session.

**\*\*Expected Result:\*\* The app successfully saves the user's journal entry, allowing them to track and reflect on their mood and workout experiences over time.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case 5\*\***

**\*\*Test Case ID:\*\* FT-MT-005**

**\*\*Test Case Title:\*\* Verify that users can set goals for improving mood and energy levels.**

**\*\*Test Case Description:\*\*** This test case verifies that the app allows users to set personalized goals for improving their mood and energy levels, fulfilling the fifth acceptance criteria.

**\*\*Test Suite:\*\*** Mood Tracking

**\*\*Test Priority:\*\*** Medium

**\*\*Preconditions:\*\***

- User is logged in

**\*\*Test Data:\*\*** No test data needed

**\*\*Test Steps:\*\***

1. Navigate to the "Goals" section of the app.
2. Select "Mood & Energy Goals".
3. Set a goal for improving mood (e.g., increase positive mood days, reduce negative mood days).
4. Set a goal for improving energy levels (e.g., increase high-energy days, reduce low-energy days).
5. Save the goals.

**\*\*Postconditions:\*\***

- The goals are saved and displayed in the user's profile or goals section.

**\*\*Expected Result:\*\*** The app allows users to set specific goals for improving their mood and energy levels, providing a framework for self-improvement and motivation.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 99**

**Project name -**

Recipe Sharing Platform

**Project description -**

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

**Feature name -**

Recipe Creation

**Feature description -**

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

**Userstory title -**

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

**Acceptance criteria -**

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', 'Recipes are saved in the user's profile.', 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

**Generated Test Cases:**

## Test Cases for Recipe Creation Feature

**\*\*Project Name:\*\* Recipe Sharing Platform**

**\*\*Feature Name:\*\* Recipe Creation**

**\*\*User Story Title:\*\* As a user, I want to create and share my own recipes so that I can showcase my culinary creations.**



**\*\*Acceptance Criteria:\*\***

1. Users can input and save ingredients and cooking instructions.
2. Users can upload photos for their recipes.
3. Recipes are saved in the user's profile.
4. Recipes are accessible to other users on the platform.
5. Users receive a confirmation upon successful recipe submission.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC\_RC\_01

**\*\*Test Case Title:\*\*** Verify that users can input and save ingredients and cooking instructions.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of the recipe creation form, ensuring users can successfully input and save ingredients and cooking instructions.

**\*\*Test Suite:\*\*** Recipe Creation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- User is on the recipe creation page.

**\*\*Test Data:\*\***

- Recipe title: "Chocolate Chip Cookies"
- Ingredients: "1 cup butter, 1 cup sugar, 2 eggs, 2 cups flour, 1 cup chocolate chips"
- Cooking Instructions: "Preheat oven to 375 degrees F. Cream together butter and sugar. Beat in eggs one at a time. Gradually add flour. Stir in chocolate chips. Drop by rounded tablespoons onto ungreased baking sheets. Bake for 10-12 minutes or until golden brown."

**\*\*Test Steps:\*\***

1. Enter the recipe title in the designated field.

2. Input the ingredients in the provided field, separating them by commas.
3. Enter the cooking instructions in the dedicated field.
4. Click on the "Save Recipe" button.

**\*\*Postconditions:\*\***

- The recipe is saved.
- The user is redirected to the recipe details page.

**\*\*Expected Result:\*\*** The system successfully saves the recipe with the input ingredients and cooking instructions.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_RC\_02

**\*\*Test Case Title:\*\*** Verify that users can upload photos for their recipes.

**\*\*Test Case Description:\*\*** This test case verifies the functionality of the image upload feature for recipes, ensuring users can successfully upload photos for their creations.

**\*\*Test Suite:\*\*** Recipe Creation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- User is on the recipe creation page.

**\*\*Test Data:\*\***

- Image file: A valid image file (e.g., .jpg, .png)

**\*\*Test Steps:\*\***

1. Click on the "Upload Photo" button.
2. Select the image file from your device.
3. Upload the image.

**\*\*Postconditions:\*\***

- The image is uploaded successfully.
- The image is displayed on the recipe details page.

