

# Generated Test Cases via llama3-70b-8192 LLM Model and Zero-shot Prompting Technique

---

## Data - 1

### Project name -

Social Media Platform

### Project description -

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

### Feature name -

User Profiles

### Feature description -

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

### Userstory title -

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

### Acceptance criteria -

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

### Generated Test Cases:

Here are 5 detailed test cases for the User Profiles feature:

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

Test Case ID: UP-001

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

Test Case Description: This test case verifies that users can add and edit their name on their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

? User is logged in to the platform.

? User has not added a name to their profile before.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user.
2. Navigate to the profile page.
3. Click on the "Edit" button next to the name field.
4. Enter a new name and save changes.
5. Verify that the new name is displayed on the profile page.

Postconditions:

? The new name is saved successfully.

Expected Result: The user's name is updated successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can add and edit their profile picture\*\***

Test Case ID: UP-002

Test Case Title: Verify that users can add and edit their profile picture

Test Case Description: This test case verifies that users can add and edit their profile picture on their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

? User is logged in to the platform.

? User has not added a profile picture before.

Test Data: A valid image file (e.g., jpg, png, gif)

Test Steps:

1. Log in to the platform as a user.
2. Navigate to the profile page.
3. Click on the "Edit" button next to the profile picture field.
4. Upload a new profile picture and save changes.
5. Verify that the new profile picture is displayed on the profile page.

Postconditions:

? The new profile picture is saved successfully.

Expected Result: The user's profile picture is updated successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can set privacy settings for each field\*\***

Test Case ID: UP-003

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

Test Case Description: This test case verifies that users can set privacy settings for each field on their profile.

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

? User is logged in to the platform.

? User has added data to their profile fields (e.g., name, bio, location, interests).

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user.
2. Navigate to the profile page.
3. Click on the "Privacy" button next to a profile field (e.g., name).
4. Select a privacy setting (e.g., public, friends, private) and save changes.
5. Verify that the selected privacy setting is applied to the profile field.

Postconditions:

? The privacy setting is saved successfully.

Expected Result: The user's privacy setting is updated successfully.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users cannot add an unsupported image format as their profile picture\*\***

Test Case ID: UP-004

Test Case Title: Verify that users cannot add an unsupported image format as their profile picture

Test Case Description: This test case verifies that users cannot add an unsupported image format as their

profile picture.

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

? User is logged in to the platform.

? User has not added a profile picture before.

Test Data: An invalid image file (e.g., pdf, docx)

Test Steps:

1. Log in to the platform as a user.
2. Navigate to the profile page.
3. Click on the "Edit" button next to the profile picture field.
4. Upload an unsupported image file and try to save changes.
5. Verify that an error message is displayed.

Postconditions:

? An error message is displayed.

Expected Result: The user cannot add an unsupported image format as their profile picture.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

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

Test Case ID: UP-005

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

Test Case Description: This test case verifies that users receive a confirmation upon saving their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

? User is logged in to the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user.
2. Make changes to the profile (e.g., add a new name, bio, profile picture).
3. Click on the "Save" button.
4. Verify that a confirmation message is displayed.

Postconditions:

? The profile is saved successfully.

Expected Result: The user receives a confirmation message upon saving their profile.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 2**

**Project name -**

Social Media Platform

**Project description -**

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

**Feature name -**

News Feed

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

Here are five detailed test cases for the News Feed feature:

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

Test Case ID: NF001

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

Test Case Description: This test case verifies that the news feed only displays posts from accounts that the user has followed.

Test Suite: News Feed

Test Priority: High

Preconditions:

\* The user has at least one followed account

\* The followed account has posted content

Test Data: No test data needed

Test Steps:

1. Log in as a user and navigate to the news feed page
2. Verify that only posts from followed accounts are displayed
3. Verify that posts from non-followed accounts are not displayed
4. Verify that the feed is updated in real-time when a new post is made by a followed account

Postconditions:

\* Verify that the user's followed accounts list is updated correctly

\* Verify that the news feed is refreshed automatically

Expected Result: The news feed only displays posts from followed accounts

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: NF002

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

Test Case Description: This test case verifies that the news feed displays content in either chronological or



algorithmic order.

Test Suite: News Feed

Test Priority: Medium

Preconditions:

- \* The user has multiple followed accounts with multiple posts

- \* The posts have different timestamps

Test Data: No test data needed

Test Steps:

1. Log in as a user and navigate to the news feed page
2. Verify that the posts are displayed in either chronological or algorithmic order
3. Verify that the timestamp of each post is correct
4. Verify that the feed is updated correctly when a new post is made by a followed account

Postconditions:

- \* Verify that the user's news feed preferences are updated correctly

- \* Verify that the news feed is refreshed automatically

Expected Result: The news feed displays content in either chronological or algorithmic order

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: NF003

Test Case Title: Verify that users can interact with posts

Test Case Description: This test case verifies that users can interact with posts in the news feed by liking, commenting, and sharing.

Test Suite: News Feed

Test Priority: High

Preconditions:

- \* The user has multiple followed accounts with multiple posts
- \* The user is logged in

Test Data: No test data needed

Test Steps:

1. Log in as a user and navigate to the news feed page
2. Select a post and click the like button
3. Verify that the like is registered correctly
4. Select a post and click the comment button
5. Verify that the comment is registered correctly
6. Select a post and click the share button
7. Verify that the share is registered correctly

Postconditions:

- \* Verify that the post's like, comment, and share counts are updated correctly

\* Verify that the user's news feed is refreshed automatically

Expected Result: Users can interact with posts in the news feed

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: NF004

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

Test Case Description: This test case verifies that the news feed updates in real-time when a new post is made by a followed account.

Test Suite: News Feed

Test Priority: High

Preconditions:

- \* The user has at least one followed account
- \* The followed account has posted content
- \* The user is logged in

Test Data: No test data needed

Test Steps:

1. Log in as a user and navigate to the news feed page
2. Have a followed account make a new post

3. Verify that the new post is displayed in the news feed in real-time
4. Verify that the feed is updated automatically

Postconditions:

- \* Verify that the user's news feed is refreshed automatically
- \* Verify that the followed account's post count is updated correctly

Expected Result: The news feed updates in real-time when a new post is made by a followed account

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: NF005

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

Test Case Description: This test case verifies that users can filter content in the news feed based on type or topic.

Test Suite: News Feed

Test Priority: Medium

Preconditions:

- \* The user has multiple followed accounts with multiple posts
- \* The posts have different types (e.g., photo, video, text) and topics

Test Data: No test data needed

Test Steps:

1. Log in as a user and navigate to the news feed page
2. Click on a filter option (e.g., photos, videos, topic)
3. Verify that only posts that match the filter option are displayed
4. Verify that the feed is updated correctly when a new post is made by a followed account that matches the filter option

Postconditions:

- \* Verify that the user's news feed preferences are updated correctly
- \* Verify that the news feed is refreshed automatically

Expected Result: Users can filter content in the news feed based on type or topic

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 3**

**Project name -**

Social Media Platform

**Project description -**

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

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

**Feature name -**

Messaging

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

Here are the detailed test cases for the Messaging feature of the Social Media Platform:

**\*\*Test Case 1: Sending a Text Message\*\***

Test Case ID: MSG-001

Test Case Title: Verify that users can send text messages to each other.

Test Case Description: This test case ensures that users can send text messages to each other in real-time.

Test Suite: Messaging

Test Priority: High

Preconditions:

\* User A and User B are friends on the social media platform.

\* User A and User B are online.

Test Data: No test data needed

Test Steps:

1. User A logs in to their account and navigates to the messaging feature.
2. User A starts a new conversation with User B.
3. User A types a text message and sends it to User B.
4. Verify that the message is delivered to User B's inbox in real-time.
5. User B logs in to their account and verifies that they have received the message.

Postconditions:

\* The message is marked as "delivered" in User A's chat window.

\* The message is displayed in User B's chat window.

Expected Result: The text message is successfully sent from User A to User B in real-time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Sending an Image\*\***

Test Case ID: MSG-002

Test Case Title: Verify that users can send images to each other.

Test Case Description: This test case ensures that users can send images to each other in real-time.

Test Suite: Messaging

Test Priority: High

Preconditions:

- \* User A and User B are friends on the social media platform.
- \* User A and User B are online.
- \* User A has a valid image file to send.

Test Data: Image file (e.g., JPEG, PNG)

Test Steps:

1. User A logs in to their account and navigates to the messaging feature.
2. User A starts a new conversation with User B.
3. User A selects an image file to send and attaches it to the message.
4. User A sends the message with the attached image to User B.
5. Verify that the message is delivered to User B's inbox in real-time.
6. User B logs in to their account and verifies that they have received the message with the attached image.

Postconditions:

- \* The message is marked as "delivered" in User A's chat window.
- \* The image is displayed in User B's chat window.



Expected Result: The image is successfully sent from User A to User B in real-time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Sending a Video\*\***

Test Case ID: MSG-003

Test Case Title: Verify that users can send videos to each other.

Test Case Description: This test case ensures that users can send videos to each other in real-time.

Test Suite: Messaging

Test Priority: High

Preconditions:

- \* User A and User B are friends on the social media platform.
- \* User A and User B are online.
- \* User A has a valid video file to send.

Test Data: Video file (e.g., MP4, AVI)

Test Steps:

1. User A logs in to their account and navigates to the messaging feature.
2. User A starts a new conversation with User B.

3. User A selects a video file to send and attaches it to the message.
4. User A sends the message with the attached video to User B.
5. Verify that the message is delivered to User B's inbox in real-time.
6. User B logs in to their account and verifies that they have received the message with the attached video.

Postconditions:

- \* The message is marked as "delivered" in User A's chat window.
- \* The video is displayed in User B's chat window.

Expected Result: The video is successfully sent from User A to User B in real-time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Read Receipts\*\***

Test Case ID: MSG-004

Test Case Title: Verify that users can see read receipts for their messages.

Test Case Description: This test case ensures that users can see read receipts for their messages.

Test Suite: Messaging

Test Priority: Medium

Preconditions:

- \* User A and User B are friends on the social media platform.

\* User A has sent a message to User B.

Test Data: No test data needed

Test Steps:

1. User A logs in to their account and navigates to the messaging feature.
2. User A sends a message to User B.
3. User B logs in to their account and reads the message.
4. Verify that the message is marked as "read" in User A's chat window.
5. Verify that the message is marked as "read" in User B's chat window.

Postconditions:

\* The message is marked as "read" in both User A's and User B's chat windows.

Expected Result: The read receipt is successfully displayed for the message.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: New Message Notifications\*\***

Test Case ID: MSG-005

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

Test Case Description: This test case ensures that users receive notifications for new messages.

Test Suite: Messaging

Test Priority: Medium

Preconditions:

- \* User A and User B are friends on the social media platform.

- \* User A is logged out of their account.

Test Data: No test data needed

Test Steps:

1. User B logs in to their account and navigates to the messaging feature.
2. User B sends a message to User A.
3. Verify that User A receives a notification for the new message.
4. User A logs in to their account and verifies that they have received the message.

Postconditions:

- \* The message is displayed in User A's inbox.

Expected Result: The notification for the new message is successfully sent to User A.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

---

## **Data - 4**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Content Sharing

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

Here are five test cases for the Content Sharing feature of the Social Media Platform:

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

\* Test Case ID: TSC\_001

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

\* Test Case Description: This test case ensures that users can create and share text posts successfully.

\* Test Suite: Content Sharing

\* Test Priority: High

\* Preconditions:

+ User is logged in to the social media platform

+ User has followers

\* Test Data: No test data needed

\* Test Steps:

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

2. Create a new text post

3. Enter some text in the post editor

4. Click the "Share" button

5. Verify that the post is shared successfully and appears in the user's news feed

\* Postconditions:

+ Verify that the post is visible to the user's followers

\* Expected Result: The text post is shared successfully and visible to the user's followers

\* Severity: Critical

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can share photos\*\***

\* Test Case ID: TSC\_002

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

\* Test Case Description: This test case ensures that users can share photos successfully.

\* Test Suite: Content Sharing

\* Test Priority: High

\* Preconditions:

+ User is logged in to the social media platform

+ User has followers

+ User has a photo to upload

\* Test Data: Sample photo to upload

\* Test Steps:

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

2. Create a new post and select the photo upload option

3. Upload a sample photo

4. Click the "Share" button

5. Verify that the photo post is shared successfully and appears in the user's news feed

\* Postconditions:

+ Verify that the photo post is visible to the user's followers

\* Expected Result: The photo post is shared successfully and visible to the user's followers

\* Severity: Critical

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

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

\* Test Case ID: TSC\_003

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

\* Test Case Description: This test case ensures that users can edit their posts successfully.

\* Test Suite: Content Sharing

\* Test Priority: Medium

\* Preconditions:

+ User is logged in to the social media platform

+ User has a previously shared post

\* Test Data: No test data needed

\* Test Steps:

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

2. Go to the user's previously shared post

3. Click the "Edit" button

4. Make some changes to the post content

5. Click the "Save" button

6. Verify that the post is updated successfully

\* Postconditions:

+ Verify that the updated post is visible to the user's followers

\* Expected Result: The post is edited successfully and visible to the user's followers

\* Severity: Major

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can delete their posts\*\***

\* Test Case ID: TSC\_004

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

\* Test Case Description: This test case ensures that users can delete their posts successfully.

\* Test Suite: Content Sharing

\* Test Priority: Medium

\* Preconditions:



+ User is logged in to the social media platform

+ User has a previously shared post

\* Test Data: No test data needed

\* Test Steps:

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

2. Go to the user's previously shared post

3. Click the "Delete" button

4. Confirm deletion

5. Verify that the post is deleted successfully

\* Postconditions:

+ Verify that the post is no longer visible to the user's followers

\* Expected Result: The post is deleted successfully and no longer visible to the user's followers

\* Severity: Major

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

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

\* Test Case ID: TSC\_005

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

\* Test Case Description: This test case ensures that users receive feedback through likes, comments, and shares on their posts.

\* Test Suite: Content Sharing

\* Test Priority: High

\* Preconditions:

+ User is logged in to the social media platform

+ User has a previously shared post

+ Other users are logged in to the social media platform

\* Test Data: No test data needed

\* Test Steps:

1. Log in to the social media platform as a user
2. Share a new post
3. Log in as another user
4. Like the post
5. Log in as yet another user
6. Comment on the post
7. Log in as yet another user
8. Share the post
9. Verify that the original user receives feedback through likes, comments, and shares

\* Postconditions:

+ Verify that the original user can view the feedback on their post

\* Expected Result: The original user receives feedback through likes, comments, and shares on their post

\* Severity: Critical

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

-----

## **Data - 5**

### **Project name -**

Social Media Platform

### **Project description -**

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

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

**Feature name -**

Privacy Settings

**Feature description -**

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

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

**Userstory title -**

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

**Acceptance criteria -**

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

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

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

**Generated Test Cases:**

Here are five detailed test cases for the Privacy Settings feature, using the provided template:

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

Test Case ID: PS-001

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

Test Case Description: This test case verifies that users can successfully set their profile to public, private, or restricted privacy settings.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

\* User is logged in to their account

\* User has not previously set their profile privacy settings

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user.
2. Navigate to the user's profile page.
3. Click on the "Privacy Settings" button.
4. Select "Public" as the profile privacy setting.
5. Verify that the profile is updated to reflect the new privacy setting.

Postconditions:

\* Profile privacy setting is updated to "Public"

\* Other users can view the user's profile without restrictions

Expected Result: The user's profile is successfully set to public, and other users can view their profile without restrictions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that privacy settings can be changed at any time\*\***

Test Case ID: PS-002

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

Test Case Description: This test case verifies that users can change their privacy settings to a different option (public, private, or restricted) at any time.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- \* User is logged in to their account
- \* User has previously set their profile privacy setting to "Private"

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user.
2. Navigate to the user's profile page.
3. Click on the "Privacy Settings" button.
4. Select "Restricted" as the new profile privacy setting.
5. Verify that the profile is updated to reflect the new privacy setting.

Postconditions:

- \* Profile privacy setting is updated to "Restricted"
- \* Other users are restricted from viewing the user's profile

Expected Result: The user's profile privacy setting is successfully changed to "Restricted", and other users are restricted from viewing their profile.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: PS-003

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

Test Case Description: This test case verifies that users can set individual posts to different privacy settings (public, private, or restricted).

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

\* User is logged in to their account

\* User has created a new post

Test Data: Post content with varying privacy settings (public, private, restricted)

Test Steps:

1. Log in to the social media platform as a registered user.
2. Create a new post with varying privacy settings (public, private, restricted).
3. Verify that the post is visible to the intended audience based on the selected privacy setting.

Postconditions:

- \* Post is visible to the intended audience based on the selected privacy setting

Expected Result: The user's individual post is successfully set to the selected privacy setting, and only the intended audience can view the post.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: PS-004

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

Test Case Description: This test case verifies that the system immediately respects the user's privacy settings and restricts access to their profile or posts accordingly.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- \* User is logged in to their account

- \* User has set their profile privacy setting to "Private"

Test Data: No test data needed

#### Test Steps:

1. Log in to the social media platform as a registered user.
2. Set the user's profile privacy setting to "Private".
3. Simultaneously, log in as another user who is not a friend of the original user.
4. Attempt to view the original user's profile.
5. Verify that access is restricted.

#### Postconditions:

\* Access to the original user's profile is restricted

Expected Result: The system immediately respects the user's privacy setting and restricts access to their profile.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: PS-005

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

Test Case Description: This test case verifies that users receive a notification when their privacy settings are changed, including changes to individual post privacy settings.

Test Suite: Privacy Settings

Test Priority: Medium



#### Preconditions:

- \* User is logged in to their account
- \* User has set their profile privacy setting to "Public"

Test Data: No test data needed

#### Test Steps:

1. Log in to the social media platform as a registered user.
2. Change the user's profile privacy setting to "Private".
3. Verify that the user receives a notification of the change in privacy setting.
4. Create a new post with varying privacy settings (public, private, restricted).
5. Verify that the user receives a notification of the change in post privacy setting.

#### Postconditions:

- \* User receives a notification of the change in privacy setting

Expected Result: The user receives a notification when their privacy settings are changed, including changes to individual post privacy settings.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

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:**

Here are five detailed test cases for the Notifications feature of the Social Media Platform project:

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

Test Case ID: NOT-01

Test Case Title: Verify that users receive notifications for likes

Test Case Description: This test case verifies that users receive notifications when someone likes their post.

Test Suite: Notifications

Test Priority: High

Preconditions:

\* The user has created a post on the platform

\* Another user is logged in and has access to the post

Test Data: No test data needed

Test Steps:

1. Log in as the user who created the post
2. Verify that no notification is displayed
3. Log in as the other user and like the post
4. Log back in as the original user and verify that a notification is displayed indicating that the post has been liked

Postconditions: None

Expected Result: The user receives a notification for the like on their post

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: NOT-02

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

Test Case Description: This test case verifies that users receive notifications when someone sends them a friend request.

Test Suite: Notifications

Test Priority: High

Preconditions:

- \* The user has not received any friend requests
- \* Another user is logged in and has the ability to send friend requests

Test Data: No test data needed

Test Steps:

1. Log in as the user
2. Verify that no notification is displayed
3. Log in as the other user and send a friend request to the original user
4. Log back in as the original user and verify that a notification is displayed indicating a new friend request

Postconditions: None

Expected Result: The user receives a notification for the new friend request

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that notifications are displayed in real-time\*\***

Test Case ID: NOT-03

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

Test Case Description: This test case verifies that notifications are displayed to the user in real-time, without requiring a page reload.

Test Suite: Notifications

Test Priority: Medium

Preconditions:

\* The user is logged in and has access to their notification feed

Test Data: No test data needed

Test Steps:

1. Log in as the user
2. Verify that no notification is displayed
3. Have another user like a post or send a friend request to the original user
4. Verify that the notification is displayed in real-time, without requiring a page reload

Postconditions: None

Expected Result: The user receives the notification in real-time, without requiring a page reload

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: NOT-04

Test Case Title: Verify that users can customize notification preferences

Test Case Description: This test case verifies that users can customize their notification preferences, such as choosing which types of notifications they receive.

Test Suite: Notifications

Test Priority: Medium

Preconditions:

\* The user is logged in and has access to their notification settings

Test Data: No test data needed

Test Steps:

1. Log in as the user
2. Access the notification settings
3. Customize the notification preferences (e.g., select to only receive notifications for likes, not comments)
4. Verify that the notifications are updated according to the customized preferences

Postconditions: None

Expected Result: The user's notification preferences are updated successfully

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: NOT-05

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

Test Case Description: This test case verifies that notifications include a link to the relevant activity, such as the post or comment that triggered the notification.

Test Suite: Notifications

Test Priority: Medium

Preconditions:

\* The user has received a notification (e.g., for a like or comment)

Test Data: No test data needed

Test Steps:

1. Log in as the user
2. Access the notification feed
3. Click on a notification (e.g., for a like or comment)
4. Verify that the link takes the user to the relevant activity (e.g., the post or comment that triggered the notification)

Postconditions: None

Expected Result: The notification includes a link to the relevant activity

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

---

## **Data - 7**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Search Functionality

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

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

**\*\*Test Case 1: Verify that search functionality returns relevant results for keywords\*\***



Test Case ID: SFTC001

Test Case Title: Verify that search functionality returns relevant results for keywords

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- \* User is logged in to the platform
- \* Search bar is available on the dashboard
- \* Test data: Keyword "social media" and related content exist on the platform

Test Data: No test data needed

Test Steps:

1. Type "social media" in the search bar and press Enter.
2. Verify that search results include profiles, posts, and topics related to "social media".
3. Verify that results are displayed in order of relevance, with most relevant results appearing at the top.

Postconditions:

- \* Search results are cleared
- \* Search bar is reset

Expected Result: Search results are displayed, including profiles, posts, and topics related to "social media", in order of relevance.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that search results can be filtered by content type\*\***

Test Case ID: SFTC002

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

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- \* User is logged in to the platform
- \* Search bar is available on the dashboard
- \* Test data: Keyword "football" and related content exist on the platform

Test Data: No test data needed

Test Steps:

1. Type "football" in the search bar and press Enter.
2. Verify that search results include profiles, posts, and topics related to "football".
3. Click on the "Posts" filter button and verify that only post results are displayed.
4. Click on the "Profiles" filter button and verify that only profile results are displayed.

Postconditions:

- \* Search results are cleared

\* Search bar is reset

Expected Result: Search results can be filtered by content type (posts, profiles, topics).

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that search functionality provides suggestions as the user types\*\***

Test Case ID: SFTC003

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

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

\* User is logged in to the platform

\* Search bar is available on the dashboard

Test Data: No test data needed

Test Steps:

1. Type "soc" in the search bar and wait for suggestions to appear.
2. Verify that suggestions include keywords, hashtags, and profiles related to "soc".
3. Select a suggested keyword and verify that it is added to the search bar.

Postconditions:

- \* Search results are cleared

- \* Search bar is reset

Expected Result: Search functionality provides suggestions as the user types.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that search functionality handles invalid input\*\***

Test Case ID: SFTC004

Test Case Title: Verify that search functionality handles invalid input

Test Suite: Search Functionality

Test Priority: Low

Preconditions:

- \* User is logged in to the platform

- \* Search bar is available on the dashboard

Test Data: No test data needed

Test Steps:

1. Type "invalidinput%" in the search bar and press Enter.

2. Verify that an error message is displayed, indicating that no results were found.
3. Verify that the search bar is reset.

Postconditions:

- \* Search results are cleared

- \* Search bar is reset

Expected Result: Search functionality handles invalid input by displaying an error message and resetting the search bar.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Negative

**\*\*Test Case 5: Verify that search results are displayed in order of relevance\*\***

Test Case ID: SFTC005

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

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- \* User is logged in to the platform

- \* Search bar is available on the dashboard

- \* Test data: Keyword "social media" and related content exist on the platform

Test Data: No test data needed

Test Steps:

1. Type "social media" in the search bar and press Enter.
2. Verify that search results are displayed in order of relevance, with most relevant results appearing at the top.
3. Verify that the top result is a popular profile related to "social media".

Postconditions:

\* Search results are cleared

\* Search bar is reset

Expected Result: Search results are displayed in order of relevance, with most relevant results appearing at the top.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 8**

**Project name -**

Social Media Platform

**Project description -**

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

**Feature name -**

Hashtags and Tagging

**Feature description -**

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

**Userstory title -**

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

**Acceptance criteria -**

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

**Generated Test Cases:**

Here are five detailed test cases for the Hashtags and Tagging feature, using the provided Test Case Creation Template:

**\*\*Test Case 1: Verify that users can add hashtags to their posts\*\***

Test Case ID: TCHT001

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

Test Case Description: This test case verifies that users can add relevant hashtags to their posts, making their content more discoverable.

Test Suite: Social Media Platform - Hashtags and Tagging

Test Priority: High

Preconditions:

- \* User is logged in to their account
- \* User has created a new post with text content
- \* Hashtags are enabled on the platform

Test Data: No test data needed

Test Steps:

1. Log in to the user account and navigate to the post creation page
2. Enter text content for the post and add a relevant hashtag (e.g., #socialmedia)
3. Click the "Post" button to publish the post
4. Verify that the hashtag is displayed alongside the post content

Postconditions:

- \* The post is successfully published with the added hashtag
- \* The hashtag is clickable and leads to related content

Expected Result: The user can add hashtags to their posts successfully, making their content more discoverable.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive



**\*\*Test Case 2: Verify that users can tag other profiles, brands, or locations\*\***

Test Case ID: TCHT002

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, increasing the visibility and reach of their content.

Test Suite: Social Media Platform - Hashtags and Tagging

Test Priority: High

Preconditions:

- \* User is logged in to their account
- \* User has created a new post with text content
- \* Tagging feature is enabled on the platform

Test Data: Create a new user profile, brand, and location for tagging

Test Steps:

1. Log in to the user account and navigate to the post creation page
2. Enter text content for the post and type the "@" symbol followed by the name of a user, brand, or location (e.g., @JohnDoe)
3. Select the suggested profile, brand, or location from the dropdown list
4. Click the "Post" button to publish the post
5. Verify that the tagged profile, brand, or location is displayed alongside the post content

Postconditions:

- \* The post is successfully published with the tagged profile, brand, or location
- \* The tagged profile, brand, or location is clickable and leads to their respective profiles

Expected Result: The user can tag other profiles, brands, or locations in their posts successfully, increasing the visibility and reach of their content.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that hashtags and tags are clickable and lead to related content\*\***

Test Case ID: TCHT003

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 direct users to related content, enhancing the discoverability of posts.