**\*\*Expected Result:\*\*** The system successfully uploads the chosen image and displays it with the recipe.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_RC\_03

**\*\*Test Case Title:\*\*** Verify that recipes are saved in the user's profile.

**\*\*Test Case Description:\*\*** This test case ensures that the created recipe is saved and accessible from the user's profile page.

**\*\*Test Suite:\*\*** Recipe Creation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- User has created a recipe.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Go to the user's profile page.
2. Navigate to the "My Recipes" section.

**\*\*Postconditions:\*\***

- The created recipe is visible in the "My Recipes" section.

**\*\*Expected Result:\*\*** The created recipe is successfully saved in the user's profile and visible in the "My Recipes" section.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RC\_04**

**\*\*Test Case Title:\*\* Verify that recipes are accessible to other users on the platform.**

**\*\*Test Case Description:\*\* This test case verifies that the created recipe is publicly accessible to other users of the platform, ensuring its visibility and discoverability.**

**\*\*Test Suite:\*\* Recipe Creation**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in.
- User has created a recipe.
- Another user is logged in.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Log out as the recipe creator.
2. Log in as a different user.
3. Search for the recipe title or browse the recipe catalog.

**\*\*Postconditions:\*\***

- The recipe is displayed in the search results or recipe catalog.

**\*\*Expected Result:\*\* The created recipe is successfully accessible to other users on the platform.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RC\_05**

**\*\*Test Case Title:\*\*** Verify that users receive a confirmation upon successful recipe submission.

**\*\*Test Case Description:\*\*** This test case ensures that the user receives a confirmation message upon successfully submitting their created recipe.

**\*\*Test Suite:\*\*** Recipe Creation

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in.
- User has created a recipe.

**\*\*Test Data:\*\***

- No test data needed.

**\*\*Test Steps:\*\***

1. Click on the "Save Recipe" button.
2. Observe the message displayed on the screen.

**\*\*Postconditions:\*\***

- The recipe is saved.

**\*\*Expected Result:\*\*** The system displays a confirmation message indicating the successful submission of the recipe.

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

-----

**Data - 100**

**Project name -**

Recipe Sharing Platform

**Project description -**

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

**Feature name -**

Recipe Discovery

**Feature description -**

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

**Userstory title -**

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

**Acceptance criteria -**

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

**Generated Test Cases:**

## Test Cases for Recipe Discovery Feature

**\*\*Project Name:\*\* Recipe Sharing Platform**

**\*\*Feature Name:\*\* Recipe Discovery**

**\*\*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.**

**\*\*Acceptance Criteria:\*\***

1. Users can search for recipes using keywords and ingredients.
2. Search results are relevant and accurately match the search criteria.
3. Users can filter search results by cuisine type, dietary preferences, and other criteria.
4. Recipes in the search results are accessible and detailed.
5. Users can save search results for future reference.

**\*\*Test Case Creation Template:\*\***

**\*\*Test Case ID:\*\*** TC\_RD\_01

**\*\*Test Case Title:\*\*** Verify that users can search for recipes using keywords and ingredients.

**\*\*Test Case Description:\*\*** This test case validates that the platform allows users to search for recipes based on keywords and ingredients.

**\*\*Test Suite:\*\*** Recipe Discovery

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the platform.

**\*\*Test Data:\*\***

- Keyword: "Chicken Curry"

- Ingredient: "Tomato"

**\*\*Test Steps:\*\***

1. Go to the recipe search page.
2. Enter the keyword "Chicken Curry" in the search bar.
3. Enter the ingredient "Tomato" in the ingredient filter.
4. Click on the search button.

**\*\*Postconditions:\*\***

- The search results should display recipes that contain both "Chicken Curry" and "Tomato".

**\*\*Expected Result:\*\*** The system should display relevant recipes that match the search criteria, including recipes containing "Chicken Curry" and "Tomato".

**\*\*Severity:\*\*** Major

**\*\*Type of Testing:\*\*** Functional

**\*\*Test Case Approach:\*\*** Positive

**\*\*Test Case ID:\*\*** TC\_RD\_02

**\*\*Test Case Title:\*\*** Verify that search results are relevant and accurately match the search criteria.

**\*\*Test Case Description:\*\*** This test case checks if the search results are relevant to the search criteria and do not include irrelevant recipes.