Test Suite: Social Media Platform - Hashtags and Tagging

Test Priority: Medium

Preconditions:

- \* User is logged in to their account
- \* A post with a hashtag and tag is published

Test Data: No test data needed

Test Steps:

1. Log in to the user account and navigate to the post with a hashtag and tag

2. Click on the hashtag (e.g., #socialmedia)
3. Verify that the system redirects to a page with related content (e.g., posts with the same hashtag)
4. Click on the tagged profile, brand, or location
5. Verify that the system redirects to the profile, brand, or location page

Postconditions:

\* The system successfully redirects to related content for hashtags and tags

Expected Result: Hashtags and tags are clickable and lead to related content, enhancing the discoverability of posts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can search for content using hashtags\*\***

Test Case ID: TCHT004

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, making it easy to find related posts.

Test Suite: Social Media Platform - Hashtags and Tagging

Test Priority: High

Preconditions:

\* User is logged in to their account

\* Multiple posts with different hashtags are published

Test Data: No test data needed

Test Steps:

1. Log in to the user account and navigate to the search page
2. Enter a hashtag (e.g., #socialmedia) in the search bar
3. Click the "Search" button
4. Verify that the system displays a list of posts with the same hashtag
5. Verify that the posts are sorted in descending order of relevance (e.g., most recent posts first)

Postconditions:

\* The system successfully displays a list of posts with the same hashtag

Expected Result: Users can search for content using hashtags and find related posts easily.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the system suggests popular hashtags as users type\*\***

Test Case ID: TCHT005

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, helping them discover relevant and trending topics.

Test Suite: Social Media Platform - Hashtags and Tagging

Test Priority: Medium

Preconditions:

- \* User is logged in to their account
- \* User is creating a new post with text content

Test Data: No test data needed

Test Steps:

1. Log in to the user account and navigate to the post creation page
2. Enter text content for the post and start typing a hashtag (e.g., #so...)
3. Verify that the system displays a dropdown list of popular hashtags starting with the same letters (e.g., #socialmedia, #socialnetworking)
4. Select a suggested hashtag from the list
5. Verify that the selected hashtag is added to the post content

Postconditions:

- \* The system successfully suggests popular hashtags as users type
- \* The selected hashtag is added to the post content

Expected Result: The system suggests popular hashtags as users type, helping them discover relevant and trending topics.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

## **Data - 9**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Emojis and Reactions

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

Here are five detailed test cases for the Emojis and Reactions feature, ensuring comprehensive coverage of the

acceptance criteria:

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

\* Test Case ID: EMR-001

\* Test Case Description: This test case ensures that users can react to posts using different emojis, covering a range of emotions.

\* Test Suite: Emojis and Reactions

\* Test Priority: High

\* Preconditions:

+ The user is logged in to the social media platform.

+ A post is available on the user's news feed.

\* Test Data: No test data needed

\* Test Steps:

1. Log in to the social media platform.
2. Navigate to the news feed.
3. Find a post on the news feed.
4. Click on the "React" button below the post.
5. Select an emoji from the available options (e.g., 😊, 🤔, 🙄).
6. Verify that the selected emoji is displayed below the post with a count of 1.

\* Postconditions:

+ The reaction is saved and visible to other users.

\* Expected Result: The user can react to a post using a variety of emojis, and the reaction is displayed correctly.

\* Severity: Major

\* Priority: High

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

\*\*Test Case 2: Verify that emojis can be used in comments and messages.\*\*

\* Test Case ID: EMR-002

\* Test Case Description: This test case ensures that users can use emojis in comments and messages to express their emotions.

\* Test Suite: Emojis and Reactions

\* Test Priority: High

\* Preconditions:

+ The user is logged in to the social media platform.

+ A post is available on the user's news feed.

\* Test Data: No test data needed

\* Test Steps:

1. Log in to the social media platform.
2. Navigate to the news feed.
3. Find a post on the news feed.
4. Click on the "Comment" button below the post.
5. Type a comment including an emoji (e.g., "Great post! 🍌").
6. Click the "Post" button.
7. Verify that the comment is displayed with the emoji.
8. Repeat steps 4-7 for a private message.

\* Postconditions:

+ The comment and message are saved and visible to the intended recipients.

\* Expected Result: The user can use emojis in comments and messages, and they are displayed correctly.

\* Severity: Major

\* Priority: High



\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can see a count of reactions on posts.\*\***

\* Test Case ID: EMR-003

\* Test Case Description: This test case ensures that users can view the total count of reactions on a post.

\* Test Suite: Emojis and Reactions

\* Test Priority: Medium

\* Preconditions:

+ The user is logged in to the social media platform.

+ A post is available on the user's news feed with at least one reaction.

\* Test Data: No test data needed

\* Test Steps:

1. Log in to the social media platform.

2. Navigate to the news feed.

3. Find a post on the news feed with at least one reaction.

4. Verify that the reaction count is displayed below the post (e.g., "10 reactions").

\* Postconditions:

+ None

\* Expected Result: The user can view the total count of reactions on a post.

\* Severity: Minor

\* Priority: Medium

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

**\*\*Test Case 4: Verify that the platform supports a wide range of emojis.\*\***

\* Test Case ID: EMR-004

\* Test Case Description: This test case ensures that the social media platform supports a diverse set of emojis for user reactions.

\* Test Suite: Emojis and Reactions

\* Test Priority: Medium

\* Preconditions:

+ The user is logged in to the social media platform.

+ A post is available on the user's news feed.

\* Test Data: No test data needed

\* Test Steps:

1. Log in to the social media platform.

2. Navigate to the news feed.

3. Find a post on the news feed.

4. Click on the "React" button below the post.

5. Verify that the emoji selection includes a diverse range of emojis (e.g., 🍌, 🍌, 🍌, 🍌).

\* Postconditions:

+ None

\* Expected Result: The platform supports a wide range of emojis for user reactions.

\* Severity: Minor

\* Priority: Medium

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

\*\*Test Case 5: Verify that users can remove or change their reactions.\*\*

\* Test Case ID: EMR-005

\* Test Case Description: This test case ensures that users can remove or change their reactions to posts and messages.

\* Test Suite: Emojis and Reactions

\* Test Priority: High

\* Preconditions:

+ The user is logged in to the social media platform.

+ A post is available on the user's news feed with a reaction from the user.

\* Test Data: No test data needed

\* Test Steps:

1. Log in to the social media platform.

2. Navigate to the news feed.

3. Find a post on the news feed with a reaction from the user.

4. Click on the "React" button below the post.

5. Select a different emoji or remove the existing reaction.

6. Verify that the reaction is updated or removed correctly.

\* Postconditions:

+ The reaction is updated or removed from the post.

\* Expected Result: The user can remove or change their reactions to posts and messages.

\* Severity: Major

\* Priority: High

\* Type of Testing: Functional Testing

\* Test Case Approach: Positive

-----

**Data - 10**

**Project name -**

## Social Media Platform

### **Project description -**

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

### **Feature name -**

Live Streaming

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

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

**\*\*Test Case 1: Verify that users can start a live stream\*\***

Test Case ID: LS-001

Test Case Title: Verify that users can start a live stream

Test Case Description: This test case ensures that users can successfully start a live stream on the social media

platform.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- \* User is logged in to their account
- \* User has granted necessary permissions for live streaming
- \* Internet connectivity is stable

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Click on the "Go Live" button on the user's profile page
3. Select the stream type (public or followers-only) and enter a title and description for the stream
4. Click "Start Streaming" to begin the live stream
5. Verify that the stream is live and visible to the selected audience

Postconditions:

- \* The live stream is stopped after 10 minutes to conserve resources
- \* The stream is saved for later viewing (if selected)

Expected Result: The user can successfully start a live stream, and the stream is visible to the selected audience.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that live streams can be public or limited to followers\*\***

Test Case ID: LS-002

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

Test Case Description: This test case ensures that users can set the visibility of their live stream to either public or followers-only.

Test Suite: Live Streaming

Test Priority: Medium

Preconditions:

- \* User is logged in to their account
- \* User has granted necessary permissions for live streaming

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Click on the "Go Live" button on the user's profile page
3. Select the stream type (public or followers-only) and enter a title and description for the stream
4. Click "Start Streaming" to begin the live stream
5. Verify that the stream is visible to the selected audience (public or followers-only)

Postconditions:

- \* The live stream is stopped after 10 minutes to conserve resources

- \* The stream is saved for later viewing (if selected)

Expected Result: The user can set the visibility of their live stream to either public or followers-only, and the stream is only visible to the selected audience.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that viewers can interact with the streamer through comments and reactions\*\***

Test Case ID: LS-003

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

Test Case Description: This test case ensures that viewers can interact with the streamer through comments and reactions during a live stream.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- \* User is logged in to their account

- \* User has granted necessary permissions for live streaming

- \* A live stream is in progress

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Join a live stream as a viewer
3. Enter a comment and send it to the streamer
4. Click on a reaction button (e.g., "Like", "Love", etc.)
5. Verify that the comment and reaction are displayed on the stream

Postconditions:

- \* The live stream is stopped after 10 minutes to conserve resources
- \* The stream is saved for later viewing (if selected)

Expected Result: Viewers can interact with the streamer through comments and reactions during a live stream, and the interactions are displayed on the stream.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the stream displays viewer count and live comments\*\***

Test Case ID: LS-004

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

Test Case Description: This test case ensures that the live stream displays the viewer count and live comments in real-time.

Test Suite: Live Streaming

Test Priority: High

Preconditions:



- \* User is logged in to their account
- \* User has granted necessary permissions for live streaming
- \* A live stream is in progress

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Join a live stream as a viewer
3. Verify that the viewer count is displayed on the stream
4. Enter a comment and send it to the streamer
5. Verify that the comment is displayed on the stream in real-time

Postconditions:

- \* The live stream is stopped after 10 minutes to conserve resources
- \* The stream is saved for later viewing (if selected)

Expected Result: The live stream displays the viewer count and live comments in real-time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can save the live stream for later viewing\*\***

Test Case ID: LS-005

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 live stream for later viewing.

Test Suite: Live Streaming

Test Priority: Medium

Preconditions:

- \* User is logged in to their account
- \* User has granted necessary permissions for live streaming
- \* A live stream is in progress

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Start a live stream
3. Click on the "Save Stream" button during the live stream
4. Verify that the stream is saved for later viewing
5. Stop the live stream and verify that the saved stream is accessible in the user's profile

Postconditions:

- \* The saved stream is available for viewing in the user's profile

Expected Result: The user can save their live stream for later viewing, and the saved stream is accessible in their profile.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **Data - 11**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Stories

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

Here are five detailed test cases for the "Stories" feature:

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

Test Case ID: ST-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 with different media types and add text overlays.

Test Suite: Stories

Test Priority: High

Preconditions:

- \* The user is logged in to their account.
- \* The user has access to the story creation feature.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a user.
2. Navigate to the story creation page.
3. Select a photo from the device's gallery.
4. Add a text overlay to the photo.
5. Save the story.
6. Repeat steps 3-5 for video media type.
7. Verify that the story is created successfully.

Postconditions:

- \* The story is displayed in the user's profile.
- \* The story is visible to the user's followers.

Expected Result: The user can create stories with photos, videos, and text overlays.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: ST-02

Test Case Title: Verify that stories disappear after 24 hours

Test Case Description: This test case ensures that stories are removed from the platform after 24 hours.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- \* The user has created a story.
- \* The story has been posted for 24 hours.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a user.
2. Navigate to the user's profile.
3. Verify that the story is no longer visible.
4. Check the story's timestamp to ensure it has exceeded 24 hours.

Postconditions:

- \* The story is no longer displayed in the user's profile.
- \* The story is no longer accessible.

Expected Result: Stories disappear after 24 hours.

Severity: Major

Priority: Medium

Type of Testing: System Testing

Test Case Approach: Positive

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

Test Case ID: ST-03

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

Test Case Description: This test case ensures that users can view the list of users who have seen their stories.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- \* The user has created a story.
- \* The story has been viewed by at least one other user.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a user.
2. Navigate to the story view page.
3. Click on the "Viewers" button.
4. Verify that the list of viewers is displayed correctly.

Postconditions:

- \* The list of viewers is updated in real-time.

Expected Result: Users can see who viewed their stories.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: ST-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.

Test Suite: Stories

Test Priority: High

Preconditions:

- \* The user has created a story.
- \* The user has a list of followers.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a user.
2. Navigate to the story creation page.
3. Select the privacy setting for the story (public or selected followers).
4. Save the story.
5. Verify that the story is visible to the intended audience.

Postconditions:

- \* The story is only visible to the intended audience.

Expected Result: Stories can be shared publicly or with selected followers.

Severity: Critical

Priority: High

Type of Testing: Functional Testing



Test Case Approach: Positive

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

Test Case ID: ST-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.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- \* The user has created a story.
- \* The story creation page has sticker and filter options.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a user.
2. Navigate to the story creation page.
3. Add a sticker to the story.
4. Add a filter to the story.
5. Save the story.
6. Verify that the sticker and filter are applied correctly.

Postconditions:

\* The sticker and filter are visible on the story.

Expected Result: Users can add stickers and filters to their stories.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

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:**

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

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

Test Case ID: ATI-001

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

Test Case Description: This test case verifies that the analytics and insights feature displays follower demographics data to the user.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

? The user is logged in to the social media platform.

? The user has at least 10 followers.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as the user.
2. Navigate to the analytics and insights page.
3. Click on the "Follower Demographics" tab.
4. Verify that the page displays data on follower demographics, including age, gender, and location.

Postconditions:

? None

Expected Result: The analytics and insights page displays follower demographics data to the user.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: ATI-002

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

Test Case Description: This test case verifies that the analytics and insights feature displays post reach and engagement metrics to the user.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

? The user is logged in to the social media platform.

? The user has at least one post with engagement (likes, comments, shares).

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as the user.
2. Navigate to the analytics and insights page.
3. Click on the "Post Performance" tab.
4. Verify that the page displays post reach and engagement metrics, including likes, comments, shares, and reach.

Postconditions:

? None

Expected Result: The analytics and insights page displays post reach and engagement metrics to the user.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: ATI-003

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

Test Case Description: This test case verifies that the analytics and insights feature allows users to track audience growth trends.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

? The user is logged in to the social media platform.

? The user has at least 10 followers with varying growth trends.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as the user.
2. Navigate to the analytics and insights page.
3. Click on the "Audience Growth" tab.
4. Verify that the page displays audience growth trends, including follower gain/loss over time.

Postconditions:

? None

Expected Result: The analytics and insights page displays audience growth trends to the user.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: ATI-004

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

Test Case Description: This test case verifies that the analytics and insights feature displays data in a user-friendly format.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

- ? The user is logged in to the social media platform.
- ? The user has access to analytics and insights data.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as the user.
2. Navigate to the analytics and insights page.
3. Verify that the page displays data in a clean, organized, and easy-to-understand format.
4. Verify that the page includes visualizations such as charts, graphs, and tables.

Postconditions:

- ? None

Expected Result: The analytics and insights page displays data in a user-friendly format.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

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

Test Case ID: ATI-005

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

Test Case Description: This test case verifies that the analytics and insights feature allows users to filter data by time range and content type.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

? The user is logged in to the social media platform.

? The user has access to analytics and insights data.

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as the user.
2. Navigate to the analytics and insights page.
3. Verify that the page allows filtering by time range (e.g., last 7 days, last 30 days).
4. Verify that the page allows filtering by content type (e.g., posts, photos, videos).
5. Apply a filter (e.g., last 7 days, posts) and verify that the data is updated accordingly.

Postconditions:

? None

Expected Result: The analytics and insights page allows filtering by time range and content type.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

## Data - 13

### Project name -

Social Media Platform

### Project description -

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

### Feature name -

Events and Groups

### Feature description -

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

### Userstory title -

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

### Acceptance criteria -

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

### Generated Test Cases:

Here are five detailed test cases for the Events and Groups feature of the Social Media Platform:



**\*\*Test Case 1: Verify that users can create events\*\***

Test Case ID: EGRP-001

Test Case Title: Verify that users can create events

Test Case Description: This test case ensures that users can create events with required details such as date, time, location, and description.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- \* User is logged in to the platform
- \* User has a valid profile
- \* User has not reached the maximum number of events created

Test Data:

- \* Event title
- \* Event date
- \* Event time
- \* Event location
- \* Event description

Test Steps:

1. Log in to the platform as a user.
2. Click on the "Create Event" button.

3. Fill in the event details (title, date, time, location, description).
4. Click on the "Create" button.
5. Verify that the event is created successfully.

Postconditions:

- \* The event is displayed on the user's profile.
- \* The event is listed in the user's events section.

Expected Result: The user can create an event with required details.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can join interest-based groups\*\***

Test Case ID: EGRP-002

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

Test Case Description: This test case ensures that users can join interest-based groups and participate in discussions.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- \* User is logged in to the platform

- \* User has a valid profile
- \* Group exists with available slots

Test Data:

- \* Group name
- \* Group description

Test Steps:

1. Log in to the platform as a user.
2. Search for an interest-based group.
3. Click on the "Join Group" button.
4. Verify that the user is added to the group members list.
5. Verify that the user can participate in group discussions.

Postconditions:

- \* The user is added to the group members list.
- \* The user can see group discussions.

Expected Result: The user can join an interest-based group and participate in discussions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that group discussions can be organized by topic\*\***

Test Case ID: EGRP-003

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

Test Case Description: This test case ensures that group discussions can be organized by topic, making it easier for users to find relevant conversations.

Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

- \* User is logged in to the platform
- \* User is a member of a group
- \* Group has multiple topics

Test Data:

- \* Group topic names

Test Steps:

1. Log in to the platform as a user.
2. Navigate to a group discussion page.
3. Verify that topics are listed in the discussion page.
4. Click on a topic to view related discussions.
5. Verify that discussions are organized by topic.

Postconditions:

- \* Topics are listed in the discussion page.

- \* Discussions are organized by topic.

Expected Result: Group discussions can be organized by topic.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that event details include date, time, location, and description\*\***

Test Case ID: EGRP-004

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

Test Case Description: This test case ensures that event details include required information such as date, time, location, and description.

Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

- \* User is logged in to the platform

- \* User has created an event

Test Data:

- \* Event title
- \* Event date
- \* Event time
- \* Event location
- \* Event description

Test Steps:

1. Log in to the platform as a user.
2. Navigate to an event page.
3. Verify that event details (date, time, location, description) are displayed.
4. Verify that event details are accurate.

Postconditions:

- \* Event details are displayed on the event page.
- \* Event details are accurate.

Expected Result: Event details include required information.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can RSVP to events and see the list of attendees\*\***

Test Case ID: EGRP-005

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

Test Case Description: This test case ensures that users can RSVP to events and view the list of attendees.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- \* User is logged in to the platform
- \* User has access to an event page
- \* Event is not full

Test Data:

- \* Event title
- \* User profile information

Test Steps:

1. Log in to the platform as a user.
2. Navigate to an event page.
3. Click on the "RSVP" button.
4. Verify that the user is added to the attendee list.
5. Verify that the user can view the list of attendees.

Postconditions:

- \* The user is added to the attendee list.

\* The user can view the list of attendees.

Expected Result: The user can RSVP to an event and view the list of attendees.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

-----

## **Data - 14**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Monetization Options

### **Feature description -**

Monetization options allow users to earn money through various means on the platform. This may include advertising revenue, sponsored content partnerships, affiliate marketing programs, subscriptions, virtual gifts, or donations from followers, providing creators and influencers with opportunities to monetize their content and audience.

### **Userstory title -**



As a content creator, I want to access monetization options so that I can earn revenue from my content.

**Acceptance criteria -**

['Users can apply for monetization options.', 'Monetization includes advertising revenue and sponsored content.', 'Users can receive payments through the platform.', 'Detailed earnings reports are available.', 'Monetization eligibility requirements are clearly stated.']

**Generated Test Cases:**

Here are five test cases for the Monetization Options feature, ensuring that each acceptance criterion is covered by at least one test case:

**\*\*Test Case 1: Verify that users can apply for monetization options\*\***

Test Case ID: MO-001

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

Test Case Description: This test case verifies that users can successfully apply for monetization options on the platform.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- \* The user has a verified account on the platform
- \* The user meets the eligibility requirements for monetization

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a verified user

2. Navigate to the monetization options page
3. Click on the "Apply for Monetization" button
4. Fill in the required application form
5. Submit the application

Postconditions:

- \* The user's application is pending review
- \* The user receives a confirmation email

Expected Result: The user can successfully apply for monetization options.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that monetization includes advertising revenue and sponsored content\*\***

Test Case ID: MO-002

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

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

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- \* The user has a verified account on the platform
- \* The user has been approved for monetization

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an approved monetization user
2. Navigate to the monetization options page
3. Verify that advertising revenue and sponsored content are listed as available monetization options
4. Click on each option to view details
5. Verify that the details are accurate and up-to-date

Postconditions:

\* The user can view the available monetization options

Expected Result: The platform offers advertising revenue and sponsored content as monetization options.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can receive payments through the platform\*\***

Test Case ID: MO-003

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

Test Case Description: This test case verifies that users can receive payments through the platform for their monetized content.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- \* The user has a verified account on the platform
- \* The user has been approved for monetization
- \* The user has earned revenue through monetized content

Test Data: Test user account with monetized content

Test Steps:

1. Log in to the platform as an approved monetization user
2. Navigate to the earnings report page
3. Verify that the user's earnings are displayed accurately
4. Click on the "Withdraw Earnings" button
5. Verify that the payment is processed successfully

Postconditions:

- \* The user receives the payment in their preferred payment method

Expected Result: Users can receive payments through the platform for their monetized content.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that detailed earnings reports are available\*\***

Test Case ID: MO-004

Test Case Title: Verify that detailed earnings reports are available

Test Case Description: This test case verifies that the platform provides detailed earnings reports to users with monetized content.

Test Suite: Monetization Options

Test Priority: Medium

Preconditions:

- \* The user has a verified account on the platform
- \* The user has been approved for monetization
- \* The user has earned revenue through monetized content

Test Data: Test user account with monetized content

Test Steps:

1. Log in to the platform as an approved monetization user
2. Navigate to the earnings report page
3. Verify that the report displays detailed information on earnings, including date, revenue source, and amount
4. Verify that the report can be filtered by date range and revenue source
5. Verify that the report can be exported in a CSV or PDF format

Postconditions:

- \* The user can view and analyze their earnings report

Expected Result: The platform provides detailed earnings reports to users with monetized content.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that monetization eligibility requirements are clearly stated\*\***

Test Case ID: MO-005

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

Test Case Description: This test case verifies that the platform clearly states the eligibility requirements for monetization.

Test Suite: Monetization Options

Test Priority: Medium

Preconditions:

\* The user has a verified account on the platform

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a verified user
2. Navigate to the monetization options page
3. Verify that the eligibility requirements are displayed prominently
4. Verify that the requirements are clear, concise, and easy to understand
5. Verify that the requirements include information on content quality, engagement, and other relevant metrics

Postconditions:

\* The user can view and understand the eligibility requirements

Expected Result: The platform clearly states the eligibility requirements for monetization.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

---

## **Data - 15**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Explore or Discover Tab

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

Here are five detailed test cases for the Explore or Discover Tab feature:

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

Test Case ID: EXP-001

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

Test Case Description: This test case verifies that the explore tab shows trending and popular content to users, allowing them to discover new posts, accounts, and topics.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- \* User is logged in to the social media platform
- \* User has not customized their explore tab preferences
- \* There is trending and popular content available on the platform

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Navigate to the Explore tab
3. Observe the content displayed on the Explore tab
4. Verify that the content is trending and popular, including posts, accounts, and topics

Postconditions:

- \* The Explore tab is updated with new content regularly
- \* Users can interact with the content displayed (e.g., like, comment, share)



Expected Result: The Explore tab displays trending and popular content, allowing users to discover new posts, accounts, and topics.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: EXP-002

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

Test Case Description: This test case verifies that the explore tab recommends content to users based on their interests, engagement history, and social connections.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- \* User is logged in to the social media platform
- \* User has engaged with content on the platform (e.g., liked, commented, shared)
- \* User has connected with other users on the platform

Test Data: User engagement history and social connections data

Test Steps:

1. Log in to the social media platform as a registered user
2. Engage with content on the platform (e.g., like, comment, share)
3. Navigate to the Explore tab

4. Observe the content recommended on the Explore tab
5. Verify that the recommended content aligns with the user's interests and engagement history

Postconditions:

- \* The Explore tab continues to update with personalized content recommendations
- \* Users can refine their content recommendations by adjusting their interests and engagement history

Expected Result: The Explore tab recommends content to users based on their interests, engagement history, and social connections.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: EXP-003

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

Test Case Description: This test case verifies that users can filter the content displayed on the Explore tab by category or topic, allowing them to discover specific types of content.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- \* User is logged in to the social media platform
- \* The Explore tab has multiple categories or topics available

Test Data: Category or topic filters (e.g., technology, entertainment, sports)

Test Steps:

1. Log in to the social media platform as a registered user
2. Navigate to the Explore tab
3. Apply a category or topic filter (e.g., select "Technology" from the filter menu)
4. Observe the content displayed on the Explore tab
5. Verify that the content is filtered by the selected category or topic

Postconditions:

- \* The Explore tab retains the user's filter preferences
- \* Users can switch between different filters to discover new content

Expected Result: Users can filter the content displayed on the Explore tab by category or topic.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the Explore tab updates regularly with new content\*\***

Test Case ID: EXP-004

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

Test Case Description: This test case verifies that the Explore tab updates regularly with new trending and popular content, ensuring that users always have access to fresh discoveries.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- \* User is logged in to the social media platform
- \* The Explore tab has existing content displayed

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Navigate to the Explore tab
3. Observe the content displayed on the Explore tab
4. Wait for a set period (e.g., 30 minutes)
5. Refresh the Explore tab
6. Verify that new content is displayed on the Explore tab

Postconditions:

- \* The Explore tab continues to update regularly with new content
- \* Users can interact with the new content displayed (e.g., like, comment, share)

Expected Result: The Explore tab updates regularly with new trending and popular content.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can save or follow content from the Explore tab\*\***

Test Case ID: EXP-005

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

Test Case Description: This test case verifies that users can save or follow content discovered on the Explore tab, allowing them to easily access the content later.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- \* User is logged in to the social media platform
- \* The Explore tab has content displayed

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform as a registered user
2. Navigate to the Explore tab
3. Select a piece of content (e.g., post, account, topic)
4. Verify that the "Save" or "Follow" button is available
5. Click the "Save" or "Follow" button
6. Verify that the content is saved or followed successfully

Postconditions:

- \* The saved or followed content is accessible from the user's profile or dedicated section

\* Users can unfollow or unsave content as needed

Expected Result: Users can save or follow content discovered on the Explore tab.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **Data - 16**

### **Project name -**

Social Media Platform

### **Project description -**

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

### **Feature name -**

Accessibility Features

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

## Generated Test Cases:

Here are five detailed test cases for the Accessibility Features of the Social Media Platform:

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

Test Case ID: AC-001

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

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

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- \* The user has a valid account and is logged in
- \* The user has uploaded an image to their profile
- \* The user has enabled screen reader compatibility

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user with a valid account
2. Upload an image to the user's profile
3. Enable screen reader compatibility

4. Use a screen reader to access the uploaded image
5. Verify that the alt text is read aloud by the screen reader

Postconditions: None

Expected Result: The alt text is read aloud by the screen reader, ensuring accessibility for users with visual impairments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Accessibility Testing

Test Case Approach: Positive

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

Test Case ID: AC-002

Test Case Title: Verify that screen reader compatibility is available

Test Case Description: This test case ensures that the platform is compatible with screen readers, allowing users with visual impairments to access the platform's content.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- \* The user has a valid account and is logged in
- \* The user has enabled screen reader compatibility
- \* The user is using a screen reader software

Test Data: No test data needed

Test Steps:



1. Log in to the platform as a user with a valid account
2. Enable screen reader compatibility
3. Use a screen reader to navigate the platform's interface
4. Verify that the screen reader can access and read aloud all platform content
5. Verify that the screen reader can interact with platform elements, such as buttons and links

Postconditions: None

Expected Result: The platform is compatible with screen readers, ensuring accessibility for users with visual impairments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Accessibility Testing

Test Case Approach: Positive

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

Test Case ID: AC-003

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

Test Case Description: This test case ensures that the platform provides keyboard shortcuts for navigation, allowing users with mobility impairments to access the platform's content.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

\* The user has a valid account and is logged in

\* The user has enabled keyboard shortcuts

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user with a valid account
2. Enable keyboard shortcuts
3. Use keyboard shortcuts to navigate the platform's interface
4. Verify that the keyboard shortcuts can access and interact with platform elements, such as buttons and links
5. Verify that the keyboard shortcuts are documented and easily accessible

Postconditions: None

Expected Result: The platform provides keyboard shortcuts for navigation, ensuring accessibility for users with mobility impairments.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Accessibility Testing

Test Case Approach: Positive

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

Test Case ID: AC-004

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

Test Case Description: This test case ensures that the platform provides options for adjusting color contrast and font size, allowing users with visual impairments to customize their experience.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

\* The user has a valid account and is logged in

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user with a valid account
2. Access the platform's accessibility settings
3. Verify that options for adjusting color contrast are available
4. Verify that options for adjusting font size are available
5. Verify that the changes to color contrast and font size are applied across the platform's interface

Postconditions: None

Expected Result: The platform provides options for adjusting color contrast and font size, ensuring accessibility for users with visual impairments.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Accessibility Testing

Test Case Approach: Positive

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

Test Case ID: AC-005

Test Case Title: Verify that accessibility features are clearly documented

Test Case Description: This test case ensures that the platform's accessibility features are clearly documented, allowing users to easily understand and utilize them.

Test Suite: Accessibility Features

Test Priority: Low

Preconditions:

- \* The user has a valid account and is logged in
- \* The user has accessed the platform's help or support section

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user with a valid account
2. Access the platform's help or support section
3. Verify that the accessibility features are clearly documented
4. Verify that the documentation provides step-by-step instructions for using the accessibility features
5. Verify that the documentation is accessible and usable for users with disabilities

Postconditions: None

Expected Result: The platform's accessibility features are clearly documented, ensuring that users can easily understand and utilize them.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

-----

**Data - 17**

**Project name -**

## Social Media Platform

### **Project description -**

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

### **Feature name -**

Safety and Moderation Tools

### **Feature description -**

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

### **Userstory title -**

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

### **Acceptance criteria -**

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

### **Generated Test Cases:**

Here are five detailed test cases for the Safety and Moderation Tools feature:

**\*\*Test Case 1: Verify that content moderation algorithms filter inappropriate content\*\***

Test Case ID: SMPT-001

Test Case Title: Verify that content moderation algorithms filter inappropriate content

Test Case Description: This test case verifies that the content moderation algorithms correctly identify and filter out inappropriate content on the platform.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- \* A user account with admin privileges is created
- \* Inappropriate content (e.g., offensive images, hate speech) is uploaded to the platform

Test Data: Inappropriate content samples

Test Steps:

1. Log in as an admin user and access the content moderation dashboard
2. Upload inappropriate content to the platform
3. Verify that the content moderation algorithm flags the content as inappropriate
4. Check that the content is automatically removed from the platform

Postconditions:

- \* The inappropriate content is removed from the platform
- \* A notification is sent to the user who uploaded the content, explaining why it was removed

Expected Result: The content moderation algorithm correctly identifies and filters out inappropriate content on the platform.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can report harmful or inappropriate content\*\***

Test Case ID: SMPT-002

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

Test Case Description: This test case verifies that users can successfully report harmful or inappropriate content on the platform.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- \* A user account is created
- \* Harmful or inappropriate content is uploaded to the platform

Test Data: Harmful or inappropriate content samples

Test Steps:

1. Log in as a user and access the platform
2. Find the harmful or inappropriate content and click the "Report" button
3. Fill out the reporting form with a valid reason for reporting the content
4. Submit the report
5. Verify that the report is successfully submitted and sent to the moderation team

Postconditions:

- \* The report is sent to the moderation team for review

\* The user receives a confirmation notification that the report was received

Expected Result: Users can successfully report harmful or inappropriate content on the platform.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that blocking capabilities are available to users\*\***

Test Case ID: SMPT-003

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

Test Case Description: This test case verifies that users can block other users on the platform to prevent unwanted interactions.

Test Suite: Safety and Moderation Tools

Test Priority: Medium

Preconditions:

\* Two user accounts are created

\* One user sends a friend request to the other user

Test Data: None

Test Steps:

1. Log in as a user and access the platform
2. Find the user who sent the friend request and click the "Block" button
3. Verify that the user is successfully blocked and can no longer interact with the blocking user



4. Log out and log in as the blocked user
5. Verify that the blocked user can no longer see the blocking user's profile or interact with them

Postconditions:

- \* The blocked user is removed from the blocking user's friend list
- \* The blocked user receives a notification that they have been blocked

Expected Result: Users can successfully block other users on the platform to prevent unwanted interactions.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that comment filters prevent offensive language\*\***

Test Case ID: SMPT-004

Test Case Title: Verify that comment filters prevent offensive language

Test Case Description: This test case verifies that the comment filters correctly identify and filter out offensive language in comments on the platform.

Test Suite: Safety and Moderation Tools

Test Priority: Medium

Preconditions:

- \* A user account is created
- \* A post with a comment section is created

Test Data: Offensive language samples

Test Steps:

1. Log in as a user and access the post with a comment section
2. Enter an offensive comment and attempt to post it
3. Verify that the comment filter blocks the comment and displays an error message
4. Enter a non-offensive comment and post it
5. Verify that the non-offensive comment is successfully posted

Postconditions:

- \* The offensive comment is not posted
- \* The user receives an error message explaining why the comment was not posted

Expected Result: The comment filters correctly identify and filter out offensive language in comments on the platform.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that community guidelines are enforced consistently\*\***

Test Case ID: SMPT-005

Test Case Title: Verify that community guidelines are enforced consistently

Test Case Description: This test case verifies that the community guidelines are consistently enforced across the platform to ensure a safe and positive experience for users.

Test Suite: Safety and Moderation Tools

Test Priority: Low

Preconditions:

- \* A user account is created
- \* A post that violates community guidelines is created

Test Data: None

Test Steps:

1. Log in as a user and access the post that violates community guidelines
2. Verify that the post is automatically flagged by the moderation algorithm
3. Verify that the post is removed from the platform
4. Verify that the user who posted the content receives a notification explaining why the post was removed
5. Verify that the community guidelines are clearly displayed on the platform

Postconditions:

- \* The post is removed from the platform
- \* The user receives a notification explaining why the post was removed
- \* The community guidelines are clearly displayed on the platform

Expected Result: The community guidelines are consistently enforced across the platform to ensure a safe and positive experience for users.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

**Data - 18**

**Project name -**

Social Media Platform

**Project description -**

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

**Feature name -**

Multi-Language Support

**Feature description -**

Multi-language support allows the platform to cater to users from diverse linguistic backgrounds by offering interface localization and content translation features. Users can select their preferred language for the interface, and posts or messages can be translated into different languages, enhancing accessibility and user experience across global audiences.

**Userstory title -**

As a user, I want multi-language support so that I can use the platform in my preferred language.

**Acceptance criteria -**

['Users can select their preferred interface language.', 'The platform offers content translation features.', 'Posts and messages can be translated into different languages.', 'Language preferences are saved and applied consistently.', 'The platform supports a wide range of languages.']

**Generated Test Cases:**

Here are five detailed test cases for the Multi-Language Support feature:

**\*\*Test Case 1: Verify that the platform allows users to select their preferred interface language.\*\***

Test Case ID: ML-001

Test Case Title: Verify that the platform allows users to select their preferred interface language.

Test Case Description: This test case validates that the platform provides an option for users to select their preferred interface language, and the selected language is applied consistently throughout the platform.

Test Suite: User Profile

Test Priority: High

Preconditions:

- \* The user has a valid account and is logged in to the platform.

- \* The platform has multiple language options available.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a valid user.
2. Navigate to the user profile settings page.
3. Click on the language dropdown menu.
4. Select a language other than the default language (e.g., Spanish).
5. Click "Save" to update the language preference.

Postconditions:

- \* The platform's interface language is updated to the selected language.

\* All further interactions with the platform are in the selected language.

Expected Result: The platform successfully updates the interface language to the user's preferred language, and all subsequent interactions are in the selected language.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the platform offers content translation features for posts and messages.\*\***

Test Case ID: ML-002

Test Case Title: Verify that the platform offers content translation features for posts and messages.

Test Case Description: This test case verifies that the platform provides a mechanism for translating posts and messages into different languages, ensuring that users can understand content in their preferred language.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

\* A post or message is created in a language different from the user's preferred language.

\* The user has selected a language for the interface that is different from the post/message language.

Test Data: Sample post/message in English, Spanish, and French

Test Steps:

1. Create a post/message in a language different from the user's preferred language (e.g., English).

2. Log in to the platform as a user with a different language preference (e.g., Spanish).
3. Navigate to the post/message.
4. Click on the "Translate" button or option.
5. Verify that the post/message is translated accurately into the user's preferred language.

Postconditions:

\* The translated content is displayed correctly in the user's preferred language.

Expected Result: The platform successfully translates the post/message into the user's preferred language, allowing users to understand content in their preferred language.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that language preferences are saved and applied consistently across the platform.\*\***

Test Case ID: ML-003

Test Case Title: Verify that language preferences are saved and applied consistently across the platform.

Test Case Description: This test case ensures that the user's selected language preference is saved and applied consistently across the platform, including different sections and features.

Test Suite: User Profile

Test Priority: Medium

Preconditions:

- \* The user has selected a language preference.
- \* The user has logged out and logged back in to the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a valid user.
2. Navigate to the user profile settings page.
3. Verify that the selected language preference is saved and displayed correctly.
4. Log out of the platform.
5. Log back in to the platform.
6. Verify that the language preference is still applied consistently across the platform (e.g., news feed, messaging, profile).

Postconditions:

- \* The user's language preference is retained and applied consistently across the platform.

Expected Result: The user's language preference is saved and applied consistently across the platform, ensuring a seamless user experience.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the platform supports a wide range of languages.\*\***



Test Case ID: ML-004

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

Test Case Description: This test case validates that the platform supports multiple languages, including languages with unique characters and scripts, to cater to a diverse user base.

Test Suite: User Profile

Test Priority: Medium

Preconditions:

\* The platform has multiple language options available, including languages with unique characters and scripts (e.g., Arabic, Chinese, Japanese).

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a valid user.
2. Navigate to the user profile settings page.
3. Verify that the language dropdown menu includes multiple languages with unique characters and scripts.
4. Select a language with unique characters and scripts (e.g., Arabic).
5. Verify that the platform's interface is updated correctly to display the selected language.

Postconditions:

\* The platform supports a wide range of languages, including languages with unique characters and scripts.

Expected Result: The platform successfully supports a wide range of languages, ensuring that users from diverse linguistic backgrounds can use the platform comfortably.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the platform handles invalid language input correctly.\*\***

Test Case ID: ML-005

Test Case Title: Verify that the platform handles invalid language input correctly.

Test Case Description: This test case ensures that the platform handles invalid language input, such as an unknown language code, and provides a suitable error message or fallback language.

Test Suite: User Profile

Test Priority: Low

Preconditions:

\* The user has attempted to select an invalid language code or input.

Test Data: Invalid language code (e.g., "XYZ")

Test Steps:

1. Log in to the platform as a valid user.
2. Navigate to the user profile settings page.
3. Enter an invalid language code or input (e.g., "XYZ").
4. Click "Save" to update the language preference.
5. Verify that the platform displays an error message or falls back to a default language.

Postconditions:

\* The platform handles invalid language input correctly, providing a suitable error message or fallback language.

Expected Result: The platform correctly handles invalid language input, ensuring that users are not affected by invalid language preferences.

Severity: Trivial

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Negative

-----

**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:**

Here are five detailed test cases for the Cross-Platform Integration feature:

**\*\*Test Case 1: Verify that users can link their accounts with other social media platforms\*\***

Test Case ID: CP-01

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

Test Case Description: This test case verifies that users can successfully link their accounts with other social media platforms, enabling cross-platform integration.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- \* User has an account on the social media platform
- \* User has an account on another social media platform (e.g., Facebook, Twitter)
- \* Both accounts are active and valid

Test Data: None

Test Steps:

1. Log in to the social media platform
2. Navigate to the account settings page
3. Click on the "Link Account" button
4. Select the social media platform to link (e.g., Facebook, Twitter)
5. Authenticate with the selected platform
6. Verify that the accounts are successfully linked

Postconditions:

- \* The accounts are linked, and the user can access both platforms from the social media platform
- \* The user's profile information is updated to reflect the linked account

Expected Result: The user can successfully link their accounts with other social media platforms.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that content can be cross-posted to other platforms\*\***

Test Case ID: CP-02

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

Test Case Description: This test case verifies that users can cross-post content to other social media platforms, enhancing content reach and convenience.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

- \* User has an account on the social media platform
- \* User has an account on another social media platform (e.g., Facebook, Twitter)
- \* Both accounts are active and valid
- \* User has created content (e.g., post, photo, video) on the social media platform

Test Data: Sample post, photo, or video content

Test Steps:

1. Log in to the social media platform
2. Create a new post, photo, or video content
3. Select the option to cross-post to other platforms
4. Choose the platforms to cross-post to (e.g., Facebook, Twitter)
5. Verify that the content is successfully cross-posted to the selected platforms

Postconditions:

- \* The content is successfully cross-posted to the selected platforms
- \* The content is displayed correctly on the target platforms

Expected Result: Content can be successfully cross-posted to other social media platforms.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can sign in using single sign-on (SSO)\*\***

Test Case ID: CP-03

Test Case Title: Verify that users can sign in using single sign-on (SSO)

Test Case Description: This test case verifies that users can use single sign-on (SSO) to access the social media platform, enhancing convenience and security.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- \* User has an account on the social media platform
- \* User has an account on another social media platform (e.g., Facebook, Twitter)
- \* Both accounts are active and valid

Test Data: None

Test Steps:

1. Navigate to the social media platform login page
2. Select the SSO option (e.g., "Log in with Facebook")
3. Authenticate with the selected platform
4. Verify that the user is successfully logged in to the social media platform

Postconditions:

- \* The user is successfully logged in to the social media platform using SSO

\* The user's profile information is updated to reflect the SSO login

Expected Result: Users can successfully sign in using single sign-on (SSO).

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that integration with external media sources is available\*\***

Test Case ID: CP-04

Test Case Title: Verify that integration with external media sources is available

Test Case Description: This test case verifies that the social media platform integrates with external media sources, enabling users to share content from various platforms.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

\* User has an account on the social media platform

\* External media source (e.g., YouTube, Instagram) is active and valid

Test Data: Sample media content from external source

Test Steps:

1. Log in to the social media platform
2. Navigate to the content sharing page
3. Select the external media source to share from (e.g., YouTube, Instagram)



4. Authenticate with the selected media source
5. Verify that the media content is successfully shared on the social media platform

Postconditions:

- \* The external media source is successfully integrated with the social media platform
- \* The shared media content is displayed correctly on the social media platform

Expected Result: Integration with external media sources is available and functional.

Severity: Major

Priority: Medium

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that content sharing across platforms is seamless\*\***

Test Case ID: CP-05

Test Case Title: Verify that content sharing across platforms is seamless

Test Case Description: This test case verifies that content sharing across platforms is seamless, ensuring a consistent user experience.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

- \* User has an account on the social media platform
- \* User has an account on another social media platform (e.g., Facebook, Twitter)
- \* Both accounts are active and valid

\* User has created content (e.g., post, photo, video) on the social media platform

Test Data: Sample post, photo, or video content

Test Steps:

1. Log in to the social media platform
2. Create a new post, photo, or video content
3. Select the option to share on other platforms (e.g., Facebook, Twitter)
4. Verify that the content is successfully shared on the selected platforms
5. Verify that the shared content is displayed correctly on the target platforms

Postconditions:

- \* The content is successfully shared across platforms
- \* The shared content is displayed correctly on the target platforms

Expected Result: Content sharing across platforms is seamless and consistent.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 20**

**Project name -**

Social Media Platform

## **Project description -**

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

## **Feature name -**

Gamification and Rewards

## **Feature description -**

Gamification and rewards enhance user engagement by incorporating game-like elements such as points, badges, leaderboards, and challenges. Users can earn rewards for their activities, achievements, and contributions on the platform, fostering a sense of accomplishment and encouraging continued interaction and participation.

## **Userstory title -**

As a user, I want gamification and rewards so that I can earn recognition and incentives for my activities.

## **Acceptance criteria -**

['Users can earn points and badges for their activities.', 'Leaderboards display top users based on points or achievements.', 'Challenges are available for users to complete and earn rewards.', 'Rewards can include virtual goods, discounts, or exclusive content.', 'Users can view their progress and achievements in a dedicated section.']

## **Generated Test Cases:**

Here are five detailed test cases for the Gamification and Rewards feature of the Social Media Platform:

**\*\*Test Case 1: Verify that users can earn points and badges for their activities\*\***

Test Case ID: GMR-001

Test Case Title: Verify that users can earn points and badges for their activities

Test Case Description: This test case verifies that users can earn points and badges for their activities on the platform, such as creating a post, commenting, or sharing content.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- \* User is logged in to the platform
- \* User has created at least one post
- \* User has commented on at least one post

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user
2. Create a new post and share it with friends
3. Comment on at least two posts from friends
4. Check the user's profile page for earned points and badges

Postconditions:

- \* Verify that points and badges are displayed on the user's profile page
- \* Verify that the points and badges are updated correctly

Expected Result: User earns points and badges for their activities on the platform

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that leaderboards display top users based on points or achievements\*\***

Test Case ID: GMR-002

Test Case Title: Verify that leaderboards display top users based on points or achievements

Test Case Description: This test case verifies that leaderboards display the top users based on their points or achievements on the platform.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- \* At least 10 users are active on the platform
- \* Users have earned points and badges for their activities

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an administrator
2. Navigate to the leaderboard page
3. Verify that the leaderboard displays the top users based on points or achievements

Postconditions:

- \* Verify that the leaderboard is updated in real-time
- \* Verify that the leaderboard displays the correct ranking of users

Expected Result: Leaderboard displays top users based on points or achievements

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that challenges are available for users to complete and earn rewards\*\***

Test Case ID: GMR-003

Test Case Title: Verify that challenges are available for users to complete and earn rewards

Test Case Description: This test case verifies that challenges are available for users to complete and earn rewards on the platform.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

\* At least one challenge is available on the platform

\* User is logged in to the platform

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user
2. Navigate to the challenge page
3. Verify that challenges are available for users to complete
4. Complete a challenge and verify that rewards are earned

Postconditions:

- \* Verify that challenges are updated in real-time

- \* Verify that rewards are awarded correctly

Expected Result: Challenges are available for users to complete and earn rewards

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that rewards can include virtual goods, discounts, or exclusive content\*\***

Test Case ID: GMR-004

Test Case Title: Verify that rewards can include virtual goods, discounts, or exclusive content

Test Case Description: This test case verifies that rewards can include virtual goods, discounts, or exclusive content on the platform.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- \* Rewards are set up on the platform

- \* User has earned rewards for completing challenges or achieving milestones

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user
2. Verify that rewards are displayed on the user's profile page
3. Verify that rewards include virtual goods, discounts, or exclusive content

Postconditions:

- \* Verify that rewards are redeemed correctly
- \* Verify that rewards are updated in real-time

Expected Result: Rewards can include virtual goods, discounts, or exclusive content

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can view their progress and achievements in a dedicated section\*\***

Test Case ID: GMR-005

Test Case Title: Verify that users can view their progress and achievements in a dedicated section

Test Case Description: This test case verifies that users can view their progress and achievements in a dedicated section on the platform.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- \* User is logged in to the platform
- \* User has earned points, badges, or rewards on the platform



Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user
2. Navigate to the dedicated section for progress and achievements
3. Verify that progress and achievements are displayed correctly

Postconditions:

- \* Verify that the dedicated section is updated in real-time
- \* Verify that progress and achievements are accurate

Expected Result: Users can view their progress and achievements in a dedicated section

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 21**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Creation

### **Feature description -**

Users can create tasks by providing detailed information such as the task title, description, due date, priority level, and assigning it to specific team members. This feature serves as the foundation for organizing and managing tasks within the tool.

### **Userstory title -**

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

### **Acceptance criteria -**

['Users can enter a task title, description, due date, and priority level.', 'Tasks can be assigned to specific team members.', 'The system should save and display created tasks.', 'Users can edit task details after creation.', 'The system should validate required fields (e.g., task title).']

### **Generated Test Cases:**

Here are five detailed test cases for the Task Creation feature, ensuring that each acceptance criterion is covered:

**\*\*Test Case 1: Verify that users can enter a task title, description, due date, and priority level.\*\***

Test Case ID: TC001

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 users can input required task details, including title, description, due date, and priority level, during task creation.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool.
- \* User has permission to create tasks.
- \* No existing tasks with the same title.

Test Data: Task title, description, due date, and priority level.

Test Steps:

1. Log in to the Task Management Tool as a user with task creation permission.
2. Click on the "Create Task" button.
3. Enter a valid task title, description, due date, and priority level.
4. Click the "Save" button.

Postconditions:

- \* The task is created successfully.
- \* Task details are displayed in the task list.

Expected Result: The system allows users to enter and save task title, description, due date, and priority level successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that tasks can be assigned to specific team members.\*\***

Test Case ID: TC002

Test Case Title: Verify that tasks can be assigned to specific team members.

Test Case Description: This test case verifies that users can assign tasks to specific team members during task creation.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool.
- \* User has permission to create tasks.
- \* At least one team member is available for assignment.

Test Data: Task title, description, due date, priority level, and team member's name.

Test Steps:

1. Log in to the Task Management Tool as a user with task creation permission.
2. Click on the "Create Task" button.
3. Enter a valid task title, description, due date, and priority level.
4. Select a team member from the dropdown list.
5. Click the "Save" button.

Postconditions:

- \* The task is assigned to the selected team member.
- \* Task details are displayed in the task list with the assigned team member's name.

Expected Result: The system allows users to assign tasks to specific team members successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system saves and displays created tasks.\*\***

Test Case ID: TC003

Test Case Title: Verify that the system saves and displays created tasks.

Test Case Description: This test case verifies that the system saves created tasks and displays them in the task list.

Test Suite: Task Creation

Test Priority: High

Preconditions:

\* User is logged in to the Task Management Tool.

\* User has permission to create tasks.

\* At least one task is created.

Test Data: Task title, description, due date, priority level, and team member's name (if assigned).

Test Steps:

1. Log in to the Task Management Tool as a user with task creation permission.
2. Create a new task with valid details.
3. Click the "Save" button.
4. Verify that the task is displayed in the task list.
5. Verify that task details are displayed correctly.

Postconditions:

- \* The task is saved successfully.
- \* Task details are displayed in the task list.

Expected Result: The system saves and displays created tasks correctly.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can edit task details after creation.\*\***

Test Case ID: TC004

Test Case Title: Verify that users can edit task details after creation.

Test Case Description: This test case verifies that users can edit task details, including title, description, due date, and priority level, after task creation.

Test Suite: Task Creation

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool.
- \* User has permission to edit tasks.
- \* At least one task is created.

Test Data: Task title, description, due date, priority level, and team member's name (if assigned).

### Test Steps:

1. Log in to the Task Management Tool as a user with task edit permission.
2. Select an existing task from the task list.
3. Click the "Edit" button.
4. Modify task details, such as title, description, due date, or priority level.
5. Click the "Save" button.

### Postconditions:

- \* Task details are updated successfully.
- \* Changes are reflected in the task list.

Expected Result: The system allows users to edit task details after creation successfully.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the system validates required fields.\*\***

Test Case ID: TC005

Test Case Title: Verify that the system validates required fields.

Test Case Description: This test case verifies that the system validates required fields, such as task title, during task creation.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool.
- \* User has permission to create tasks.

Test Data: No test data needed.

Test Steps:

1. Log in to the Task Management Tool as a user with task creation permission.
2. Click on the "Create Task" button.
3. Leave the task title field blank.
4. Click the "Save" button.

Postconditions:

- \* An error message is displayed indicating that the task title is required.
- \* The task is not created.

Expected Result: The system validates required fields, such as task title, during task creation.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative

-----



**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Assignment

**Feature description -**

Tasks can be assigned to specific team members, delegating responsibility and ensuring clarity about who is accountable for each task. Assignees receive notifications about their assigned tasks, helping them stay informed and focused on their responsibilities.

**Userstory title -**

As a user, I want to assign tasks to specific team members so that responsibilities are clear and tasks are managed effectively.

**Acceptance criteria -**

['Users can select team members to assign tasks.', 'Assignees receive notifications about their tasks.', 'The system should display assigned tasks in the assignee's task list.', 'Users can reassign tasks to different team members.', 'Task assignment history should be tracked.']