**\*\*Test Suite:\*\*** Recipe Discovery

**\*\*Test Priority:\*\*** High

**\*\*Preconditions:\*\***

- User is logged in to the platform.

**\*\*Test Data:\*\***

- Keyword: "Chocolate Cake"

**\*\*Test Steps:\*\***

1. Go to the recipe search page.
2. Enter the keyword "Chocolate Cake" in the search bar.
3. Click on the search button.
4. Verify the displayed recipes.

**\*\*Postconditions:\*\***

- The search results should only display recipes containing "Chocolate Cake" as the main ingredient.

**\*\*Expected Result:\*\*** The system should only display relevant recipes that include "Chocolate Cake" as the main ingredient and exclude recipes for other desserts or dishes.



**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RD\_03**

**\*\*Test Case Title:\*\* Verify that users can filter search results by cuisine type, dietary preferences, and other criteria.**

**\*\*Test Case Description:\*\* This test case checks the functionality of filters to refine search results based on cuisine type, dietary restrictions, and other preferences.**

**\*\*Test Suite:\*\* Recipe Discovery**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in to the platform.

**\*\*Test Data:\*\***

- Cuisine: "Italian"

- Dietary Preference: "Vegetarian"

**\*\*Test Steps:\*\***

1. Go to the recipe search page.
2. Enter a general keyword like "Pasta" in the search bar.
3. Select "Italian" from the cuisine filter.
4. Select "Vegetarian" from the dietary preference filter.
5. Click on the search button.

**\*\*Postconditions:\*\***

- The search results should display only Italian vegetarian pasta recipes.

**\*\*Expected Result:\*\* The system should display only relevant recipes that match both "Italian" cuisine and "Vegetarian" dietary restrictions.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RD\_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 recipes in search results are accessible and provide sufficient information about the dish.**

**\*\*Test Suite:\*\* Recipe Discovery**

**\*\*Test Priority:\*\* High**

**\*\*Preconditions:\*\***

- User is logged in to the platform.

**\*\*Test Data:\*\***

- Search result recipe: "Lemon Chicken"

**\*\*Test Steps:\*\***

1. Go to the recipe search page.
2. Enter a keyword like "Lemon Chicken" in the search bar.
3. Click on the search button.
4. Select a "Lemon Chicken" recipe from the search results.
5. Verify the displayed information including ingredients, instructions, cooking time, and servings.

**\*\*Postconditions:\*\***

- The selected recipe details are accessible and provide sufficient information.

**\*\*Expected Result:\*\* The system should display a detailed recipe with all relevant information like ingredients, step-by-step instructions, cooking time, serving size, etc.**

**\*\*Severity:\*\* Major**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

**\*\*Test Case ID:\*\* TC\_RD\_05**

**\*\*Test Case Title:\*\* Verify that users can save search results for future reference.**

**\*\*Test Case Description:\*\* This test case checks if users can save their search results for easy access later.**

**\*\*Test Suite:\*\* Recipe Discovery**

**\*\*Test Priority:\*\* Medium**

**\*\*Preconditions:\*\***

- User is logged in to the platform.

**\*\*Test Data:\*\***

- Saved Search: "Vegan Breakfast Recipes"

**\*\*Test Steps:\*\***

1. Go to the recipe search page.
2. Enter the keyword "Vegan Breakfast Recipes" in the search bar.
3. Click on the search button.
4. Click on the "Save Search" button.
5. Navigate to the "Saved Searches" section.
6. Verify that the "Vegan Breakfast Recipes" search is saved.

**\*\*Postconditions:\*\***

- The saved search is accessible in the "Saved Searches" section.

**\*\*Expected Result:\*\* The system should allow users to save their search results and access them later from the "Saved Searches" section.**

**\*\*Severity:\*\* Minor**

**\*\*Type of Testing:\*\* Functional**

**\*\*Test Case Approach:\*\* Positive**

-----