**Generated Test Cases:**

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

**\*\*Test Case 1: Verify that users can select team members to assign tasks\*\***

Test Case ID: TA-001

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

Test Case Description: Test that users can select team members to assign tasks from the task creation/edit page.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* User has created a task
- \* User has team members listed in the system

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Create a new task or edit an existing one
3. Click on the "Assign to" dropdown menu
4. Select a team member from the list
5. Click "Save" to save the task assignment

Postconditions:

- \* Task is updated with the assigned team member
- \* Assigned team member receives a notification about the task assignment

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

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that assignees receive notifications about their tasks\*\***

Test Case ID: TA-002

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

Test Case Description: Test that assignees receive notifications about their assigned tasks.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

\* User has assigned a task to a team member

\* Team member has notifications enabled in their profile settings

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Assign a task to a team member
3. Verify that the team member receives a notification about the task assignment
4. Check the notification content to ensure it includes task details and assignment information

Postconditions:

\* Team member has received a notification about the task assignment

Expected Result: The assignee receives a notification about the task assignment with correct task details and assignment information.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system displays assigned tasks in the assignee's task list\*\***

Test Case ID: TA-003

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

Test Case Description: Test that assigned tasks are displayed in the assignee's task list.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

\* User has assigned a task to a team member

\* Team member has logged in to the Task Management Tool

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as the assignee
2. Navigate to the task list
3. Verify that the assigned task is displayed in the task list
4. Check the task details to ensure they match the assigned task

Postconditions:

\* Assigned task is displayed in the assignee's task list

Expected Result: The assigned task is displayed in the assignee's task list with correct task details.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can reassign tasks to different team members\*\***

Test Case ID: TA-004

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

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

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

\* User has assigned a task to a team member

\* User wants to reassign the task to a different team member

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Navigate to the task edit page
3. Click on the "Assign to" dropdown menu
4. Select a different team member from the list
5. Click "Save" to save the task reassignment

Postconditions:

- \* Task is updated with the new assigned team member
- \* Previous assignee no longer has the task assigned
- \* New assignee receives a notification about the task reassignment

Expected Result: The task is successfully reassigned to the new team member.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that task assignment history is tracked\*\***

Test Case ID: TA-005

Test Case Title: Verify that task assignment history is tracked

Test Case Description: Test that task assignment history is tracked and displayed correctly.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

\* User has assigned and reassigned a task to different team members

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Navigate to the task edit page
3. Click on the "Task History" tab
4. Verify that the task assignment history is displayed correctly, including previous assignees and reassignments

Postconditions:

\* Task assignment history is displayed correctly

Expected Result: The task assignment history is displayed correctly, showing all previous assignees and reassignments.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 23**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Deadline Reminders

**Feature description -**

Users receive reminders and notifications for upcoming task deadlines, ensuring that tasks are completed on time and preventing delays. Reminders can be customized based on users' preferences and urgency levels.

**Userstory title -**

As a user, I want to receive reminders for upcoming task deadlines so that I can complete tasks on time.

**Acceptance criteria -**

['Users can set reminders for tasks based on deadlines.', 'Reminders can be customized by urgency level.', 'The system sends notifications for upcoming deadlines.', 'Users can choose the delivery method for reminders (e.g., email, in-app).', 'Reminder settings can be adjusted or disabled.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Deadline Reminders feature:



**\*\*Test Case 1: Verify that reminders can be set for tasks based on deadlines\*\***

Test Case ID: TD001

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

Test Case Description: This test case verifies that users can set reminders for tasks based on deadlines, ensuring timely task completion.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* A task with a deadline has been created
- \* The user has access to the task management tool

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool as a user
2. Create a new task with a deadline
3. Set a reminder for the task based on the deadline
4. Save the task changes
5. Verify that the reminder is scheduled for the specified deadline

Postconditions:

- \* The reminder is scheduled for the specified deadline
- \* The user receives a notification for the upcoming deadline

Expected Result: The user can set reminders for tasks based on deadlines, and the reminder is scheduled for the specified deadline.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that reminders can be customized by urgency level\*\***

Test Case ID: TD002

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

Test Case Description: This test case verifies that users can customize reminders based on urgency levels, ensuring reminders align with task priorities.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A task with a deadline has been created
- \* The user has access to the task management tool

Test Data:

- \* Task with high urgency level
- \* Task with medium urgency level
- \* Task with low urgency level

Test Steps:

1. Log in to the task management tool as a user
2. Create three tasks with different urgency levels (high, medium, low)
3. Set reminders for each task with different customization options based on urgency levels
4. Save the task changes
5. Verify that the reminders are customized according to the specified urgency levels

Postconditions:

\* The reminders are customized according to the specified urgency levels

Expected Result: Users can customize reminders based on urgency levels, and the reminders align with task priorities.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system sends notifications for upcoming deadlines\*\***

Test Case ID: TD003

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

Test Case Description: This test case verifies that the system sends notifications for upcoming deadlines, ensuring users are aware of approaching task deadlines.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* A task with a deadline has been created
- \* The user has access to the task management tool
- \* The user has enabled notifications for reminders

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool as a user
2. Create a new task with a deadline
3. Set a reminder for the task
4. Wait for the reminder to trigger (e.g., 1 hour before the deadline)
5. Verify that the user receives a notification for the upcoming deadline

Postconditions:

- \* The user receives a notification for the upcoming deadline

Expected Result: The system sends notifications for upcoming deadlines, and the user receives a notification for the upcoming deadline.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can choose the delivery method for reminders\*\***

Test Case ID: TD004

Test Case Title: Verify that users can choose the delivery method for reminders

Test Case Description: This test case verifies that users can choose the delivery method for reminders, ensuring flexibility in receiving reminders.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A task with a deadline has been created
- \* The user has access to the task management tool

Test Data:

- \* Email delivery method
- \* In-app delivery method

Test Steps:

1. Log in to the task management tool as a user
2. Create a new task with a deadline
3. Set a reminder for the task
4. Choose the email delivery method for the reminder
5. Save the task changes
6. Verify that the reminder is delivered via email
7. Repeat steps 3-6 with the in-app delivery method

Postconditions:

\* The user can choose the delivery method for reminders

Expected Result: Users can choose the delivery method for reminders, and the reminder is delivered via the chosen method.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that reminder settings can be adjusted or disabled\*\***

Test Case ID: TD005

Test Case Title: Verify that reminder settings can be adjusted or disabled

Test Case Description: This test case verifies that users can adjust or disable reminder settings, ensuring flexibility in managing reminders.

Test Suite: Task Management

Test Priority: Low

Preconditions:

\* A task with a deadline has been created

\* The user has access to the task management tool

\* Reminders have been enabled for the task

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool as a user
2. Create a new task with a deadline
3. Set a reminder for the task
4. Adjust the reminder settings (e.g., change the reminder time or frequency)
5. Verify that the reminder settings are updated
6. Disable the reminder for the task
7. Verify that the reminder is no longer scheduled

Postconditions:

\* The reminder settings can be adjusted or disabled

Expected Result: Users can adjust or disable reminder settings, and the reminder settings are updated or disabled accordingly.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 24**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Prioritization

### **Feature description -**

Users can prioritize tasks based on their urgency and importance, allowing them to focus on high-priority tasks and manage their workload effectively. Tasks can be categorized into different priority levels, such as high, medium, or low, based on their importance to the project or team.

### **Userstory title -**

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

### **Acceptance criteria -**

['Users can set priority levels for tasks (High, Medium, Low).', 'Tasks can be sorted based on priority levels.', 'Users can change the priority level of a task.', 'The system visually distinguishes tasks based on priority.', 'Tasks without a specified priority default to Medium.']

### **Generated Test Cases:**

Here are five test cases for the Task Prioritization feature:

**\*\*Test Case 1: Verify that users can set priority levels for tasks\*\***

Test Case ID: TP-001

Test Case Title: Verify that users can set priority levels for tasks

Test Case Description: This test case verifies that users can set priority levels for tasks, ensuring that users can categorize tasks based on their urgency and importance.

Test Suite: Task Management

Test Priority: High

Preconditions:



- \* The user is logged in to the system.
- \* A task has been created with no priority level assigned.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Create a new task with no priority level assigned.
3. Click on the "Priority" dropdown menu and select "High".
4. Save the task changes.

Postconditions:

- \* The task priority level is updated to "High".
- \* The task is visually distinguished with a high-priority indicator (e.g., red color).

Expected Result: The user can set a priority level for a task, and the task is visually distinguished based on the priority level.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that tasks can be sorted based on priority levels\*\***

Test Case ID: TP-002

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

Test Case Description: This test case verifies that tasks can be sorted based on their priority levels, ensuring that users can easily identify and focus on high-priority tasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* Multiple tasks have been created with different priority levels (High, Medium, Low).
- \* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Go to the task list view.
3. Click on the "Priority" column header to sort tasks by priority.
4. Verify that tasks are sorted in the correct order (High, Medium, Low).

Postconditions:

- \* Tasks are sorted based on their priority levels.

Expected Result: Tasks can be sorted based on their priority levels, and the task list is updated accordingly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can change the priority level of a task\*\***

Test Case ID: TP-003

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

Test Case Description: This test case verifies that users can change the priority level of a task, ensuring that users can update task priorities as needed.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A task has been created with a priority level assigned (e.g., Medium).
- \* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Select a task with a priority level assigned (e.g., Medium).
3. Click on the "Priority" dropdown menu and select a different priority level (e.g., High).
4. Save the task changes.

Postconditions:

- \* The task priority level is updated to the new priority level (e.g., High).

Expected Result: Users can change the priority level of a task, and the task is updated accordingly.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the system visually distinguishes tasks based on priority\*\***

Test Case ID: TP-004

Test Case Title: Verify that the system visually distinguishes tasks based on priority

Test Case Description: This test case verifies that the system visually distinguishes tasks based on their priority levels, ensuring that users can easily identify high-priority tasks.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* Multiple tasks have been created with different priority levels (High, Medium, Low).
- \* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Go to the task list view.
3. Verify that tasks with different priority levels are visually distinguished (e.g., red color for High, yellow for Medium, green for Low).

Postconditions:

\* Tasks are visually distinguished based on their priority levels.

Expected Result: The system visually distinguishes tasks based on their priority levels, making it easy for users to identify high-priority tasks.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that tasks without a specified priority default to Medium\*\***

Test Case ID: TP-005

Test Case Title: Verify that tasks without a specified priority default to Medium

Test Case Description: This test case verifies that tasks without a specified priority level default to Medium, ensuring that tasks are assigned a default priority level.

Test Suite: Task Management

Test Priority: Low

Preconditions:

\* A task has been created with no priority level assigned.

\* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Create a new task with no priority level assigned.

3. Verify that the task has a default priority level of Medium.

Postconditions:

\* The task priority level is Medium.

Expected Result: Tasks without a specified priority level default to Medium, ensuring that tasks are assigned a default priority level.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 25**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Subtasks

**Feature description -**

Tasks can be broken down into smaller subtasks, enabling users to divide complex tasks into manageable components. Subtasks can have their own due dates, assignees, and dependencies, providing granularity and

clarity in task execution.

### **Userstory title -**

As a user, I want to create subtasks so that I can break down complex tasks into manageable parts.

### **Acceptance criteria -**

['Users can create subtasks under main tasks.', 'Subtasks can have their own due dates and assignees.', 'The system tracks the progress of each subtask.', 'Subtasks can be marked as complete independently of the main task.', 'Dependencies between subtasks can be set.']

### **Generated Test Cases:**

Here are five detailed test cases for the Subtasks feature of the Task Management Tool project:

**\*\*Test Case 1: Verify that users can create subtasks under main tasks\*\***

Test Case ID: TCT-001

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

Test Case Description: This test case verifies that users can create subtasks under main tasks, allowing them to break down complex tasks into manageable parts.

Test Suite: Task Management

Test Priority: High

Preconditions:

\* A main task is created with a valid title and description.

\* The user is logged in and has permission to create subtasks.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with permission to create subtasks.
2. Create a new main task with a title and description.
3. Click on the "Create Subtask" button within the main task.
4. Enter a title and description for the subtask.
5. Click "Save" to create the subtask.

Postconditions:

- \* The subtask is created and displayed under the main task.
- \* The subtask has a unique ID and title.

Expected Result: The user can create a subtask under a main task successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that subtasks can have their own due dates and assignees\*\***

Test Case ID: TCT-002

Test Case Title: Verify that subtasks can have their own due dates and assignees

Test Case Description: This test case verifies that subtasks can have their own due dates and assignees, providing granularity and clarity in task execution.

Test Suite: Task Management

Test Priority: High

Preconditions:



\* A main task is created with a valid title and description.

\* A subtask is created under the main task.

Test Data: Due dates and assignee names

Test Steps:

1. Log in to the system as a user with permission to edit subtasks.
2. Edit the subtask created in the previous test case.
3. Set a due date for the subtask.
4. Assign the subtask to a team member.
5. Click "Save" to update the subtask.

Postconditions:

\* The subtask has a unique due date and assignee.

\* The due date and assignee are displayed correctly.

Expected Result: The subtask can have its own due date and assignee successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system tracks the progress of each subtask\*\***

Test Case ID: TCT-003

Test Case Title: Verify that the system tracks the progress of each subtask

Test Case Description: This test case verifies that the system tracks the progress of each subtask, providing an accurate status of task execution.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A main task is created with a valid title and description.
- \* A subtask is created under the main task.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with permission to view task progress.
2. View the subtask created in the previous test case.
3. Update the subtask status to "In Progress".
4. Verify that the subtask progress is updated correctly.
5. Repeat steps 3-4 for multiple subtasks.

Postconditions:

- \* The system tracks the progress of each subtask correctly.
- \* The subtask progress is updated in real-time.

Expected Result: The system tracks the progress of each subtask accurately.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that subtasks can be marked as complete independently of the main task\*\***

Test Case ID: TCT-004

Test Case Title: Verify that subtasks can be marked as complete independently of the main task

Test Case Description: This test case verifies that subtasks can be marked as complete independently of the main task, allowing users to track progress accurately.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A main task is created with a valid title and description.

- \* A subtask is created under the main task.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with permission to mark subtasks as complete.
2. View the subtask created in the previous test case.
3. Mark the subtask as complete.
4. Verify that the subtask is marked as complete correctly.
5. Verify that the main task status is not affected.

Postconditions:

- \* The subtask is marked as complete independently of the main task.

\* The main task status remains unchanged.

Expected Result: Subtasks can be marked as complete independently of the main task successfully.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that dependencies between subtasks can be set\*\***

Test Case ID: TCT-005

Test Case Title: Verify that dependencies between subtasks can be set

Test Case Description: This test case verifies that dependencies between subtasks can be set, ensuring that tasks are executed in the correct order.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

\* A main task is created with a valid title and description.

\* Two or more subtasks are created under the main task.

Test Data: Dependency relationships between subtasks

Test Steps:

1. Log in to the system as a user with permission to set dependencies between subtasks.
2. View the subtasks created in the previous test case.
3. Set a dependency between two subtasks (e.g., Subtask A depends on Subtask B).

4. Verify that the dependency is set correctly.
5. Try to mark Subtask A as complete without completing Subtask B.

Postconditions:

- \* The dependency between subtasks is set correctly.
- \* The system prevents marking Subtask A as complete without completing Subtask B.

Expected Result: Dependencies between subtasks can be set successfully.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 26**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Dependencies

**Feature description -**

Users can define dependencies between tasks, specifying the order in which tasks should be completed to

ensure smooth workflow and project progression. Task dependencies help in identifying critical paths and managing interdependencies between tasks.

### **Userstory title -**

As a user, I want to set task dependencies so that tasks are completed in the correct sequence.

### **Acceptance criteria -**

['Users can set dependencies between tasks.', 'The system displays dependencies in the task details.', 'Tasks cannot be marked as complete if dependent tasks are not completed.', 'Users can update or remove dependencies as needed.', 'The system provides alerts or warnings if a dependent task is not completed.']

### **Generated Test Cases:**

Here are five detailed test cases for the Task Dependencies feature, ensuring that each acceptance criterion is covered:

**\*\*Test Case 1: Verify that users can set dependencies between tasks\*\***

Test Case ID: TC-TD-001

Test Case Title: Verify that users can set dependencies between tasks

Test Case Description: This test case verifies that users can successfully set dependencies between tasks, ensuring that tasks are completed in the correct sequence.

Test Suite: Task Management Tool - Task Dependencies

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least two tasks are created and available in the task list
- \* No dependencies are set between tasks

Test Data: No test data needed

Test Steps:

1. Create a new task, Task A
2. Create another new task, Task B
3. Navigate to Task B's details page
4. Click on the "Add Dependency" button
5. Select Task A as the dependent task
6. Save the changes

Postconditions:

- \* Task B is now dependent on Task A
- \* The dependency is displayed in Task B's details page

Expected Result: The dependency between Task A and Task B is successfully set, and Task B is displayed as dependent on Task A.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the system displays dependencies in the task details\*\***

Test Case ID: TC-TD-002

Test Case Title: Verify that the system displays dependencies in the task details

Test Case Description: This test case verifies that the system correctly displays dependencies in the task details, providing users with a clear understanding of task relationships.

Test Suite: Task Management Tool - Task Dependencies

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task with dependencies is available in the task list

Test Data: No test data needed

Test Steps:

1. Navigate to a task with dependencies (e.g., Task B from Test Case 1)
2. Verify that the dependent task (Task A) is displayed in the task details page
3. Verify that the dependency relationship is correctly displayed (e.g., "Depends on Task A")

Postconditions: None

Expected Result: The system correctly displays the dependency in the task details page, providing users with a clear understanding of task relationships.

Severity: Critical

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that tasks cannot be marked as complete if dependent tasks are not completed\*\***

Test Case ID: TC-TD-003

Test Case Title: Verify that tasks cannot be marked as complete if dependent tasks are not completed

Test Case Description: This test case verifies that tasks cannot be marked as complete if their dependent tasks



are not completed, ensuring that tasks are completed in the correct sequence.

Test Suite: Task Management Tool - Task Dependencies

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task with dependencies is available in the task list
- \* The dependent task is not completed

Test Data: No test data needed

Test Steps:

1. Navigate to a task with dependencies (e.g., Task B)
2. Attempt to mark Task B as complete
3. Verify that the system prevents the task from being marked as complete
4. Verify that an error message is displayed, indicating that dependent tasks must be completed first

Postconditions: Task B remains in an incomplete state

Expected Result: The system prevents Task B from being marked as complete, ensuring that tasks are completed in the correct sequence.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative

**\*\*Test Case 4: Verify that users can update or remove dependencies as needed\*\***

Test Case ID: TC-TD-004

Test Case Title: Verify that users can update or remove dependencies as needed

Test Case Description: This test case verifies that users can update or remove dependencies between tasks, allowing for flexibility in task management.

Test Suite: Task Management Tool - Task Dependencies

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task with dependencies is available in the task list

Test Data: No test data needed

Test Steps:

1. Navigate to a task with dependencies (e.g., Task B)
2. Update the dependency to a new task (e.g., Task C)
3. Verify that the dependency is updated correctly
4. Remove the dependency
5. Verify that the dependency is removed successfully

Postconditions: The dependency is updated or removed as expected

Expected Result: The system allows users to update or remove dependencies between tasks, providing flexibility in task management.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the system provides alerts or warnings if a dependent task is not completed\*\***

Test Case ID: TC-TD-005

Test Case Title: Verify that the system provides alerts or warnings if a dependent task is not completed

Test Case Description: This test case verifies that the system provides alerts or warnings if a dependent task is not completed, ensuring that users are notified of potential blockers.

Test Suite: Task Management Tool - Task Dependencies

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task with dependencies is available in the task list
- \* The dependent task is not completed

Test Data: No test data needed

Test Steps:

1. Navigate to a task with dependencies (e.g., Task B)
2. Verify that an alert or warning is displayed, indicating that the dependent task (Task A) is not completed
3. Click on the alert or warning
4. Verify that the system redirects to the dependent task (Task A)

Postconditions: The alert or warning is displayed and functional

Expected Result: The system provides alerts or warnings if a dependent task is not completed, ensuring that users are notified of potential blockers.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **Data - 27**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Comments and Collaboration

### **Feature description -**

Users can collaborate on tasks by leaving comments, attaching files, and engaging in discussions directly within the task interface. This feature facilitates communication and teamwork, allowing team members to provide updates, share feedback, and ask questions related to the task.

### **Userstory title -**

As a user, I want to comment on tasks so that I can collaborate with my team.

### **Acceptance criteria -**

['Users can add comments to tasks.', 'Comments can include text, attachments, and links.', 'Users can reply to and edit their comments.', 'The system notifies users of new comments.', 'Comments are displayed in chronological order.']

### **Generated Test Cases:**

Here are five detailed test cases for the Task Comments and Collaboration feature:

**\*\*Test Case 1: Verify that users can add comments to tasks\*\***

Test Case ID: TC001

Test Case Title: Verify that users can add comments to tasks

Test Case Description: This test case verifies that users can successfully add comments to tasks, including text, attachments, and links.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* The user is logged in to the system
- \* A task is created and assigned to the user
- \* The user has necessary permissions to add comments

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the task details page
3. Click on the "Add Comment" button
4. Enter a comment with text, attachment, and link
5. Click on the "Post Comment" button

Postconditions:

- \* The comment is displayed on the task details page
- \* The comment includes the text, attachment, and link

Expected Result: The user can add a comment to a task successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can reply to and edit their comments\*\***

Test Case ID: TC002

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 comments and edit their own comments on a task.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* The user is logged in to the system
- \* A task is created and assigned to the user
- \* The user has necessary permissions to add comments

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user

2. Navigate to the task details page
3. Add a comment to the task
4. Reply to the comment
5. Edit the original comment
6. Save the changes

Postconditions:

- \* The reply is displayed below the original comment
- \* The edited comment is updated on the task details page

Expected Result: The user can reply to and edit their comments successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system notifies users of new comments\*\***

Test Case ID: TC003

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 following.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* The user is logged in to the system
- \* A task is created and assigned to the user
- \* The user is following the task
- \* Another user has necessary permissions to add comments

Test Data: No test data needed

Test Steps:

1. Log in to the system as another user
2. Navigate to the task details page
3. Add a new comment to the task
4. Log back in as the original user
5. Check the notification panel or email notifications

Postconditions:

- \* The notification panel or email shows a new comment notification
- \* The notification links to the task details page

Expected Result: The system sends a notification to users when a new comment is added to a task they are following.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that comments are displayed in chronological order\*\***



Test Case ID: TC004

Test Case Title: Verify that comments are displayed in chronological order

Test Case Description: This test case verifies that comments are displayed in the correct chronological order on the task details page.

Test Suite: Task Management

Test Priority: Low

Preconditions:

- \* The user is logged in to the system
- \* A task is created and assigned to the user
- \* Multiple comments are added to the task

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the task details page
3. Add multiple comments to the task with different timestamps
4. Check the comment order on the task details page

Postconditions:

- \* The comments are displayed in the correct chronological order

Expected Result: Comments are displayed in the correct chronological order on the task details page.

Severity: Trivial

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that invalid input is handled correctly for comment addition\*\***

Test Case ID: TC005

Test Case Title: Verify that invalid input is handled correctly for comment addition

Test Case Description: This test case verifies that the system handles invalid input correctly when adding a comment to a task.

Test Suite: Task Management

Test Priority: Low

Preconditions:

- \* The user is logged in to the system
- \* A task is created and assigned to the user

Test Data:

- \* Special characters (e.g., <script>, etc.)
- \* Very long text (e.g., 1000 characters)

Test Steps:

1. Log in to the system as a user
2. Navigate to the task details page
3. Attempt to add a comment with invalid input (e.g., special characters, very long text)

#### 4. Check the error message or validation response

Postconditions:

- \* The system prevents the invalid input from being added as a comment
- \* An error message or validation response is displayed

Expected Result: The system handles invalid input correctly when adding a comment to a task.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Negative

-----

**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:**

Here are five detailed test cases for the File Attachments feature of the Task Management Tool project:

**\*\*Test Case 1: Verify that multiple files can be attached to a task\*\***

Test Case ID: TTMA-001

Test Case Title: Verify that multiple files can be attached to a task

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* User is logged in to the system
- \* A task is created and available for attachment
- \* Multiple files of different types (e.g., document, image, spreadsheet) are available for upload

Test Data: 3 files of different types (e.g., document, image, spreadsheet)

Test Steps:

1. Log in to the system and navigate to the task management page

2. Select a task and click on the "Attach file" button
3. Upload multiple files (3) of different types
4. Verify that all files are successfully attached to the task
5. Verify that the task page displays all attached files with their respective file names and types

Postconditions:

- \* The task page displays all attached files
- \* The files are stored in the system's database

Expected Result: Multiple files can be attached to a task successfully

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that attachments can include various file types\*\***

Test Case ID: TTMA-002

Test Case Title: Verify that attachments can include various file types

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* User is logged in to the system
- \* A task is created and available for attachment
- \* Various file types (e.g., document, image, spreadsheet, audio, video) are available for upload

Test Data: 5 files of different types (e.g., document, image, spreadsheet, audio, video)

Test Steps:

1. Log in to the system and navigate to the task management page
2. Select a task and click on the "Attach file" button
3. Upload files of different types (5)
4. Verify that all files are successfully attached to the task
5. Verify that the task page displays all attached files with their respective file names and types

Postconditions:

- \* The task page displays all attached files
- \* The files are stored in the system's database

Expected Result: Attachments can include various file types successfully

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can view and download attachments\*\***

Test Case ID: TTMA-003

Test Case Title: Verify that users can view and download attachments

Test Suite: Task Management

Test Priority: High

Preconditions:

\* User is logged in to the system

\* A task with attached files is available

Test Data: No test data needed

Test Steps:

1. Log in to the system and navigate to the task management page
2. Select a task with attached files
3. Click on an attached file to view its contents
4. Verify that the file can be viewed successfully
5. Click on the "Download" button to download the attached file
6. Verify that the file is downloaded successfully

Postconditions:

\* The file is viewed and downloaded successfully

Expected Result: Users can view and download attachments successfully

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the system limits the size of attachments\*\***

Test Case ID: TTMA-004

Test Case Title: Verify that the system limits the size of attachments

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* User is logged in to the system
- \* A task is created and available for attachment
- \* A large file (> allowed size limit) is available for upload

Test Data: A large file (> allowed size limit)

Test Steps:

1. Log in to the system and navigate to the task management page
2. Select a task and click on the "Attach file" button
3. Attempt to upload a large file (> allowed size limit)
4. Verify that the system prevents the upload of the large file
5. Verify that an error message is displayed indicating the size limit exceeded

Postconditions:

- \* The system prevents the upload of the large file
- \* An error message is displayed

Expected Result: The system limits the size of attachments successfully

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

**\*\*Test Case 5: Verify that users can remove attachments\*\***



Test Case ID: TTMA-005

Test Case Title: Verify that users can remove attachments

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* User is logged in to the system
- \* A task with attached files is available

Test Data: No test data needed

Test Steps:

1. Log in to the system and navigate to the task management page
2. Select a task with attached files
3. Click on an attached file and select the "Remove" option
4. Verify that the attached file is removed from the task
5. Verify that the task page is updated to reflect the removed attachment

Postconditions:

- \* The attachment is removed from the task
- \* The task page is updated

Expected Result: Users can remove attachments successfully

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

## Data - 29

### Project name -

Task Management Tool

### Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### Feature name -

Recurring Tasks

### Feature description -

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

### Userstory title -

As a user, I want to set recurring tasks so that regular activities are automated.

### Acceptance criteria -

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

### Generated Test Cases:

Here are five detailed test cases for the Recurring Tasks feature:

**\*\*Test Case 1: Verify that users can set tasks to recur at specified intervals\*\***

Test Case ID: RT001

Test Case Title: Verify that users can set tasks to recur at specified intervals

Test Case Description: This test case verifies that users can set tasks to recur at specified intervals (daily, weekly, monthly).

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* The user is logged in to the Task Management Tool

- \* The user has created a new task

Test Data: No test data needed

Test Steps:

1. Create a new task and set its recurrence pattern to daily
2. Verify that the task is set to recur daily
3. Create another new task and set its recurrence pattern to weekly
4. Verify that the task is set to recur weekly
5. Create another new task and set its recurrence pattern to monthly
6. Verify that the task is set to recur monthly

Postconditions:

- \* The task recurrence pattern is updated successfully

\* The system generates instances of the recurring task automatically

Expected Result: The task is set to recur at the specified interval (daily, weekly, monthly)

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the system generates instances of recurring tasks automatically\*\***

Test Case ID: RT002

Test Case Title: Verify that the system generates instances of recurring tasks automatically

Test Case Description: This test case verifies that the system generates instances of recurring tasks automatically.

Test Suite: Task Management

Test Priority: High

Preconditions:

\* The user is logged in to the Task Management Tool

\* A recurring task is created with a daily recurrence pattern

Test Data: No test data needed

Test Steps:

1. Create a recurring task with a daily recurrence pattern
2. Verify that a new instance of the task is generated automatically the next day
3. Verify that the new instance has the same task details (description, attachments, etc.) as the original task

4. Repeat steps 1-3 for weekly and monthly recurrence patterns

Postconditions:

- \* The system generates new instances of the recurring task automatically
- \* The new instances have the same task details as the original task

Expected Result: The system generates new instances of the recurring task automatically

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can modify the recurrence pattern or cancel the recurrence\*\***

Test Case ID: RT003

Test Case Title: Verify that users can modify the recurrence pattern or cancel the recurrence

Test Case Description: This test case verifies that users can modify the recurrence pattern or cancel the recurrence of a task.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* The user is logged in to the Task Management Tool
- \* A recurring task is created with a daily recurrence pattern

Test Data: No test data needed

#### Test Steps:

1. Modify the recurrence pattern of the task from daily to weekly
2. Verify that the task recurrence pattern is updated successfully
3. Cancel the recurrence of the task
4. Verify that the task is no longer recurring

#### Postconditions:

- \* The task recurrence pattern is updated successfully
- \* The task is no longer recurring

Expected Result: Users can modify the recurrence pattern or cancel the recurrence of a task

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the system notifies users of upcoming instances of recurring tasks\*\***

Test Case ID: RT004

Test Case Title: Verify that the system notifies users of upcoming instances of recurring tasks

Test Case Description: This test case verifies that the system notifies users of upcoming instances of recurring tasks.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* The user is logged in to the Task Management Tool
- \* A recurring task is created with a daily recurrence pattern

Test Data: No test data needed

Test Steps:

1. Create a recurring task with a daily recurrence pattern
2. Verify that the system sends a notification to the user when the next instance of the task is due
3. Verify that the notification includes the task details (description, attachments, etc.)

Postconditions:

- \* The system sends a notification to the user when the next instance of the task is due
- \* The notification includes the task details

Expected Result: The system notifies users of upcoming instances of recurring tasks

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can view a list of all instances of a recurring task\*\***

Test Case ID: RT005

Test Case Title: Verify that users can view a list of all instances of a recurring task

Test Case Description: This test case verifies that users can view a list of all instances of a recurring task.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* The user is logged in to the Task Management Tool
- \* A recurring task is created with a daily recurrence pattern

Test Data: No test data needed

Test Steps:

1. Create a recurring task with a daily recurrence pattern
2. Verify that the system displays a list of all instances of the task
3. Verify that the list includes the task details (description, attachments, etc.) for each instance

Postconditions:

- \* The system displays a list of all instances of the recurring task
- \* The list includes the task details for each instance

Expected Result: Users can view a list of all instances of a recurring task

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----



**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Tags and Labels

**Feature description -**

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

**Userstory title -**

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

**Acceptance criteria -**

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Tags and Labels feature:

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

Test Case ID: TT-001

Test Case Title: Verify that users can add tags and labels to tasks.

Test Case Description: This test case ensures that users can successfully add tags and labels to tasks, making it possible to categorize and organize tasks effectively.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* A user is logged in to the system.
- \* A task is created with a description and deadline.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Create a new task with a description and deadline.
3. Click on the "Add Tag" button next to the task.
4. Enter a valid tag name and click "Save".
5. Repeat steps 3-4 to add multiple tags to the task.

Postconditions:

- \* The task is updated with the added tags.

Expected Result: The task is successfully tagged with the specified labels, and the tags are displayed prominently in the task list.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: TT-002

Test Case Title: Verify that tags can be customized and edited.

Test Case Description: This test case ensures that users can customize and edit tags, making it possible to modify categorization and organization of tasks as needed.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A user is logged in to the system.
- \* A task is created with a description and deadline.
- \* A tag is added to the task.

Test Data: A valid tag name

Test Steps:

1. Log in to the system as a user.
2. Locate the task with the added tag.
3. Click on the "Edit Tag" button next to the task.
4. Modify the tag name and click "Save".
5. Verify that the tag is updated successfully.

Postconditions:

- \* The task is updated with the modified tag.

Expected Result: The tag is successfully customized and edited, and the changes are reflected in the task list.

Severity: Critical

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: TT-003

Test Case Title: Verify that tasks can be filtered and sorted based on tags and labels.

Test Case Description: This test case ensures that users can filter and sort tasks based on tags and labels, making it easier to organize and prioritize tasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* A user is logged in to the system.
- \* Multiple tasks are created with different tags and labels.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Click on the "Filter" button in the task list.
3. Select a tag or label to filter tasks.
4. Verify that only tasks with the selected tag or label are displayed.
5. Repeat steps 2-4 to test sorting tasks by different tags and labels.

Postconditions:

\* The task list is updated with the filtered and sorted tasks.

Expected Result: Tasks are successfully filtered and sorted based on tags and labels, making it easier to organize and prioritize tasks.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: TT-004

Test Case Title: Verify that the system displays tags and labels prominently in the task list.

Test Case Description: This test case ensures that tags and labels are prominently displayed in the task list, making it easier for users to identify and organize tasks.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

\* A user is logged in to the system.

\* A task is created with a description and deadline.

\* A tag is added to the task.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.

2. Locate the task with the added tag.
3. Verify that the tag is displayed prominently next to the task.
4. Repeat steps 2-3 to test multiple tasks with different tags and labels.

Postconditions:

\* The task list is updated with the displayed tags and labels.

Expected Result: Tags and labels are prominently displayed in the task list, making it easier for users to identify and organize tasks.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: TT-005

Test Case Title: Verify that users can remove tags and labels if needed.

Test Case Description: This test case ensures that users can remove tags and labels from tasks, making it possible to modify categorization and organization of tasks as needed.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

\* A user is logged in to the system.

\* A task is created with a description and deadline.

\* A tag is added to the task.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Locate the task with the added tag.
3. Click on the "Remove Tag" button next to the task.
4. Verify that the tag is removed successfully.
5. Repeat steps 2-4 to test removing multiple tags and labels.

Postconditions:

\* The task is updated without the removed tag.

Expected Result: Tags and labels are successfully removed from tasks, making it possible to modify categorization and organization of tasks as needed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 31**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Status Tracking

**Feature description -**

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

**Userstory title -**

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

**Acceptance criteria -**

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Status Tracking feature:

**\*\*Test Case 1: Verify that users can update the status of tasks\*\***

Test Case ID: TS-001

Test Case Title: Verify that users can update the status of tasks

Test Case Description: This test case verifies that users can update the status of tasks to track progress.

Test Suite: Task Management

Test Priority: High

Preconditions:



- \* The user is logged in to the system.
- \* A task is created with a default status of "To Do".

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Navigate to the task list and select a task with a status of "To Do".
3. Click on the "Update Status" button.
4. Select a new status (e.g., "In Progress") from the dropdown list.
5. Click "Save" to update the task status.

Postconditions:

- \* The task status is updated in the database.
- \* The updated task status is reflected in the task list.

Expected Result: The task status is successfully updated to the new status.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the system displays task status prominently in the task list\*\***

Test Case ID: TS-002

Test Case Title: Verify that the system displays task status prominently in the task list

Test Case Description: This test case verifies that the system displays task status prominently in the task list for easy tracking.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* The user is logged in to the system.

- \* Multiple tasks are created with different statuses (e.g., "To Do", "In Progress", "Completed").

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Navigate to the task list.
3. Verify that each task in the list displays its current status prominently (e.g., as a badge or icon).

Postconditions:

- \* None

Expected Result: The task status is displayed prominently in the task list for each task.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can view the progress percentage of tasks\*\***

Test Case ID: TS-003

Test Case Title: Verify that users can view the progress percentage of tasks

Test Case Description: This test case verifies that users can view the progress percentage of tasks to track progress.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* The user is logged in to the system.
- \* A task is created with a progress percentage (e.g., 50%).

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Navigate to the task list and select a task with a progress percentage.
3. Verify that the task details page displays the progress percentage.

Postconditions:

- \* None

Expected Result: The progress percentage is displayed accurately for each task.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the system provides notifications for status changes\*\***

Test Case ID: TS-004

Test Case Title: Verify that the system provides notifications for status changes

Test Case Description: This test case verifies that the system sends notifications to users when task statuses change.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* The user is logged in to the system.
- \* A task is created with a default status of "To Do".
- \* The user has enabled notifications for task status changes.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Navigate to the task list and select a task with a status of "To Do".
3. Update the task status to "In Progress".
4. Verify that a notification is sent to the user (e.g., email or in-app notification).

Postconditions:

- \* The notification is sent to the user's email or in-app notification center.

Expected Result: A notification is sent to the user when the task status changes.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that task status can be filtered and sorted\*\***

Test Case ID: TS-005

Test Case Title: Verify that task status can be filtered and sorted

Test Case Description: This test case verifies that users can filter and sort tasks by status to easily track progress.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* The user is logged in to the system.

- \* Multiple tasks are created with different statuses (e.g., "To Do", "In Progress", "Completed").

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user.
2. Navigate to the task list.
3. Click on the "Filter" button and select a specific status (e.g., "In Progress").
4. Verify that only tasks with the selected status are displayed.
5. Click on the "Sort" button and select "Status" as the sorting criteria.
6. Verify that tasks are sorted by status (e.g., "To Do" first, then "In Progress", then "Completed").

Postconditions:

\* None

Expected Result: Tasks are filtered and sorted by status correctly.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 32**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Filtering and Sorting

**Feature description -**

Users can filter and sort tasks based on various criteria such as due date, priority, assignee, or project. Filtering and sorting options help users focus on specific tasks or view tasks from different perspectives, enhancing task management and organization.

**Userstory title -**

As a user, I want to filter and sort tasks so that I can focus on specific tasks or view tasks from different perspectives.

**Acceptance criteria -**

['Users can filter tasks based on criteria such as due date, priority, assignee, or project.', 'Tasks can be sorted by various attributes.', 'The system provides multiple filter and sort options in the task list.', 'Users can save and apply custom filter and sort settings.', 'Filters and sorts do not affect the actual data, only the view.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Filtering and Sorting feature of the Task Management Tool:

**\*\*Test Case 1: Verify that users can filter tasks by due date\*\***

Test Case ID: TF001

Test Case Title: Verify that users can filter tasks by due date

Test Case Description: This test case verifies that users can filter tasks based on due date, allowing them to focus on tasks with nearing deadlines.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

\* At least three tasks with different due dates exist in the system.

\* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system and navigate to the task list.
2. Click on the "Filter" button and select "Due Date" from the dropdown menu.
3. Select a specific date range (e.g., "Today", "This Week", or "Next Week") and click "Apply".
4. Verify that the task list is updated to display only tasks with due dates within the selected range.

Postconditions:

\* The task list is updated to display only tasks with due dates within the selected range.

Expected Result: The task list is successfully filtered by due date, displaying only tasks with due dates within the selected range.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

\*\*Test Case 2: Verify that tasks can be sorted by priority\*\*

Test Case ID: TS002



Test Case Title: Verify that tasks can be sorted by priority

Test Case Description: This test case verifies that tasks can be sorted by priority, allowing users to focus on high-priority tasks first.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- \* At least three tasks with different priorities (High, Medium, Low) exist in the system.
- \* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system and navigate to the task list.
2. Click on the "Sort" button and select "Priority" from the dropdown menu.
3. Select the desired sorting order (e.g., "High to Low" or "Low to High") and click "Apply".
4. Verify that the task list is updated to display tasks in the selected priority order.

Postconditions:

- \* The task list is updated to display tasks in the selected priority order.

Expected Result: The task list is successfully sorted by priority, displaying tasks in the selected order.

Severity: Major

Priority: High

Type of Testing: Functional Testing

## Test Case Approach: Positive

**\*\*Test Case 3: Verify that filters and sorts do not affect the actual data\*\***

Test Case ID: TF003

Test Case Description: This test case verifies that applying filters and sorts does not modify the actual task data, only the view.

Test Suite: Task Filtering and Sorting

Test Priority: Medium

Preconditions:

- \* At least three tasks with different due dates and priorities exist in the system.
- \* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system and navigate to the task list.
2. Apply a filter (e.g., by due date) and a sort (e.g., by priority).
3. Verify that the task list is updated accordingly.
4. Remove the filter and sort and verify that the original task list is displayed.

Postconditions:

- \* The original task list is displayed after removing the filter and sort.

Expected Result: Applying filters and sorts does not modify the actual task data, only the view.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can save and apply custom filter and sort settings\*\***

Test Case ID: TF004

Test Case Title: Verify that users can save and apply custom filter and sort settings

Test Case Description: This test case verifies that users can save and apply custom filter and sort settings, allowing them to reuse their preferred settings.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

\* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system and navigate to the task list.
2. Apply a custom filter (e.g., by due date and priority) and sort (e.g., by priority).
3. Click on the "Save" button and name the custom setting (e.g., "My Favorite View").
4. Log out and log back in to the system.
5. Click on the "Load" button and select the saved custom setting.
6. Verify that the task list is updated accordingly.

Postconditions:

- \* The task list is updated to display the saved custom filter and sort settings.

Expected Result: Users can successfully save and apply custom filter and sort settings.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that multiple filter and sort options are provided in the task list\*\***

Test Case ID: TS005

Test Case Title: Verify that multiple filter and sort options are provided in the task list

Test Case Description: This test case verifies that the system provides multiple filter and sort options in the task list, allowing users to customize their view.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- \* The user is logged in to the system.

Test Data: No test data needed

Test Steps:

1. Log in to the system and navigate to the task list.
2. Click on the "Filter" button and verify that multiple filter options are available (e.g., due date, priority, assignee, project).
3. Click on the "Sort" button and verify that multiple sort options are available (e.g., due date, priority, task name).

Postconditions:

\* Multiple filter and sort options are available in the task list.

Expected Result: The system provides multiple filter and sort options in the task list, allowing users to customize their view.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 33**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Time Tracking

**Feature description -**

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

**Userstory title -**

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

**Acceptance criteria -**

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

**Generated Test Cases:**

Here are five detailed test cases that cover each acceptance criterion for the Time Tracking feature of the Task Management Tool:

**\*\*Test Case 1: Verify that users can start and stop a timer for each task\*\***

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 that users can start and stop a timer for individual tasks, enabling accurate time tracking.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- \* The user has created a new task
- \* The task is not already being tracked

Test Data: No test data needed

Test Steps:

1. Log in to the application as a registered user
2. Create a new task with a unique name and description
3. Click on the "Start Timer" button for the created task
4. Verify that the timer starts counting up
5. Click on the "Stop Timer" button
6. Verify that the timer stops and the total time spent is displayed

Postconditions:

- \* The task's timer is stopped
- \* The total time spent on the task is updated

Expected Result: The user can successfully start and stop a timer for each task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the system records and displays total time spent on tasks\*\***

Test Case ID: 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.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

\* The user has started and stopped a timer for a task

Test Data: No test data needed

Test Steps:

1. Log in to the application as a registered user
2. Start and stop a timer for a task multiple times
3. Verify that the total time spent on the task is updated accurately
4. Verify that the total time spent is displayed correctly on the task details page

Postconditions:

\* The total time spent on the task is updated

\* The total time spent is displayed correctly on the task details page

Expected Result: The system accurately records and displays the total time spent on each task.

Severity: Major

Priority: High



Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can manually enter or adjust time entries\*\***

Test Case ID: TT-03

Test Case Title: Verify that users can manually enter or adjust time entries

Test Case Description: This test case verifies that users can manually enter or adjust time entries for tasks.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

\* The user has created a new task

Test Data:

\* Task name: "Test Task"

\* Time entry: 2 hours 30 minutes

Test Steps:

1. Log in to the application as a registered user
2. Create a new task with the specified task name
3. Click on the "Add Time Entry" button
4. Enter the specified time entry manually
5. Verify that the time entry is saved successfully
6. Edit the time entry to a new value (e.g., 3 hours)

7. Verify that the edited time entry is saved successfully

Postconditions:

- \* The time entry is saved successfully

- \* The edited time entry is saved successfully

Expected Result: Users can successfully manually enter or adjust time entries for tasks.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that time tracking data is stored and can be exported\*\***

Test Case ID: TT-04

Test Case Title: Verify that time tracking data is stored and can be exported

Test Case Description: This test case verifies that time tracking data is stored and can be exported in a readable format.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

- \* The user has tracked time for multiple tasks

Test Data: No test data needed

#### Test Steps:

1. Log in to the application as a registered user
2. Track time for multiple tasks
3. Click on the "Export Time Tracking Data" button
4. Select a format (e.g., CSV, Excel)
5. Verify that the time tracking data is exported successfully

#### Postconditions:

- \* The time tracking data is exported successfully
- \* The exported data is in the selected format

Expected Result: Time tracking data is stored and can be exported in a readable format.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the system provides reports on time spent per task or project\*\***

Test Case ID: TT-05

Test Case Title: Verify that the system provides reports on time spent per task or project

Test Case Description: This test case verifies that the system generates accurate reports on time spent per task or project.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- \* The user has tracked time for multiple tasks and projects

Test Data: No test data needed

Test Steps:

1. Log in to the application as a registered user
2. Track time for multiple tasks and projects
3. Click on the "Time Tracking Reports" button
4. Select a report type (e.g., task-wise, project-wise)
5. Verify that the report is generated accurately

Postconditions:

- \* The report is generated successfully
- \* The report displays accurate time tracking data

Expected Result: The system generates accurate reports on time spent per task or project.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Templates

**Feature description -**

Users can create and use templates for common task types or project workflows, streamlining task creation and standardizing processes across projects. Task templates can include predefined task structures, descriptions, assignees, and due dates, saving time and ensuring consistency in task execution.

**Userstory title -**

As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

**Acceptance criteria -**

['Users can create and save task templates.', 'Templates can include predefined task structures, descriptions, assignees, and due dates.', 'Users can apply templates to new tasks or projects.', 'Templates can be edited or deleted as needed.', 'The system displays a list of available templates for selection.']

**Generated Test Cases:**

Here are the test cases for the Task Templates feature:

**\*\*Test Case 1: Verify that users can create and save task templates\*\***

Test Case ID: TCT-001

Test Case Title: Verify that users can create and save task templates

Test Case Description: This test case verifies that users can create and save task templates with predefined task structures, descriptions, assignees, and due dates.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- \* The user is logged in to the system with the necessary permissions.
- \* The user has access to the Task Templates feature.
- \* No existing templates are present.

Test Data: Template data (task structure, description, assignee, due date)

Test Steps:

1. Log in to the system as a user with the necessary permissions.
2. Navigate to the Task Templates feature.
3. Click on the "Create Template" button.
4. Enter the template data (task structure, description, assignee, due date).
5. Click on the "Save Template" button.
6. Verify that the template is created and saved successfully.

Postconditions:

- \* The template is displayed in the list of available templates.
- \* The template data is correctly stored in the system.

Expected Result: The user can create and save a task template with the specified data.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that templates can be edited or deleted\*\***

Test Case ID: TCT-002

Test Case Title: Verify that templates can be edited or deleted

Test Case Description: This test case verifies that users can edit or delete existing task templates.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- \* The user is logged in to the system with the necessary permissions.
- \* An existing template is present.

Test Data: Updated template data (task structure, description, assignee, due date)

Test Steps:

1. Log in to the system as a user with the necessary permissions.
2. Navigate to the Task Templates feature.
3. Select an existing template.
4. Click on the "Edit Template" button.
5. Update the template data (task structure, description, assignee, due date).
6. Click on the "Save Changes" button.
7. Verify that the template is updated successfully.

8. Click on the "Delete Template" button.
9. Verify that the template is deleted successfully.

Postconditions:

- \* The updated template data is correctly stored in the system.
- \* The template is no longer displayed in the list of available templates.

Expected Result: The user can edit or delete an existing task template.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can apply templates to new tasks or projects\*\***

Test Case ID: TCT-003

Test Case Title: Verify that users can apply templates to new tasks or projects

Test Case Description: This test case verifies that users can apply task templates to new tasks or projects.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- \* The user is logged in to the system with the necessary permissions.
- \* An existing template is present.

Test Data: Task or project data



#### Test Steps:

1. Log in to the system as a user with the necessary permissions.
2. Navigate to the Task Creation feature.
3. Select an existing template from the list of available templates.
4. Click on the "Apply Template" button.
5. Verify that the template is applied to the new task or project.
6. Verify that the task structure, description, assignee, and due date are populated correctly.

#### Postconditions:

\* The new task or project is created with the applied template data.

Expected Result: The user can apply a task template to a new task or project.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the system displays a list of available templates for selection\*\***

Test Case ID: TCT-004

Test Case Title: Verify that the system displays a list of available templates for selection

Test Case Description: This test case verifies that the system displays a list of available task templates for selection.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

\* The user is logged in to the system with the necessary permissions.

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with the necessary permissions.
2. Navigate to the Task Creation feature.
3. Verify that the list of available templates is displayed.
4. Verify that each template is displayed with its name, description, and creation date.

Postconditions:

\* The list of available templates is updated correctly.

Expected Result: The system displays a list of available task templates for selection.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that template names are unique\*\***

Test Case ID: TCT-005

Test Case Title: Verify that template names are unique

Test Case Description: This test case verifies that users cannot create templates with duplicate names.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- \* The user is logged in to the system with the necessary permissions.
- \* An existing template is present.

Test Data: Template data with duplicate name

Test Steps:

1. Log in to the system as a user with the necessary permissions.
2. Navigate to the Task Templates feature.
3. Click on the "Create Template" button.
4. Enter the template data with a duplicate name.
5. Click on the "Save Template" button.
6. Verify that an error message is displayed indicating that the template name is already in use.

Postconditions:

- \* The duplicate template is not created.

Expected Result: The system prevents the creation of templates with duplicate names.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

---

## **Data - 35**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Calendar Integration

### **Feature description -**

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

### **Userstory title -**

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

### **Acceptance criteria -**

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

### **Generated Test Cases:**

Here are five detailed test cases for the Calendar Integration feature:

**\*\*Test Case 1: Verify that tasks can be synced with external calendars\*\***

Test Case ID: TC-CI-001

Test Case Title: Verify that tasks can be synced with external calendars

Test Case Description: This test case verifies that users can sync tasks with external calendars, such as Google Calendar or Outlook.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- \* A user account with calendar integration enabled
- \* A task with a deadline set in the Task Management Tool

Test Data: None

Test Steps:

1. Log in to the Task Management Tool as a user with calendar integration enabled.
2. Create a new task with a deadline set in the future.
3. Go to the calendar integration settings and connect to an external calendar (e.g., Google Calendar).
4. Verify that the task is synced with the external calendar.
5. Check the external calendar to ensure the task deadline is displayed correctly.

Postconditions:

- \* The task is synced with the external calendar
- \* The task deadline is displayed correctly in the external calendar

Expected Result: The task is successfully synced with the external calendar, and the deadline is displayed correctly.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that task deadlines are displayed in the calendar view\*\***

Test Case ID: TC-CI-002

Test Case Title: Verify that task deadlines are displayed in the calendar view

Test Case Description: This test case verifies that task deadlines are displayed in the calendar view within the Task Management Tool.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- \* A user account with calendar integration enabled
- \* A task with a deadline set in the Task Management Tool

Test Data: None

Test Steps:

1. Log in to the Task Management Tool as a user with calendar integration enabled.
2. Create a new task with a deadline set in the future.
3. Go to the calendar view and verify that the task deadline is displayed correctly.
4. Check that the deadline is displayed in the correct format (e.g., date and time).

Postconditions:

- \* The task deadline is displayed correctly in the calendar view

- \* The deadline is displayed in the correct format

Expected Result: The task deadline is successfully displayed in the calendar view, and the format is correct.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that reminders can be set for tasks via the calendar\*\***

Test Case ID: TC-CI-003

Test Case Title: Verify that reminders can be set for tasks via the calendar

Test Case Description: This test case verifies that users can set reminders for tasks via the calendar integration.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- \* A user account with calendar integration enabled

- \* A task with a deadline set in the Task Management Tool

Test Data: None

Test Steps:

1. Log in to the Task Management Tool as a user with calendar integration enabled.
2. Create a new task with a deadline set in the future.

3. Go to the calendar integration settings and enable reminders for the task.
4. Verify that a reminder is sent to the user before the deadline.
5. Check that the reminder is displayed correctly in the external calendar.

Postconditions:

- \* A reminder is sent to the user before the deadline
- \* The reminder is displayed correctly in the external calendar

Expected Result: A reminder is successfully sent to the user before the deadline, and it is displayed correctly in the external calendar.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that calendar events update automatically when tasks are changed\*\***

Test Case ID: TC-CI-004

Test Case Title: Verify that calendar events update automatically when tasks are changed

Test Case Description: This test case verifies that calendar events update automatically when tasks are changed.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- \* A user account with calendar integration enabled



\* A task with a deadline set in the Task Management Tool

\* The task is synced with an external calendar

Test Data: None

Test Steps:

1. Log in to the Task Management Tool as a user with calendar integration enabled.
2. Create a new task with a deadline set in the future.
3. Sync the task with an external calendar.
4. Update the task deadline to a new date and time.
5. Verify that the calendar event is updated automatically to reflect the new deadline.

Postconditions:

\* The calendar event is updated automatically to reflect the new deadline

\* The updated deadline is displayed correctly in the external calendar

Expected Result: The calendar event is successfully updated automatically to reflect the new deadline, and the updated deadline is displayed correctly in the external calendar.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

\*\*Test Case 5: Verify that users can manage calendar sync settings within the app\*\*

Test Case ID: TC-CI-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 calendar sync settings within the Task Management Tool.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

\* A user account with calendar integration enabled

Test Data: None

Test Steps:

1. Log in to the Task Management Tool as a user with calendar integration enabled.
2. Go to the calendar integration settings and verify that the sync settings can be managed.
3. Disable calendar sync and verify that tasks are no longer synced with the external calendar.
4. Enable calendar sync again and verify that tasks are synced correctly.

Postconditions:

\* Calendar sync settings can be managed within the app

\* Tasks are synced correctly when calendar sync is enabled

Expected Result: Users can successfully manage calendar sync settings within the app, and tasks are synced correctly when enabled.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

## **Data - 36**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Duplication

### **Feature description -**

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

### **Userstory title -**

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

### **Acceptance criteria -**

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

### **Generated Test Cases:**

Here are five detailed test cases for the Task Duplication feature, ensuring comprehensive coverage of the acceptance criteria:

**\*\*Test Case 1: Verify that duplicated tasks retain all original details\*\***

Test Case ID: TC-TD-001

Test Case Title: Verify that duplicated tasks retain all original details

Test Case Description: This test case ensures that when a task is duplicated, it retains all original details, including descriptions, assignees, and deadlines.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* A task with description, assignee, and deadline exists in the system
- \* The user has permission to duplicate tasks

Test Data:

- \* Original task with description, assignee, and deadline
- \* No test data needed

Test Steps:

1. Log in to the system as a user with permission to duplicate tasks
2. Navigate to the task list and select a task with description, assignee, and deadline
3. Click the "Duplicate" button
4. Verify that the duplicated task retains the original task's description, assignee, and deadline
5. Verify that the duplicated task is saved and displayed in the task list

Postconditions:

- \* The original task remains unchanged
- \* The duplicated task is saved and displayed in the task list

Expected Result: The duplicated task retains all original details, including description, assignee, and deadline.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can modify the duplicated task\*\***

Test Case ID: TC-TD-002

Test Case Title: Verify that users can modify the duplicated task

Test Case Description: This test case ensures that users can modify the duplicated task, changing its details as needed.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A duplicated task exists in the system
- \* The user has permission to edit tasks

Test Data:

\* Duplicated task

\* Modified task details (e.g., new description, assignee, deadline)

#### Test Steps:

1. Log in to the system as a user with permission to edit tasks
2. Navigate to the task list and select the duplicated task
3. Edit the duplicated task, changing its details (e.g., description, assignee, deadline)
4. Verify that the changes are saved successfully
5. Verify that the modified duplicated task is displayed in the task list

#### Postconditions:

\* The original task remains unchanged

\* The modified duplicated task is saved and displayed in the task list

Expected Result: Users can modify the duplicated task, changing its details as needed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system clearly indicates that a task has been duplicated\*\***

Test Case ID: TC-TD-003

Test Case Title: Verify that the system clearly indicates that a task has been duplicated

Test Case Description: This test case ensures that the system clearly indicates when a task has been duplicated, providing a visual cue to the user.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A task has been duplicated in the system

Test Data:

- \* No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the task list and select the duplicated task
3. Verify that the system displays a clear indication that the task has been duplicated (e.g., "Duplicated from [Original Task Name]")
4. Verify that the indication is consistent across the system (e.g., in task list, task details page)

Postconditions:

- \* The duplicated task is saved and displayed in the task list

Expected Result: The system clearly indicates that a task has been duplicated, providing a visual cue to the user.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that duplicated tasks are saved and displayed in the task list\*\***

Test Case ID: TC-TD-004

Test Case Title: Verify that duplicated tasks are saved and displayed in the task list

Test Case Description: This test case ensures that duplicated tasks are saved and displayed in the task list, allowing users to access and manage them.

Test Suite: Task Management

Test Priority: High

Preconditions:

\* A task has been duplicated in the system

Test Data:

\* No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the task list
3. Verify that the duplicated task is displayed in the task list
4. Verify that the duplicated task can be accessed and managed from the task list



5. Verify that the duplicated task is saved successfully

Postconditions:

\* The duplicated task is saved and displayed in the task list

Expected Result: Duplicated tasks are saved and displayed in the task list, allowing users to access and manage them.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that task duplication fails when no permission\*\***

Test Case ID: TC-TD-005

Test Case Title: Verify that task duplication fails when no permission

Test Case Description: This test case ensures that task duplication fails when the user does not have permission to duplicate tasks, preventing unauthorized actions.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

\* A user without permission to duplicate tasks exists

\* A task exists in the system

#### Test Data:

- \* Task to be duplicated
- \* User without permission to duplicate tasks

#### Test Steps:

1. Log in to the system as a user without permission to duplicate tasks
2. Navigate to the task list and select a task
3. Attempt to duplicate the task
4. Verify that the system prevents the duplication and displays an error message
5. Verify that the original task remains unchanged

#### Postconditions:

- \* The original task remains unchanged
- \* The duplication attempt is blocked

Expected Result: Task duplication fails when the user does not have permission, preventing unauthorized actions.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Negative

-----

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task History

**Feature description -**

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

**Userstory title -**

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

**Acceptance criteria -**

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

**Generated Test Cases:**

Here are five test cases for the Task History feature:

**\*\*Test Case 1: Verify that users can view a detailed history of changes to tasks\*\***

Test Case ID: TH-001

Test Case Title: Verify that users can view a detailed history of changes to tasks

Test Case Description: This test case ensures that users can view the history of changes to tasks, including modifications, comments, and other relevant activity.

Test Suite: Task History

Test Priority: High

Preconditions:

- \* A user is logged in to the Task Management Tool

- \* A task with modifications, comments, and other relevant activity exists

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Navigate to the task details page for a task with modifications, comments, and other relevant activity
3. Click on the "Task History" tab
4. Verify that the task history is displayed, showing all modifications, comments, and other relevant activity

Postconditions:

- \* None

Expected Result: The task history is displayed, showing all modifications, comments, and other relevant activity

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that task history includes modifications, comments, and other relevant activity\*\***

Test Case ID: TH-002

Test Case Title: Verify that task history includes modifications, comments, and other relevant activity

Test Case Description: This test case ensures that the task history includes all types of changes, including modifications, comments, and other relevant activity.

Test Suite: Task History

Test Priority: Medium

Preconditions:

\* A user is logged in to the Task Management Tool

\* A task with multiple modifications, comments, and other relevant activity exists

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Navigate to the task details page for a task with multiple modifications, comments, and other relevant activity
3. Click on the "Task History" tab
4. Verify that the task history includes all modifications, comments, and other relevant activity
5. Verify that each type of change is correctly labeled (e.g., "Updated Task Description", "Added Comment", etc.)

Postconditions:

\* None

Expected Result: The task history includes all modifications, comments, and other relevant activity, with each type of change correctly labeled

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can filter the task history by type of change or date\*\***

Test Case ID: TH-003

Test Case Title: Verify that users can filter the task history by type of change or date

Test Case Description: This test case ensures that users can filter the task history by type of change or date, making it easier to find specific changes.

Test Suite: Task History

Test Priority: Medium

Preconditions:

\* A user is logged in to the Task Management Tool

\* A task with multiple modifications, comments, and other relevant activity exists

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Navigate to the task details page for a task with multiple modifications, comments, and other relevant activity

3. Click on the "Task History" tab
4. Verify that filter options are available for type of change (e.g., "Modifications", "Comments", etc.) and date
5. Filter the task history by type of change (e.g., "Modifications") and verify that only modifications are displayed
6. Filter the task history by date (e.g., "Today") and verify that only changes from the selected date are displayed

Postconditions:

\* None

Expected Result: The task history can be filtered by type of change or date, and only the corresponding changes are displayed

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that timestamps are provided for all changes in the task history\*\***

Test Case ID: TH-004

Test Case Title: Verify that timestamps are provided for all changes in the task history

Test Case Description: This test case ensures that timestamps are provided for all changes in the task history, allowing users to track when changes were made.

Test Suite: Task History

Test Priority: Low

Preconditions:

\* A user is logged in to the Task Management Tool

\* A task with multiple modifications, comments, and other relevant activity exists

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Navigate to the task details page for a task with multiple modifications, comments, and other relevant activity
3. Click on the "Task History" tab
4. Verify that each change in the task history has a corresponding timestamp (e.g., "Updated 10 minutes ago")

Postconditions:

\* None

Expected Result: Timestamps are provided for all changes in the task history

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that task history is accessible from the task details page\*\***

Test Case ID: TH-005

Test Case Title: Verify that task history is accessible from the task details page

Test Case Description: This test case ensures that the task history is accessible from the task details page, making it easy for users to view changes made to a task.



Test Suite: Task History

Test Priority: High

Preconditions:

\* A user is logged in to the Task Management Tool

\* A task exists

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Navigate to the task details page for a task
3. Verify that a "Task History" tab or button is visible
4. Click on the "Task History" tab or button and verify that the task history is displayed

Postconditions:

\* None

Expected Result: The task history is accessible from the task details page

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Notifications

**Feature description -**

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

**Userstory title -**

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

**Acceptance criteria -**

['Users receive notifications for task updates, comments, mentions, and other relevant activities.',  
'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.',  
'The system logs notification history for future reference.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Notifications feature using the provided template:

**\*\*Test Case 1: Verify that users receive notifications for task updates\*\***

Test Case ID: TN001

Test Case Title: Verify that users receive notifications for task updates

Test Case Description: This test case verifies that users receive notifications when a task is updated, ensuring they stay informed about ongoing task progress.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- \* User is logged in to the task management tool
- \* User has enabled notifications for task updates in their settings
- \* A task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Update a task assigned to the user (e.g., change the task description or due date)
2. Verify that the user receives a notification about the task update (via email, mobile push notification, or in-app alert)
3. Check the notification content to ensure it provides detailed information about the task update

Postconditions:

- \* The user has received a notification about the task update
- \* The notification contains relevant information about the task update

Expected Result: The user receives a notification about the task update, providing detailed information about the change.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can customize notification settings based on preferences\*\***

Test Case ID: TN002

Test Case Title: Verify that users can customize notification settings based on preferences

Test Case Description: This test case verifies that users can customize their notification settings to receive notifications only for specific task updates, ensuring they receive relevant information.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

\* User is logged in to the task management tool

\* User has access to notification settings

Test Data:

\* User preferences for notification settings (e.g., specific task updates, frequency of notifications)

Test Steps:

1. Configure user notification settings to receive notifications only for specific task updates (e.g., task assignments, deadlines)
2. Update a task that meets the user's notification settings (e.g., assign a task to the user)

3. Verify that the user receives a notification about the task update
4. Update a task that does not meet the user's notification settings (e.g., update a task description)
5. Verify that the user does not receive a notification about the task update

Postconditions:

- \* The user has customized their notification settings
- \* The system logs the user's notification preferences

Expected Result: The user receives notifications only for task updates that match their customized notification settings.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that notifications provide detailed information about the activity\*\***

Test Case ID: TN003

Test Case Title: Verify that notifications provide detailed information about the activity

Test Case Description: This test case verifies that notifications provide detailed information about the task update or activity, ensuring users are well-informed about ongoing task progress.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- \* User is logged in to the task management tool
- \* User has enabled notifications for task updates in their settings
- \* A task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Update a task assigned to the user (e.g., change the task description or due date)
2. Verify that the user receives a notification about the task update
3. Check the notification content to ensure it provides detailed information about the task update, including:
  - \* Task ID and name
  - \* Update description (e.g., "Task description updated")
  - \* Update details (e.g., new due date)

Postconditions:

- \* The user has received a notification about the task update
- \* The notification contains detailed information about the task update

Expected Result: The notification provides detailed information about the task update, enabling the user to take informed action.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the system logs notification history for future reference\*\***

Test Case ID: TN004

Test Case Title: Verify that the system logs notification history for future reference

Test Case Description: This test case verifies that the system logs notification history, ensuring users can review past notifications and track task progress over time.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- \* User is logged in to the task management tool
- \* User has enabled notifications for task updates in their settings
- \* A task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Update a task assigned to the user (e.g., change the task description or due date)
2. Verify that the user receives a notification about the task update
3. Check the system logs to ensure the notification is recorded in the notification history
4. Repeat steps 1-3 multiple times, updating the task in different ways (e.g., adding a comment, changing the task status)
5. Verify that the system logs all notifications in the notification history, including:
  - \* Notification type (e.g., task update, comment)
  - \* Notification content
  - \* Timestamp of notification

Postconditions:

- \* The system has logged notification history for the user
- \* The notification history is accessible for future reference

Expected Result: The system logs all notifications in the notification history, enabling users to review past notifications and track task progress over time.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users do not receive redundant notifications\*\***

Test Case ID: TN005

Test Case Title: Verify that users do not receive redundant notifications

Test Case Description: This test case verifies that users do not receive redundant notifications for the same task update, ensuring they are not overwhelmed with unnecessary information.

Test Suite: Task Notifications

Test Priority: Low

Preconditions:

- \* User is logged in to the task management tool
- \* User has enabled notifications for task updates in their settings
- \* A task is created and assigned to the user



Test Data: No test data needed

Test Steps:

1. Update a task assigned to the user (e.g., change the task description or due date)
2. Verify that the user receives a notification about the task update
3. Immediately update the task again (e.g., change the task status)
4. Verify that the user does not receive a redundant notification about the same task update
5. Check the system logs to ensure that only one notification was sent for the task update

Postconditions:

- \* The user has received one notification about the task update
- \* The system has prevented redundant notifications from being sent

Expected Result: The user does not receive redundant notifications for the same task update, reducing notification fatigue.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Negative

-----

**Data - 39**

**Project name -**

Task Management Tool

## **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

## **Feature name -**

Task Archiving

## **Feature description -**

Completed or inactive tasks can be archived or moved to a separate section, decluttering the task list and maintaining focus on active tasks. Task archiving helps users keep their task lists organized and prioritized, without losing access to important historical data.

## **Userstory title -**

As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks.

## **Acceptance criteria -**

['Users can archive completed or inactive tasks.', 'Archived tasks are moved to a separate section for easy access.', 'Users can search and view archived tasks.', 'The system provides an option to restore archived tasks to the active list.', 'Archived tasks retain all original details and history.']

## **Generated Test Cases:**

Here are five test cases for the Task Archiving feature:

**\*\*Test Case 1: Verify that completed tasks can be archived\*\***

Test Case ID: TA-001

Test Case Title: Verify that completed tasks can be archived

Test Case Description: This test case verifies that users can archive completed tasks, moving them to a separate section for easy access.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* A task is created and marked as completed
- \* The user is logged in to the system

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the task list page
3. Select a completed task and click on the "Archive" button
4. Verify that the task is moved to the archived tasks section
5. Verify that the task is no longer visible in the active task list

Postconditions:

- \* The task is removed from the active task list
- \* The task is visible in the archived tasks section

Expected Result: The completed task is successfully archived and moved to the separate section.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that archived tasks can be searched and viewed\*\***

Test Case ID: TA-002

Test Case Title: Verify that archived tasks can be searched and viewed

Test Case Description: This test case verifies that users can search and view archived tasks, retaining all original details and history.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A task is archived
- \* The user is logged in to the system

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the archived tasks section
3. Search for the archived task using various search criteria (e.g., task name, description, tags)
4. Verify that the archived task is displayed in the search results
5. View the archived task and verify that all original details and history are retained

Postconditions:

- \* The archived task is displayed in the search results
- \* All original details and history are retained

Expected Result: The archived task is successfully searched and viewed, retaining all original details and

history.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can restore archived tasks to the active list\*\***

Test Case ID: TA-003

Test Case Title: Verify that users can restore archived tasks to the active list

Test Case Description: This test case verifies that users can restore archived tasks to the active list, making them visible again.

Test Suite: Task Management

Test Priority: High

Preconditions:

\* A task is archived

\* The user is logged in to the system

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the archived tasks section
3. Select an archived task and click on the "Restore" button
4. Verify that the task is moved back to the active task list
5. Verify that the task is visible in the active task list

Postconditions:

- \* The task is removed from the archived tasks section
- \* The task is visible in the active task list

Expected Result: The archived task is successfully restored to the active list.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that archived tasks retain all original details and history\*\***

Test Case ID: TA-004

Test Case Title: Verify that archived tasks retain all original details and history

Test Case Description: This test case verifies that archived tasks retain all original details and history, including attachments and comments.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A task is archived
- \* The user is logged in to the system

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the archived tasks section
3. View an archived task and verify that all original details are retained (e.g., task name, description, due date, tags)
4. Verify that all original history is retained, including attachments and comments
5. Verify that the task's original status and priority are retained

Postconditions:

- \* All original details are retained
- \* All original history is retained

Expected Result: The archived task retains all original details and history.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that invalid archival attempts are handled correctly\*\***

Test Case ID: TA-005

Test Case Title: Verify that invalid archival attempts are handled correctly

Test Case Description: This test case verifies that the system handles invalid archival attempts correctly, preventing unintended changes to the task list.

Test Suite: Task Management

Test Priority: Low

Preconditions:

- \* A task is active
- \* The user is logged in to the system

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Navigate to the task list page
3. Attempt to archive an active task
4. Verify that an error message is displayed, indicating that the task cannot be archived
5. Verify that the task remains in the active task list

Postconditions:

- \* An error message is displayed
- \* The task remains in the active task list

Expected Result: The system handles invalid archival attempts correctly, preventing unintended changes to the task list.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Negative

-----



**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Guest Access

**Feature description -**

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

**Userstory title -**

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

**Acceptance criteria -**

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

**Generated Test Cases:**

Here are five detailed test cases for the Guest Access feature:

**\*\*Test Case 1: Verify that users can invite external collaborators as guests with limited access\*\***

Test Case ID: GUEST-001

Test Case Title: Verify that users can invite external collaborators as guests with limited access

Test Case Description: This test case ensures that users can successfully invite external collaborators as guests with limited access, and the invitee receives the invitation email.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- \* The user has a valid account and is logged in to the platform.
- \* The user has permission to grant guest access.
- \* The external collaborator has a valid email address.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a user with permission to grant guest access.
2. Navigate to the task or project page.
3. Click the "Invite Guest" button.
4. Enter the external collaborator's email address and select the "Guest" role.
5. Click "Send Invitation".
6. Verify that the invitation email is sent to the external collaborator's email address.

Postconditions:

- \* The external collaborator receives the invitation email.
- \* The user can view the guest invitation status on the task or project page.

Expected Result: The user can successfully invite external collaborators as guests with limited access.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that guests can view and collaborate on specific tasks or projects\*\***

Test Case ID: GUEST-002

Test Case Title: Verify that guests can view and collaborate on specific tasks or projects

Test Case Description: This test case ensures that guests can view and collaborate on specific tasks or projects after accepting the guest invitation.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- \* The user has invited an external collaborator as a guest with limited access.
- \* The guest has accepted the invitation and logged in to the platform.

Test Data: Task or project with guest access permission

Test Steps:

1. Log in to the platform as a guest.
2. Navigate to the task or project page.
3. Verify that the guest can view the task or project details.
4. Attempt to edit the task or project description.

5. Verify that the guest can comment on the task or project.

Postconditions:

- \* The guest can view the task or project details.
- \* The guest can collaborate on the task or project by commenting.

Expected Result: Guests can view and collaborate on specific tasks or projects after accepting the guest invitation.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system provides clear distinctions between guest and full access roles\*\***

Test Case ID: GUEST-003

Test Case Title: Verify that the system provides clear distinctions between guest and full access roles

Test Case Description: This test case ensures that the system clearly distinguishes between guest and full access roles, and guests cannot access sensitive or restricted areas of the platform.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- \* The user has granted guest access to an external collaborator.
- \* The guest has logged in to the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a guest.
2. Attempt to access a restricted area of the platform (e.g., settings, user management).
3. Verify that the guest is denied access and receives an error message.
4. Log in to the platform as a user with full access.
5. Verify that the user can access the restricted area.

Postconditions:

- \* The guest is denied access to restricted areas.
- \* The user with full access can access the restricted area.

Expected Result: The system provides clear distinctions between guest and full access roles.

Severity: Blocker

Priority: High

Type of Testing: Security Testing

Test Case Approach: Negative

**\*\*Test Case 4: Verify that users can manage and revoke guest access at any time\*\***

Test Case ID: GUEST-004

Test Case Title: Verify that users can manage and revoke guest access at any time

Test Case Description: This test case ensures that users can manage and revoke guest access at any time, and guests can no longer access the task or project.

Test Suite: Guest Access

Test Priority: Medium

Preconditions:

- \* The user has granted guest access to an external collaborator.
- \* The guest has logged in to the platform.

Test Data: Task or project with guest access permission

Test Steps:

1. Log in to the platform as a user with permission to grant guest access.
2. Navigate to the task or project page.
3. Revoke the guest access permission.
4. Verify that the guest can no longer access the task or project.
5. Log in to the platform as the guest.
6. Verify that the guest receives an error message when attempting to access the task or project.

Postconditions:

- \* The guest can no longer access the task or project.
- \* The user can view the revoked guest access status on the task or project page.

Expected Result: Users can manage and revoke guest access at any time, and guests can no longer access the task or project.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that guests cannot access sensitive or restricted areas of the platform\*\***

Test Case ID: GUEST-005

Test Case Title: Verify that guests cannot access sensitive or restricted areas of the platform

Test Case Description: This test case ensures that guests cannot access sensitive or restricted areas of the platform, such as user management or settings.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- \* The user has granted guest access to an external collaborator.

- \* The guest has logged in to the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a guest.
2. Attempt to access a sensitive or restricted area of the platform (e.g., user management, settings).
3. Verify that the guest is denied access and receives an error message.

Postconditions:

- \* The guest is denied access to sensitive or restricted areas.

- \* The guest receives an error message.

Expected Result: Guests cannot access sensitive or restricted areas of the platform.

Severity: Blocker

Priority: High

Type of Testing: Security Testing

Test Case Approach: Negative

-----

## **Data - 41**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Import/Export

### **Feature description -**

Users can import tasks from external sources or export tasks to other task management tools or file formats. Import/export capabilities facilitate data migration and integration with other systems, ensuring seamless collaboration and workflow management across platforms.

### **Userstory title -**

As a user, I want to import and export tasks so that I can migrate data between different task management tools.

### **Acceptance criteria -**

['Users can import tasks from external sources in various formats (e.g., CSV, Excel).', 'Users can export tasks to other task management tools or file formats.', 'The system provides clear mapping of fields during import/export.', 'Users can preview imported data before finalizing the process.', 'The system logs



import/export activities for future reference.']

## **Generated Test Cases:**

Here are five detailed test cases for the Task Import/Export feature of the Task Management Tool:

**\*\*Test Case 1: Verify that users can import tasks from external sources in CSV format\*\***

Test Case ID: TC-TIE-001

Test Case Title: Verify that users can import tasks from external sources in CSV format

Test Case Description: This test case verifies that users can import tasks from CSV files into the Task Management Tool.

Test Suite: Import/Export

Test Priority: High

Preconditions:

- \* The user has a valid CSV file with task data
- \* The Task Management Tool is logged in and ready for import
- \* The import feature is enabled

Test Data:

- \* Sample CSV file with task data (e.g., task name, description, deadline, priority)

Test Steps:

1. Log in to the Task Management Tool as a user
2. Click on the Import button and select CSV as the file type

3. Select the sample CSV file and click Import
4. Verify that the tasks are imported successfully and displayed in the task list
5. Verify that the task fields (e.g., name, description, deadline, priority) are accurately mapped from the CSV file

Postconditions:

- \* The imported tasks are added to the task list
- \* The task fields are updated accordingly

Expected Result: The tasks are successfully imported from the CSV file, and the task fields are accurately mapped.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can export tasks to other task management tools or file formats\*\***

Test Case ID: TC-TIE-002

Test Case Title: Verify that users can export tasks to other task management tools or file formats

Test Case Description: This test case verifies that users can export tasks from the Task Management Tool to other task management tools or file formats (e.g., Excel, Trello).

Test Suite: Import/Export

Test Priority: High

Preconditions:

- \* The user has created tasks in the Task Management Tool
- \* The export feature is enabled

Test Data:

- \* No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Select the tasks to be exported
3. Click on the Export button and select the desired file format (e.g., Excel, CSV)
4. Verify that the tasks are exported successfully and downloaded as the selected file type
5. Verify that the task fields (e.g., name, description, deadline, priority) are accurately exported

Postconditions:

- \* The exported tasks are downloaded as the selected file type
- \* The task fields are exported accordingly

Expected Result: The tasks are successfully exported to the selected file format, and the task fields are accurately exported.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the system provides clear mapping of fields during import/export\*\***

Test Case ID: TC-TIE-003

Test Case Title: Verify that the system provides clear mapping of fields during import/export

Test Case Description: This test case verifies that the system provides clear mapping of fields during the import and export process.

Test Suite: Import/Export

Test Priority: Medium

Preconditions:

- \* The user has a valid CSV file with task data
- \* The Task Management Tool is logged in and ready for import
- \* The import feature is enabled

Test Data:

- \* Sample CSV file with task data (e.g., task name, description, deadline, priority)

Test Steps:

1. Log in to the Task Management Tool as a user
2. Click on the Import button and select CSV as the file type
3. Select the sample CSV file and click Import
4. Verify that the system provides a clear mapping of fields (e.g., task name, description, deadline, priority) during the import process

5. Verify that the task fields are accurately mapped from the CSV file

Postconditions:

- \* The imported tasks are added to the task list
- \* The task fields are updated accordingly

Expected Result: The system provides clear mapping of fields during the import process, and the task fields are accurately mapped from the CSV file.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can preview imported data before finalizing the process\*\***

Test Case ID: TC-TIE-004

Test Case Title: Verify that users can preview imported data before finalizing the process

Test Case Description: This test case verifies that users can preview imported data before finalizing the import process.

Test Suite: Import/Export

Test Priority: Medium

Preconditions:

- \* The user has a valid CSV file with task data
- \* The Task Management Tool is logged in and ready for import

\* The import feature is enabled

Test Data:

\* Sample CSV file with task data (e.g., task name, description, deadline, priority)

Test Steps:

1. Log in to the Task Management Tool as a user
2. Click on the Import button and select CSV as the file type
3. Select the sample CSV file and click Import
4. Verify that the system provides a preview of the imported data before finalizing the process
5. Verify that the user can cancel the import process if necessary

Postconditions:

- \* The imported tasks are added to the task list
- \* The task fields are updated accordingly

Expected Result: The system provides a preview of the imported data before finalizing the process, allowing the user to review and cancel the import if necessary.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the system logs import/export activities for future reference\*\***

Test Case ID: TC-TIE-005

Test Case Title: Verify that the system logs import/export activities for future reference

Test Case Description: This test case verifies that the system logs import and export activities for future reference.

Test Suite: Import/Export

Test Priority: Low

Preconditions:

- \* The user has performed an import or export action
- \* The Task Management Tool is configured to log import/export activities

Test Data:

- \* No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user
2. Perform an import or export action
3. Verify that the system logs the import/export activity in the system log
4. Verify that the log entry includes relevant details (e.g., date, time, user, file type, number of tasks imported/exported)
5. Verify that the log entry is stored for future reference

Postconditions:

- \* The import/export activity is logged in the system log
- \* The log entry includes relevant details

Expected Result: The system logs import and export activities for future reference, including relevant details about the action.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **Data - 42**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Prioritization

### **Feature description -**

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.



### **Userstory title -**

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

### **Acceptance criteria -**

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.', 'Priority changes are logged and tracked.']

### **Generated Test Cases:**

Here are five detailed test cases for the Task Prioritization feature of the Task Management Tool:

**\*\*Test Case 1: Verify that tasks can be prioritized with different levels\*\***

Test Case ID: TP-001

Test Case Title: Verify that tasks can be prioritized with different levels

Test Case Description: This test case verifies that users can set and adjust task priorities with different levels (e.g., High, Medium, Low) and that the system updates the task list accordingly.

Test Suite: Task Management

Test Priority: High

Preconditions:

\* A user is logged in to the system

\* A task is created with a default priority level

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user

2. Create a new task with a default priority level
3. Update the task priority to High
4. Verify that the task is updated with the new priority level
5. Repeat steps 3-4 for Medium and Low priority levels

Postconditions:

- \* The task priority is updated in the system
- \* The task list is updated to reflect the new priority level

Expected Result: The system allows users to set and adjust task priorities with different levels, and updates the task list accordingly.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that tasks are sorted and filtered based on priority levels\*\***

Test Case ID: TP-002

Test Case Title: Verify that tasks are sorted and filtered based on priority levels

Test Case Description: This test case verifies that tasks can be sorted and filtered based on priority levels, allowing users to focus on high-priority tasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* A user is logged in to the system
- \* Multiple tasks are created with different priority levels

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Create multiple tasks with different priority levels (High, Medium, Low)
3. Sort tasks by priority level (High to Low)
4. Verify that tasks are sorted correctly
5. Filter tasks by priority level (e.g., show only High-priority tasks)
6. Verify that tasks are filtered correctly

Postconditions:

- \* Tasks are sorted and filtered based on priority levels
- \* The task list is updated to reflect the new sort and filter criteria

Expected Result: The system allows users to sort and filter tasks based on priority levels, facilitating task management and focus on high-priority tasks.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that visual indicators display task priority\*\***

Test Case ID: TP-003

Test Case Title: Verify that visual indicators display task priority

Test Case Description: This test case verifies that the system provides visual indicators of task priority, enabling users to quickly identify high-priority tasks.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A user is logged in to the system

- \* A task is created with a priority level

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Create a new task with a priority level (e.g., High)
3. Verify that a visual indicator (e.g., icon, color) is displayed to indicate the task priority
4. Repeat steps 2-3 for different priority levels (Medium, Low)

Postconditions:

- \* Visual indicators are displayed to indicate task priority

- \* The visual indicators are consistent across the system

Expected Result: The system provides visual indicators of task priority, enabling users to quickly identify

high-priority tasks.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that notifications are sent for high-priority tasks\*\***

Test Case ID: TP-004

Test Case Title: Verify that notifications are sent for high-priority tasks

Test Case Description: This test case verifies that the system sends notifications to users for high-priority tasks, ensuring they are aware of critical deadlines.

Test Suite: Task Management

Test Priority: High

Preconditions:

\* A user is logged in to the system

\* A high-priority task is created with a deadline

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Create a new task with a high priority level and a deadline
3. Verify that a notification is sent to the user (e.g., email, in-app notification)
4. Verify that the notification includes the task details and deadline

Postconditions:

- \* A notification is sent to the user for the high-priority task
- \* The notification includes the task details and deadline

Expected Result: The system sends notifications to users for high-priority tasks, ensuring they are aware of critical deadlines.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that priority changes are logged and tracked\*\***

Test Case ID: TP-005

Test Case Title: Verify that priority changes are logged and tracked

Test Case Description: This test case verifies that the system logs and tracks changes to task priority, enabling users to maintain an audit trail of task history.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* A user is logged in to the system
- \* A task is created with a priority level

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user
2. Create a new task with a priority level
3. Update the task priority level (e.g., from Medium to High)
4. Verify that the priority change is logged and tracked in the task history
5. Repeat steps 3-4 for multiple priority changes

Postconditions:

- \* Priority changes are logged and tracked in the task history
- \* The task history is updated to reflect the new priority level

Expected Result: The system logs and tracks changes to task priority, enabling users to maintain an audit trail of task history.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 43**

**Project name -**

Task Management Tool

## **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

## **Feature name -**

Task Collaboration

## **Feature description -**

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

## **Userstory title -**

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay updated on task progress.

## **Acceptance criteria -**

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

## **Generated Test Cases:**

Here are five detailed test cases to cover the acceptance criteria for the Task Collaboration feature:

**\*\*Test Case 1: Verify that users can assign tasks to team members\*\***

Test Case ID: TC001

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

Test Case Description: This test case verifies that users can successfully assign tasks to team members, ensuring effective collaboration and task delegation.



Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* User has created a task
- \* User has at least one team member in their contact list

Test Data: Task details (title, description, deadline), team member details (name, email)

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Create a new task with a title, description, and deadline.
3. Click on the "Assign to" button and select a team member from the contact list.
4. Verify that the team member is successfully assigned to the task.
5. Log out and log in as the assigned team member.

Postconditions:

- \* The task is updated to reflect the assigned team member.
- \* The assigned team member receives a notification about the task assignment.

Expected Result: The user can successfully assign a task to a team member, and the team member receives a notification.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can share updates and communicate through comments or messages\*\***

Test Case ID: TC002

Test Case Title: Verify that users can share updates and communicate through comments or messages

Test Case Description: This test case verifies that users can share updates and communicate effectively with team members through comments or messages, ensuring seamless collaboration.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* User has created a task and assigned it to a team member
- \* User has added a comment or message to the task

Test Data: Task details (title, description, deadline), comment or message text

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Create a new task and assign it to a team member.
3. Add a comment or message to the task.
4. Verify that the comment or message is visible to the assigned team member.
5. Log out and log in as the assigned team member.
6. Verify that the team member can view the comment or message.

Postconditions:

- \* The comment or message is updated in the task details.
- \* The assigned team member receives a notification about the comment or message.

Expected Result: The user can share updates and communicate with team members through comments or messages, and the team members receive notifications.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that task updates and comments are visible to assigned team members\*\***

Test Case ID: TC003

Test Case Title: Verify that task updates and comments are visible to assigned team members

Test Case Description: This test case verifies that assigned team members can view task updates and comments, ensuring transparency and collaboration.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool
- \* User has created a task and assigned it to a team member
- \* User has updated the task or added a comment

Test Data: Task details (title, description, deadline), comment or update text

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Create a new task and assign it to a team member.
3. Update the task or add a comment.
4. Verify that the update or comment is visible to the assigned team member.
5. Log out and log in as the assigned team member.
6. Verify that the team member can view the update or comment.

Postconditions:

- \* The task update or comment is reflected in the task details.
- \* The assigned team member receives a notification about the update or comment.

Expected Result: Assigned team members can view task updates and comments, ensuring transparency and collaboration.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users receive notifications for task updates and comments\*\***

Test Case ID: TC004

Test Case Title: Verify that users receive notifications for task updates and comments

Test Case Description: This test case verifies that users receive notifications for task updates and comments, ensuring timely awareness and collaboration.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* User has created a task and assigned it to a team member
- \* User has updated the task or added a comment

Test Data: Task details (title, description, deadline), comment or update text

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Create a new task and assign it to a team member.
3. Update the task or add a comment.
4. Verify that the assigned team member receives a notification about the update or comment.
5. Log out and log in as the assigned team member.
6. Verify that the team member can view the notification.

Postconditions:

- \* The notification is reflected in the task details or notification center.
- \* The assigned team member is aware of the task update or comment.

Expected Result: Users receive notifications for task updates and comments, ensuring timely awareness and collaboration.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the system provides a clear view of task collaboration history\*\***

Test Case ID: TC005

Test Case Title: Verify that the system provides a clear view of task collaboration history

Test Case Description: This test case verifies that the system provides a clear and organized view of task collaboration history, including task updates, comments, and assignee changes.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool
- \* User has created a task and assigned it to a team member
- \* User has updated the task or added comments multiple times

Test Data: Task details (title, description, deadline), comment or update text

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Create a new task and assign it to a team member.
3. Update the task or add comments multiple times.
4. Verify that the system provides a clear and organized view of task collaboration history.
5. Verify that the task collaboration history includes task updates, comments, and assignee changes.

Postconditions:

\* The task collaboration history is reflected in the task details or collaboration tab.

\* The task collaboration history is easily accessible and understandable.

Expected Result: The system provides a clear and organized view of task collaboration history, including task updates, comments, and assignee changes.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **Data - 44**

### **Project name -**

Task Management Tool

### **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

### **Feature name -**

Task Tracking

### **Feature description -**

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

### **Userstory title -**

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

### **Acceptance criteria -**

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.', 'System provides visual indicators for task progress.', '']

## Generated Test Cases:

Here are five detailed test cases for the Task Tracking feature of the Task Management Tool:

**\*\*Test Case 1: Verify that users can view task statuses\*\***

Test Case ID: TT-001

Test Case Title: Verify that users can view task statuses

Test Case Description: This test case ensures that users can view the current status of tasks, including completed, in-progress, and pending tasks.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Go to the Task List page.
3. Verify that each task in the list has a visible status indicator (e.g., completed, in-progress, pending).
4. Click on a task to view its details.
5. Verify that the task status is displayed on the task details page.



Postconditions:

- \* Task statuses are correctly updated in real-time.

Expected Result: Users can view the current status of tasks on the Task List page and task details page.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can track task progress and deadlines\*\***

Test Case ID: TT-002

Test Case Title: Verify that users can track task progress and deadlines

Test Case Description: This test case ensures that users can track the progress of tasks and view deadlines, including upcoming deadlines and overdue tasks.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool

- \* At least one task is created and assigned to the user with a deadline

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Go to the Task List page.
3. Filter tasks by upcoming deadlines.
4. Verify that tasks with upcoming deadlines are displayed on the Task List page.
5. Click on a task to view its details.
6. Verify that the task deadline is displayed on the task details page.

Postconditions:

\* Task deadlines are correctly updated in real-time.

Expected Result: Users can track task progress and view deadlines on the Task List page and task details page.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that tasks are updated in real-time\*\***

Test Case ID: TT-003

Test Case Title: Verify that tasks are updated in real-time

Test Case Description: This test case ensures that task updates, including changes to task status and progress, are reflected in real-time on the Task List page and task details page.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Go to the Task List page.
3. Update the status of a task to "in-progress".
4. Verify that the task status is updated in real-time on the Task List page.
5. Click on the task to view its details.
6. Verify that the task status is updated in real-time on the task details page.

Postconditions:

- \* Task updates are correctly reflected in real-time.

Expected Result: Task updates are reflected in real-time on the Task List page and task details page.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the system provides visual indicators for task progress\*\***

Test Case ID: TT-004

Test Case Title: Verify that the system provides visual indicators for task progress

Test Case Description: This test case ensures that the system provides visual indicators, such as color-coded status bars or progress meters, to indicate task progress.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Go to the Task List page.
3. Verify that each task has a visual indicator of its progress (e.g., color-coded status bar, progress meter).
4. Click on a task to view its details.
5. Verify that the visual indicator is displayed on the task details page.

Postconditions:

- \* Visual indicators are correctly updated in real-time.

Expected Result: The system provides visual indicators for task progress on the Task List page and task details page.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can view task deadlines and receive notifications\*\***

Test Case ID: TT-005

Test Case Title: Verify that users can view task deadlines and receive notifications

Test Case Description: This test case ensures that users can view task deadlines and receive notifications for upcoming deadlines and overdue tasks.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

\* User is logged in to the Task Management Tool

\* At least one task is created and assigned to the user with a deadline

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Go to the Task List page.
3. Filter tasks by upcoming deadlines.
4. Verify that tasks with upcoming deadlines are displayed on the Task List page.
5. Verify that notifications are sent to the user for upcoming deadlines and overdue tasks.

Postconditions:

\* Notifications are correctly sent to users for upcoming deadlines and overdue tasks.

Expected Result: Users can view task deadlines and receive notifications for upcoming deadlines and overdue tasks.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 45**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Notifications

**Feature description -**

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

## **Userstory title -**

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

## **Acceptance criteria -**

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.', 'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', '']

## **Generated Test Cases:**

Here are five detailed test cases for the Task Notifications feature:

**\*\*Test Case 1: Verify that users receive notifications for new tasks\*\***

Test Case ID: TN-001

Test Case Title: Verify that users receive notifications for new tasks

Test Case Description: This test case verifies that users receive notifications when a new task is created and assigned to them.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* User is logged in to the system
- \* User has notification settings enabled for new tasks

Test Data: No test data needed

Test Steps:

1. Create a new task and assign it to a user
2. Verify that the user receives a notification for the new task
3. Check the notification content to ensure it includes task details (e.g., task name, description, and deadline)

Postconditions:

\* Notification is marked as read after user views it

Expected Result: User receives a notification for the new task with correct task details

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users receive notifications for task updates and comments\*\***

Test Case ID: TN-002

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 is updated or a comment is added to a task they are assigned to.

Test Suite: Task Management

Test Priority: High

Preconditions:

\* User is logged in to the system

\* User has notification settings enabled for task updates and comments

Test Data: No test data needed

Test Steps:

1. Create a task and assign it to a user
2. Update the task description and add a comment
3. Verify that the user receives notifications for the task update and comment
4. Check the notification content to ensure it includes task details (e.g., task name, description, and deadline) and comment text

Postconditions:

\* Notification is marked as read after user views it



Expected Result: User receives notifications for task updates and comments with correct task details and comment text

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users receive reminders for upcoming deadlines\*\***

Test Case ID: TN-003

Test Case Title: Verify that users receive reminders for upcoming deadlines

Test Case Description: This test case verifies that users receive reminders for upcoming deadlines for tasks they are assigned to.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* User is logged in to the system
- \* User has notification settings enabled for deadline reminders

Test Data: No test data needed

Test Steps:

1. Create a task with an upcoming deadline and assign it to a user
2. Verify that the user receives a reminder notification for the upcoming deadline
3. Check the notification content to ensure it includes task details (e.g., task name, description, and deadline)

Postconditions:

- \* Notification is marked as read after user views it

Expected Result: User receives a reminder notification for the upcoming deadline with correct task details

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that notification settings can be customized by users\*\***

Test Case ID: TN-004

Test Case Title: Verify that notification settings can be customized by users

Test Case Description: This test case verifies that users can customize their notification settings for task-related events.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

\* User is logged in to the system

Test Data: No test data needed

Test Steps:

1. Access the notification settings page
2. Customize the notification settings for new tasks, task updates, and deadline reminders
3. Verify that the changes are saved and reflected in the user's notification preferences

Postconditions:

\* Notification settings are updated and saved

Expected Result: User can customize notification settings for task-related events

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users do not receive notifications for tasks they are not assigned to\*\***

Test Case ID: TN-005

Test Case Title: Verify that users do not receive notifications for tasks they are not assigned to

Test Case Description: This test case verifies that users do not receive notifications for tasks they are not assigned to.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* User is logged in to the system
- \* User has notification settings enabled for task-related events

Test Data: No test data needed

Test Steps:

1. Create a task and assign it to another user
2. Verify that the original user does not receive a notification for the task
3. Check the notification log to ensure no notification was sent to the original user

Postconditions:

- \* No notification is sent to the original user

Expected Result: User does not receive notifications for tasks they are not assigned to

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

-----

## **Project name -**

Task Management Tool

## **Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

## **Feature name -**

Task Categorization

## **Feature description -**

Users can categorize tasks into different groups or projects for better organization and management. Task categorization aims to help users organize tasks by type, priority, or project.

## **Userstory title -**

As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

## **Acceptance criteria -**

['Users can create and assign tasks to different categories or projects.', 'Users can view tasks grouped by categories or projects.', 'Tasks can be moved between categories or projects.', 'Category or project information is accurately reflected in task views.', '']

## **Generated Test Cases:**

Here are five detailed test cases for the Task Categorization feature:

**\*\*Test Case 1: Verify that users can create and assign tasks to different categories or projects\*\***

Test Case ID: TCT-001

Test Case Title: Verify that users can create and assign tasks to different categories or projects

Test Case Description: This test case verifies that users can create new categories or projects and assign tasks

to them.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* User is logged in to the system
- \* User has create task and category/project creation permissions
- \* No existing tasks or categories/projects are present in the system

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with create task and category/project creation permissions.
2. Click on the "Create Task" button.
3. Enter task details and select "Create New Category" or "Create New Project" option.
4. Enter category/project name and description.
5. Click on the "Create" button.
6. Verify that the task is created and assigned to the newly created category or project.

Postconditions:

- \* New category or project is created
- \* Task is assigned to the new category or project

Expected Result: The system allows users to create new categories or projects and assign tasks to them.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can view tasks grouped by categories or projects\*\***

Test Case ID: TCT-002

Test Case Title: Verify that users can view tasks grouped by categories or projects

Test Case Description: This test case verifies that users can view tasks organized by categories or projects.

Test Suite: Task Management

Test Priority: High

Preconditions:

- \* User is logged in to the system
- \* User has view task permissions
- \* Tasks are assigned to different categories or projects

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with view task permissions.
2. Click on the "Tasks" tab.
3. Select a category or project from the filter dropdown.
4. Verify that tasks are grouped by the selected category or project.
5. Repeat steps 2-4 for different categories or projects.

Postconditions:

- \* Tasks are displayed grouped by category or project

Expected Result: The system allows users to view tasks organized by categories or projects.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that tasks can be moved between categories or projects\*\***

Test Case ID: TCT-003

Test Case Title: Verify that tasks can be moved between categories or projects

Test Case Description: This test case verifies that users can move tasks between categories or projects.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* User is logged in to the system
- \* User has edit task permissions
- \* Tasks are assigned to different categories or projects

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with edit task permissions.
2. Click on the "Tasks" tab.
3. Select a task and click on the "Edit" button.
4. Change the category or project assignment and click on the "Save" button.
5. Verify that the task is moved to the new category or project.

Postconditions:

- \* Task is moved to the new category or project

Expected Result: The system allows users to move tasks between categories or projects.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that category or project information is accurately reflected in task views\*\***

Test Case ID: TCT-004

Test Case Title: Verify that category or project information is accurately reflected in task views

Test Case Description: This test case verifies that category or project information is correctly displayed in task views.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- \* User is logged in to the system

- \* User has view task permissions

- \* Tasks are assigned to different categories or projects

Test Data: No test data needed

Test Steps:



1. Log in to the system as a user with view task permissions.
2. Click on the "Tasks" tab.
3. Select a task and click on the "View" button.
4. Verify that the category or project information is correctly displayed.
5. Repeat steps 2-4 for different tasks and categories or projects.

Postconditions:

\* Category or project information is correctly displayed in task views

Expected Result: The system accurately reflects category or project information in task views.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users cannot create duplicate categories or projects\*\***

Test Case ID: TCT-005

Test Case Title: Verify that users cannot create duplicate categories or projects

Test Case Description: This test case verifies that users cannot create duplicate categories or projects.

Test Suite: Task Management

Test Priority: Low

Preconditions:

\* User is logged in to the system

\* User has create category/project creation permissions

\* Existing categories or projects are present in the system

Test Data: No test data needed

Test Steps:

1. Log in to the system as a user with create category/project creation permissions.
2. Click on the "Create Category" or "Create Project" button.
3. Enter an existing category or project name.
4. Click on the "Create" button.
5. Verify that an error message is displayed indicating that the category or project already exists.

Postconditions:

\* Error message is displayed

Expected Result: The system prevents users from creating duplicate categories or projects.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing, Acceptance Testing

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:**

Here are five detailed test cases for the Task Search feature:

**\*\*Test Case 1: Verify that users can search for tasks by name\*\***

Test Case ID: TC-001

Test Case Title: Verify that users can search for tasks by name

Test Case Description: This test case verifies that users can search for tasks by their name, and the search results display accurate and relevant tasks.

Test Suite: Task Search

Test Priority: High

Preconditions:

\* User is logged in to the Task Management Tool

\* At least one task is created with a unique name

Test Data: Task name to search for (e.g., "Meet with John")

Test Steps:

1. Log in to the Task Management Tool
2. Create a new task with a unique name (e.g., "Meet with John")
3. Navigate to the Task Search page
4. Enter the task name in the search field (e.g., "Meet with John")
5. Click the Search button

Postconditions:

\* Search results display the task with the matching name

\* Search results do not display tasks with different names

Expected Result: The task with the matching name is displayed in the search results.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can search for tasks by description\*\***

Test Case ID: TC-002

Test Case Title: Verify that users can search for tasks by description

Test Case Description: This test case verifies that users can search for tasks by their description, and the search results display accurate and relevant tasks.

Test Suite: Task Search

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least one task is created with a unique description

Test Data: Task description to search for (e.g., "Meeting with John to discuss project progress")

Test Steps:

1. Log in to the Task Management Tool
2. Create a new task with a unique description (e.g., "Meeting with John to discuss project progress")
3. Navigate to the Task Search page
4. Enter the task description in the search field (e.g., "Meeting with John to discuss project progress")
5. Click the Search button

Postconditions:

- \* Search results display the task with the matching description
- \* Search results do not display tasks with different descriptions

Expected Result: The task with the matching description is displayed in the search results.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can filter search results by category\*\***

Test Case ID: TC-003

Test Case Title: Verify that users can filter search results by category

Test Case Description: This test case verifies that users can filter search results by category, and the filtered

results display accurate and relevant tasks.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- \* User is logged in to the Task Management Tool

- \* At least two tasks are created with different categories (e.g., "Marketing" and "Sales")

Test Data: Category to filter by (e.g., "Marketing")

Test Steps:

1. Log in to the Task Management Tool
2. Create two new tasks with different categories (e.g., "Marketing" and "Sales")
3. Navigate to the Task Search page
4. Enter a search query (e.g., "Meet with John")
5. Select the category filter (e.g., "Marketing")
6. Click the Search button

Postconditions:

- \* Search results display only tasks with the selected category

- \* Search results do not display tasks with different categories

Expected Result: The filtered search results display only tasks with the selected category.

Severity: Medium

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that search results are accurate and relevant\*\***

Test Case ID: TC-004

Test Case Title: Verify that search results are accurate and relevant

Test Case Description: This test case verifies that search results are accurate and relevant, and do not display irrelevant tasks.

Test Suite: Task Search

Test Priority: High

Preconditions:

- \* User is logged in to the Task Management Tool
- \* At least three tasks are created with different names and descriptions

Test Data: Search query (e.g., "Meeting with John")

Test Steps:

1. Log in to the Task Management Tool
2. Create three new tasks with different names and descriptions
3. Navigate to the Task Search page
4. Enter the search query (e.g., "Meeting with John")
5. Click the Search button

Postconditions:

- \* Search results display only relevant tasks
- \* Search results do not display irrelevant tasks

Expected Result: The search results display accurate and relevant tasks.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that invalid search queries return no results\*\***

Test Case ID: TC-005

Test Case Title: Verify that invalid search queries return no results

Test Case Description: This test case verifies that invalid search queries (e.g., empty search field or special characters only) return no results.

Test Suite: Task Search

Test Priority: Low

Preconditions:

\* User is logged in to the Task Management Tool

Test Data: Invalid search query (e.g., empty search field or special characters only)

Test Steps:

1. Log in to the Task Management Tool
2. Navigate to the Task Search page
3. Enter an invalid search query (e.g., empty search field or special characters only)
4. Click the Search button

Postconditions:

\* No search results are displayed

\* An error message is displayed indicating that no results were found

Expected Result: No search results are displayed, and an error message is displayed.

Severity: Minor

Priority: Low



Type of Testing: Functional Testing

Test Case Approach: Negative

-----

**Data - 48**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Collaboration

**Feature description -**

Users can collaborate on tasks by adding comments, attaching files, and sharing tasks with other users. Task collaboration aims to enhance team communication and productivity.

**Userstory title -**

As a user, I want to collaborate on tasks by adding comments, attaching files, and sharing tasks with other users so that I can work more effectively with my team.

**Acceptance criteria -**

['Users can add comments to tasks.', 'Users can attach files to tasks.', 'Users can share tasks with other users.', 'Task collaboration features work seamlessly and enhance team communication.', '']

**Generated Test Cases:**

Here are five detailed test cases for the Task Collaboration feature:

**\*\*Test Case 1: Verify that users can add comments to tasks\*\***

Test Case ID: TC001

Test Case Title: Verify that users can add comments to tasks

Test Case Description: This test case verifies that users can add comments to tasks, allowing them to collaborate and communicate with team members.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- \* The user is logged in to the Task Management Tool.

- \* A task is created and assigned to the user.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Navigate to the task list and select a task.
3. Click on the "Add Comment" button.
4. Enter a comment in the comment box.
5. Click the "Post" button.

Postconditions:

- \* The comment is displayed below the task description.

- \* The task owner and team members receive a notification about the new comment.

Expected Result: The user can successfully add a comment to a task, and the comment is displayed below the task description.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can attach files to tasks\*\***

Test Case ID: TC002

Test Case Title: Verify that users can attach files to tasks

Test Case Description: This test case verifies that users can attach files to tasks, allowing them to share relevant documents and media with team members.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

\* The user is logged in to the Task Management Tool.

\* A task is created and assigned to the user.

Test Data: A sample file (e.g., PDF, DOCX)

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Navigate to the task list and select a task.
3. Click on the "Attach File" button.
4. Select the sample file to attach.

5. Click the "Upload" button.

Postconditions:

- \* The attached file is displayed below the task description.
- \* The task owner and team members receive a notification about the new attachment.

Expected Result: The user can successfully attach a file to a task, and the file is displayed below the task description.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can share tasks with other users\*\***

Test Case ID: TC003

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, allowing them to collaborate and work together on tasks.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- \* The user is logged in to the Task Management Tool.
- \* A task is created and assigned to the user.
- \* Another user is available to share the task with.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.
2. Navigate to the task list and select a task.
3. Click on the "Share Task" button.
4. Enter the email address or username of the other user.
5. Click the "Share" button.

Postconditions:

- \* The task is shared with the other user.
- \* The other user receives a notification about the shared task.

Expected Result: The user can successfully share a task with another user, and the task is displayed in the shared user's task list.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that task collaboration features work seamlessly\*\***

Test Case ID: TC004

Test Case Title: Verify that task collaboration features work seamlessly

Test Case Description: This test case verifies that task collaboration features, including commenting, file

attachment, and task sharing, work seamlessly and enhance team communication.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- \* The user is logged in to the Task Management Tool.

- \* A task is created and assigned to the user.

- \* Another user is available to collaborate on the task.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user.

2. Navigate to the task list and select a task.

3. Add a comment to the task.

4. Attach a file to the task.

5. Share the task with another user.

6. Log in as the other user and verify that the comment, file, and task sharing features are working correctly.

Postconditions:

- \* The task collaboration features are working seamlessly.

- \* The users can collaborate effectively on the task.

Expected Result: The task collaboration features work seamlessly, and users can collaborate effectively on tasks.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that task collaboration features handle multiple users and concurrent updates\*\***

Test Case ID: TC005

Test Case Title: Verify that task collaboration features handle multiple users and concurrent updates

Test Case Description: This test case verifies that task collaboration features can handle multiple users and concurrent updates without errors or data loss.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- \* The user is logged in to the Task Management Tool.
- \* A task is created and assigned to the user.
- \* Multiple users are available to collaborate on the task.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as multiple users.
2. Navigate to the task list and select the same task.
3. Each user adds a comment, attaches a file, and shares the task with other users concurrently.
4. Verify that the task collaboration features are working correctly and without errors.

Postconditions:

\* The task collaboration features can handle multiple users and concurrent updates.

\* No data is lost or corrupted during concurrent updates.

Expected Result: The task collaboration features can handle multiple users and concurrent updates without errors or data loss.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

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:**

Here are five detailed test cases for the Course Creation Tools feature:

**\*\*Test Case 1: Verify that instructors can create a course using various multimedia elements\*\***

Test Case ID: CECT-001

Test Case Title: Verify that instructors can create a course using videos, presentations, quizzes, and assignments

Test Case Description: This test case ensures that instructors can successfully create a course incorporating various multimedia elements, such as videos, presentations, quizzes, and assignments, to provide a comprehensive learning experience for students.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in
- \* Course creation tool is accessible
- \* Sample multimedia elements (video, presentation, quiz, assignment) are available

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and access the course creation tool
2. Create a new course and add a video lecture
3. Add a presentation and a quiz to the course
4. Add an assignment to the course
5. Save and preview the course

Postconditions:

- \* The course is created successfully
- \* All added multimedia elements are visible in the course preview

Expected Result: The instructor can create a course using videos, presentations, quizzes, and assignments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the course creation tools are user-friendly and require minimal technical skills\*\***

Test Case ID: CECT-002

Test Case Title: Verify that the course creation tools are intuitive and easy to use

Test Case Description: This test case ensures that the course creation tools are designed to be user-friendly, requiring minimal technical skills from instructors, allowing them to focus on creating engaging content.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in
- \* Course creation tool is accessible

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and access the course creation tool
2. Create a new course and add a video lecture
3. Observe the UI elements and instructional guidance provided
4. Attempt to add a quiz and an assignment to the course
5. Evaluate the overall user experience and ease of use

Postconditions:

- \* The instructor can create a course without requiring technical expertise
- \* The UI elements and instructional guidance are clear and helpful

Expected Result: The course creation tools are user-friendly and require minimal technical skills from instructors.

Severity: Major

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that courses can be saved as drafts\*\***

Test Case ID: CECT-003

Test Case Title: Verify that instructors can save courses as drafts

Test Case Description: This test case ensures that instructors can save their courses as drafts, allowing them to revisit and update their content at a later time.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in

- \* Course creation tool is accessible

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and access the course creation tool
2. Create a new course and add a video lecture
3. Click on the "Save as Draft" button
4. Verify that the course is saved as a draft
5. Log out and log back in to verify that the draft course is accessible

Postconditions:

- \* The course is saved as a draft

- \* The draft course is accessible from the instructor's dashboard

Expected Result: Instructors can save courses as drafts for later editing.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that courses can be edited post-publishing\*\***

Test Case ID: CECT-004

Test Case Title: Verify that instructors can edit courses after publishing

Test Case Description: This test case ensures that instructors can edit their published courses, allowing them to make updates and improvements based on student feedback or new content.

Test Suite: Course Creation

Test Priority: High

Preconditions:

\* Instructor account is created and logged in

\* Course creation tool is accessible

\* A published course exists

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and access the course creation tool
2. Select a published course and click on the "Edit" button
3. Make changes to the course content (e.g., update a video lecture)

4. Click on the "Save" button
5. Verify that the changes are reflected in the published course

Postconditions:

- \* The course is updated successfully
- \* The updated course is accessible to students

Expected Result: Instructors can edit published courses.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that courses can be previewed before publishing\*\***

Test Case ID: CECT-005

Test Case Title: Verify that instructors can preview courses before publishing

Test Case Description: This test case ensures that instructors can preview their courses before publishing, allowing them to review and refine their content before making it available to students.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in
- \* Course creation tool is accessible

\* A draft course exists

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and access the course creation tool
2. Select a draft course and click on the "Preview" button
3. Review the course content and verify that it is displayed as expected
4. Click on the "Back to Edit" button
5. Verify that the course is still in draft mode

Postconditions:

- \* The course is previewed successfully
- \* The course remains in draft mode after previewing

Expected Result: Instructors can preview courses before publishing.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 50**

**Project name -**

## Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Interactive Lessons

### **Feature description -**

Lessons are thoughtfully designed to foster learner engagement through interactive components such as quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these interactive elements enhance comprehension and retention of course material.

### **Userstory title -**

As a learner, I want interactive lessons so that I can engage with the content more effectively.

### **Acceptance criteria -**

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided after each interactive component.', 'Interactive lessons can be accessed on various devices.']

### **Generated Test Cases:**

Here are five detailed test cases for the Interactive Lessons feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that interactive lessons include quizzes\*\***

Test Case ID: IL001

Test Case Title: Verify that interactive lessons include quizzes

Test Case Description: This test case ensures that interactive lessons include quizzes as an interactive component.



Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- \* A course with an interactive lesson is available
- \* The learner is logged in to the platform
- \* The learner has access to the course

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Enroll in a course with an interactive lesson
3. Access the interactive lesson
4. Verify that a quiz is included in the lesson content
5. Attempt to answer the quiz questions

Postconditions:

- \* The quiz results are displayed
- \* The learner's progress is updated

Expected Result: The interactive lesson includes a quiz that can be completed by the learner.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that interactive elements are integrated seamlessly into the lesson content\*\***

Test Case ID: IL002

Test Case Title: Verify that interactive elements are integrated seamlessly into the lesson content

Test Case Description: This test case ensures that interactive elements, such as polls and simulations, are integrated seamlessly into the lesson content.

Test Suite: Interactive Lessons

Test Priority: Medium

Preconditions:

- \* A course with an interactive lesson is available
- \* The learner is logged in to the platform
- \* The learner has access to the course

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Enroll in a course with an interactive lesson
3. Access the interactive lesson
4. Verify that polls and simulations are integrated into the lesson content
5. Interact with the polls and simulations

Postconditions:

- \* The interactive elements are functional and responsive
- \* The learner's progress is updated

Expected Result: Interactive elements, such as polls and simulations, are integrated seamlessly into the lesson content.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that learners can track their progress through interactive activities\*\***

Test Case ID: IL003

Test Case Title: Verify that learners can track their progress through interactive activities

Test Case Description: This test case ensures that learners can track their progress through interactive activities, such as quizzes and discussions.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- \* A course with an interactive lesson is available
- \* The learner is logged in to the platform
- \* The learner has access to the course

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Enroll in a course with an interactive lesson
3. Access the interactive lesson
4. Complete interactive activities, such as quizzes and discussions
5. Verify that the learner's progress is updated and reflected in the platform

Postconditions:

- \* The learner's progress is updated in real-time
- \* The learner can view their progress history

Expected Result: Learners can track their progress through interactive activities.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that feedback is provided after each interactive component\*\***

Test Case ID: IL004

Test Case Title: Verify that feedback is provided after each interactive component

Test Case Description: This test case ensures that feedback is provided after each interactive component, such

as quizzes and simulations.

Test Suite: Interactive Lessons

Test Priority: Medium

Preconditions:

- \* A course with an interactive lesson is available
- \* The learner is logged in to the platform
- \* The learner has access to the course

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Enroll in a course with an interactive lesson
3. Access the interactive lesson
4. Complete an interactive component, such as a quiz or simulation
5. Verify that feedback is provided after completing the interactive component

Postconditions:

- \* The feedback is relevant and accurate
- \* The learner can view their feedback history

Expected Result: Feedback is provided after each interactive component.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that interactive lessons can be accessed on various devices\*\***

Test Case ID: IL005

Test Case Title: Verify that interactive lessons can be accessed on various devices

Test Case Description: This test case ensures that interactive lessons can be accessed on various devices, such as desktops, laptops, tablets, and mobile phones.

Test Suite: Interactive Lessons

Test Priority: Low

Preconditions:

- \* A course with an interactive lesson is available

- \* The learner has access to different devices (desktop, laptop, tablet, mobile phone)

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner on different devices (desktop, laptop, tablet, mobile phone)
2. Enroll in a course with an interactive lesson on each device
3. Access the interactive lesson on each device
4. Verify that the interactive lesson is accessible and functional on each device

Postconditions:

- \* The interactive lesson is accessible on all devices
- \* The learner's progress is synced across devices

Expected Result: Interactive lessons can be accessed on various devices.

Severity: Minor

Priority: Low

Type of Testing: Compatibility Testing

Test Case Approach: Positive

-----

## **Data - 51**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Personalized Learning Paths

### **Feature description -**

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

### **Userstory title -**

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

### **Acceptance criteria -**

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

### **Generated Test Cases:**

Here are five detailed test cases for the Personalized Learning Paths feature:

**\*\*Test Case 1: Verify that learners can choose courses and modules based on their interests\*\***

Test Case ID: PLP-001

Test Case Title: Verify that learners can choose courses and modules based on their interests

Test Case Description: This test case ensures that learners can select courses and modules that align with their interests and preferences.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

\* The learner has a registered account on the online learning platform.

\* The learner has specified their interests and preferences during the registration process.

Test Data: No test data needed

Test Steps:



1. Log in to the online learning platform as a learner.
2. Navigate to the "Personalized Learning Paths" section.
3. Click on "Explore Courses" and select a course that aligns with the learner's interests.
4. Click on "Add to Learning Path" to add the course to the learner's personalized learning path.
5. Verify that the course is successfully added to the learning path.

Postconditions:

- \* The course is added to the learner's personalized learning path.
- \* The learner can view the added course in their learning path.

Expected Result: The learner can successfully choose courses and modules based on their interests and add them to their personalized learning path.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that personalized learning paths adjust dynamically based on learner progress\*\***

Test Case ID: PLP-002

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress

Test Case Description: This test case ensures that the personalized learning path adapts to the learner's progress, recommending new courses and modules based on their performance.

Test Suite: Personalized Learning Paths

Test Priority: High

#### Preconditions:

- \* The learner has a registered account on the online learning platform.
- \* The learner has a personalized learning path with at least one course or module.

Test Data: No test data needed

#### Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the "Personalized Learning Paths" section.
3. Complete a module or course in the learning path.
4. Verify that the personalized learning path is updated with new course or module recommendations based on the learner's progress.
5. Click on a recommended course or module and verify that it is added to the learning path.

#### Postconditions:

- \* The personalized learning path is updated with new course or module recommendations.
- \* The learner can view the updated learning path with new course or module recommendations.

Expected Result: The personalized learning path adjusts dynamically based on the learner's progress, recommending new courses and modules.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that learners receive recommendations for new courses and modules\*\***

Test Case ID: PLP-003

Test Case Title: Verify that learners receive recommendations for new courses and modules

Test Case Description: This test case ensures that learners receive personalized recommendations for new courses and modules based on their interests, preferences, and learning objectives.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- \* The learner has a registered account on the online learning platform.
- \* The learner has specified their interests and preferences during the registration process.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the "Personalized Learning Paths" section.
3. Verify that the learner receives personalized recommendations for new courses and modules.
4. Click on a recommended course or module and verify that it is added to the learning path.

Postconditions:

- \* The learner receives personalized recommendations for new courses and modules.
- \* The learner can view the recommended courses and modules in their learning path.

Expected Result: The learner receives personalized recommendations for new courses and modules.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that progress tracking is personalized to each learner's path\*\***

Test Case ID: PLP-004

Test Case Title: Verify that progress tracking is personalized to each learner's path

Test Case Description: This test case ensures that progress tracking is tailored to each learner's personalized learning path, reflecting their unique learning objectives and goals.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

\* The learner has a registered account on the online learning platform.

\* The learner has a personalized learning path with at least one course or module.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the "Personalized Learning Paths" section.
3. Verify that the progress tracking reflects the learner's personalized learning path.
4. Complete a module or course in the learning path and verify that the progress tracking is updated.

Postconditions:

- \* The progress tracking reflects the learner's personalized learning path.
- \* The learner can view their progress tracking in the learning path.

Expected Result: Progress tracking is personalized to each learner's path, reflecting their unique learning objectives and goals.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that learners can adjust their learning paths at any time\*\***

Test Case ID: PLP-005

Test Case Title: Verify that learners can adjust their learning paths at any time

Test Case Description: This test case ensures that learners can modify their personalized learning paths at any time, adding or removing courses and modules as needed.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- \* The learner has a registered account on the online learning platform.
- \* The learner has a personalized learning path with at least one course or module.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the "Personalized Learning Paths" section.
3. Add a new course or module to the learning path.
4. Verify that the learning path is updated with the new addition.
5. Remove a course or module from the learning path and verify that it is removed.

Postconditions:

- \* The learner can modify their personalized learning path at any time.
- \* The learning path is updated with the learner's changes.

Expected Result: Learners can adjust their learning paths at any time, adding or removing courses and modules as needed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 52**

**Project name -**

Online Learning Platform

## **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

## **Feature name -**

Progress Tracking

## **Feature description -**

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

## **Userstory title -**

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

## **Acceptance criteria -**

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

## **Generated Test Cases:**

Here are five detailed test cases for the Progress Tracking feature:

**\*\*Test Case 1: Verify that completed activities are displayed in the progress tracker\*\***

Test Case ID: PT-001

Test Case Title: Verify that completed activities are displayed in the progress tracker

Test Case Description: This test case verifies that the progress tracker accurately displays all completed activities for a learner.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

\* Learner has completed at least one activity in a course

\* Learner is logged in to the online learning platform

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner
2. Navigate to the course dashboard
3. Click on the progress tracker section
4. Verify that all completed activities are displayed in the progress tracker

Postconditions: None

Expected Result: The progress tracker displays all completed activities for the learner

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that detailed reports on learner progress are available\*\***

Test Case ID: PT-002

Test Case Title: Verify that detailed reports on learner progress are available

Test Case Description: This test case verifies that learners can view detailed reports on their progress,



including completed activities, grades, and achievements.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

\* Learner has completed at least one activity in a course

\* Learner is logged in to the online learning platform

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner
2. Navigate to the course dashboard
3. Click on the progress tracker section
4. Click on the "View Detailed Report" button
5. Verify that the detailed report displays completed activities, grades, and achievements

Postconditions: None

Expected Result: The detailed report is available and displays accurate data

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that achievements are highlighted in the progress tracker\*\***

Test Case ID: PT-003

Test Case Title: Verify that achievements are highlighted in the progress tracker

Test Case Description: This test case verifies that achievements are highlighted in the progress tracker, making it easy for learners to identify their accomplishments.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

\* Learner has completed at least one activity in a course and achieved a milestone

\* Learner is logged in to the online learning platform

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner
2. Navigate to the course dashboard
3. Click on the progress tracker section
4. Verify that achievements are highlighted in the progress tracker

Postconditions: None

Expected Result: Achievements are highlighted in the progress tracker

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that learners receive notifications about their progress\*\***

Test Case ID: PT-004

Test Case Title: Verify that learners receive notifications about their progress

Test Case Description: This test case verifies that learners receive notifications about their progress, including completed activities, grades, and achievements.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

- \* Learner has completed at least one activity in a course
- \* Learner has enabled notifications in their profile settings

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner
2. Navigate to the course dashboard
3. Complete an activity
4. Verify that a notification is received about the completed activity
5. Verify that the notification includes details about the activity, grade, and achievement (if applicable)

Postconditions: None

Expected Result: Learner receives a notification about their progress

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that progress tracking can be exported as reports\*\***

Test Case ID: PT-005

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 tracking data as reports, allowing them to track their progress offline.

Test Suite: Progress Tracking

Test Priority: Low

Preconditions:

- \* Learner has completed at least one activity in a course

- \* Learner is logged in to the online learning platform

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner
2. Navigate to the course dashboard
3. Click on the progress tracker section
4. Click on the "Export Report" button
5. Verify that a report is generated and can be downloaded or printed

Postconditions: None

Expected Result: Progress tracking data can be exported as a report

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

## **Data - 53**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Assessment and Grading

### **Feature description -**

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

### **Userstory title -**

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

### **Acceptance criteria -**

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

### **Generated Test Cases:**

Here are five detailed test cases for the Assessment and Grading feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that instructors can create diverse assessments\*\***

Test Case ID: ATG-001

Test Case Title: Verify that instructors can create diverse assessments

Test Case Description: This test case ensures that instructors can create different types of assessments, such as quizzes, tests, and assignments, to evaluate learner comprehension.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in
- \* Course is created and available for instructor to access

Test Data: No test data needed

Test Steps:

1. Log in to the instructor account and navigate to the course page
2. Click on the "Create Assessment" button
3. Select the assessment type (Quiz, Test, or Assignment) from the dropdown menu
4. Fill in the assessment details, such as title, description, and duration
5. Click "Save" to create the assessment

Postconditions:

- \* Assessment is created and visible in the course page

\* Instructor can view and edit the assessment

Expected Result: Instructors can create diverse assessments, including quizzes, tests, and assignments.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that assessments can include various question types\*\***

Test Case ID: ATG-002

Test Case Title: Verify that assessments can include various question types

Test Case Description: This test case ensures that instructors can create assessments with different question types, such as multiple-choice questions, short answers, and essays.

Test Suite: Assessment and Grading

Test Priority: Medium

Preconditions:

\* Instructor account is created and logged in

\* Assessment is created and available for instructor to access

Test Data: No test data needed

Test Steps:

1. Log in to the instructor account and navigate to the assessment page
2. Click on the "Add Question" button
3. Select the question type (Multiple Choice, Short Answer, Essay, etc.) from the dropdown menu

4. Fill in the question details, such as question text, options, and points

5. Click "Save" to add the question to the assessment

Postconditions:

- \* Question is added to the assessment

- \* Instructor can view and edit the question

Expected Result: Assessments can include various question types, such as multiple-choice questions, short answers, and essays.

Severity: Critical

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that automatic grading is available for objective questions\*\***

Test Case ID: ATG-003

Test Case Title: Verify that automatic grading is available for objective questions

Test Case Description: This test case ensures that the system can automatically grade objective questions, such as multiple-choice questions, and provide instant feedback to learners.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in

- \* Assessment with objective questions is created and available for instructor to access



Test Data: No test data needed

Test Steps:

1. Log in to the instructor account and navigate to the assessment page
2. Click on the "Grade" button
3. Select the objective question type from the dropdown menu
4. Enter the correct answer for the question
5. Click "Save" to grade the question

Postconditions:

- \* Question is graded and feedback is provided to learners
- \* Instructor can view the grading results

Expected Result: Automatic grading is available for objective questions, and learners receive instant feedback.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that manual grading is available for subjective questions\*\***

Test Case ID: ATG-004

Test Case Title: Verify that manual grading is available for subjective questions

Test Case Description: This test case ensures that the system allows instructors to manually grade subjective questions, such as essays, and provide personalized feedback to learners.

Test Suite: Assessment and Grading

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in

- \* Assessment with subjective questions is created and available for instructor to access

Test Data: No test data needed

Test Steps:

1. Log in to the instructor account and navigate to the assessment page
2. Click on the "Grade" button
3. Select the subjective question type from the dropdown menu
4. Enter the grading criteria and feedback for the question
5. Click "Save" to grade the question

Postconditions:

- \* Question is graded and feedback is provided to learners

- \* Instructor can view the grading results

Expected Result: Manual grading is available for subjective questions, and learners receive personalized feedback.

Severity: Critical

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that feedback is provided to learners after grading\*\***

Test Case ID: ATG-005

Test Case Title: Verify that feedback is provided to learners after grading

Test Case Description: This test case ensures that the system provides feedback to learners after grading, including correct answers, explanations, and instructor comments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- \* Learner account is created and logged in
- \* Assessment is created and available for learner to take
- \* Instructor has graded the assessment

Test Data: No test data needed

Test Steps:

1. Log in to the learner account and navigate to the assessment page
2. Click on the "View Results" button
3. Verify that feedback is provided, including correct answers, explanations, and instructor comments

Postconditions:

- \* Feedback is provided to learners
- \* Learners can view the grading results

Expected Result: Feedback is provided to learners after grading, including correct answers, explanations, and instructor comments.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **Data - 54**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Discussion Forums

### **Feature description -**

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge sharing, enriching the learning experience through diverse perspectives and active participation.

### **Userstory title -**

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

### **Acceptance criteria -**

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all

devices.']

## **Generated Test Cases:**

Here are five detailed test cases for the Discussion Forums feature, ensuring thorough coverage of the acceptance criteria:

### **\*\*Test Case 1: Post Questions and Responses\*\***

Test Case ID: DF-001

Test Case Title: Verify that learners can post questions and responses in forums.

Test Case Description: This test case ensures that learners can create new posts and respond to existing ones in the discussion forums.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

\* A learner is logged in to the platform.

\* A course with an active discussion forum is available.

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access a course with an active discussion forum.
2. Click on the "New Post" button to create a new topic.
3. Enter a title and message body for the post.
4. Click the "Post" button to submit the new topic.
5. Verify that the post is visible in the forum.

6. Respond to an existing post by clicking the "Reply" button.
7. Enter a response message and click the "Post" button.
8. Verify that the response is visible as a threaded discussion.

Postconditions:

- \* The new post is visible in the forum.
- \* The response is visible as a threaded discussion.

Expected Result: Learners can successfully post questions and responses in the discussion forums.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Threaded Discussions\*\***

Test Case ID: DF-002

Test Case Title: Verify that forums support threaded discussions.

Test Case Description: This test case ensures that learners can engage in threaded discussions, and responses are displayed hierarchically.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- \* A learner is logged in to the platform.
- \* A course with an active discussion forum is available.

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access a course with an active discussion forum.
2. Create a new post and submit it.
3. Respond to the post by clicking the "Reply" button.
4. Enter a response message and click the "Post" button.
5. Verify that the response is displayed as a threaded discussion below the original post.
6. Respond to the response by clicking the "Reply" button.
7. Enter another response message and click the "Post" button.
8. Verify that the new response is displayed as a nested threaded discussion.

Postconditions:

- \* The threaded discussion is displayed correctly.
- \* Responses are displayed hierarchically.

Expected Result: Forums support threaded discussions, and responses are displayed correctly.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Instructor Moderation\*\***

Test Case ID: DF-003

Test Case Title: Verify that instructors can moderate discussions.

Test Case Description: This test case ensures that instructors can manage discussions by deleting or editing posts.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- \* An instructor is logged in to the platform.
- \* A course with an active discussion forum is available.

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and access a course with an active discussion forum.
2. Find a post that requires moderation.
3. Click the "Edit" button to edit the post.
4. Make changes to the post and click the "Update" button.
5. Verify that the changes are reflected in the forum.
6. Delete a post by clicking the "Delete" button.
7. Verify that the post is removed from the forum.

Postconditions:

- \* The post is updated successfully.
- \* The post is deleted successfully.

Expected Result: Instructors can successfully moderate discussions by editing or deleting posts.



Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Notification for New Posts\*\***

Test Case ID: DF-004

Test Case Title: Verify that learners receive notifications for new posts.

Test Case Description: This test case ensures that learners receive notifications when new posts are made in a forum they are subscribed to.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- \* A learner is logged in to the platform.
- \* A course with an active discussion forum is available.
- \* The learner is subscribed to the forum.

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access a course with an active discussion forum.
2. Subscribe to the forum.
3. Log out and have another user create a new post in the forum.
4. Log back in as the learner and verify that a notification is received for the new post.

Postconditions:

- \* A notification is received for the new post.

Expected Result: Learners receive notifications for new posts in forums they are subscribed to.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Accessibility on All Devices\*\***

Test Case ID: DF-005

Test Case Title: Verify that forums are accessible on all devices.

Test Case Description: This test case ensures that the discussion forums are accessible and functional on various devices, including desktops, laptops, tablets, and mobile phones.

Test Suite: Discussion Forums

Test Priority: Low

Preconditions:

- \* A learner is logged in to the platform.

- \* A course with an active discussion forum is available.

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access a course with an active discussion forum on a desktop device.

2. Verify that the forum is accessible and functional.
3. Repeat steps 1-2 on a laptop, tablet, and mobile phone.

Postconditions:

\* The forum is accessible and functional on all devices.

Expected Result: The discussion forums are accessible and functional on all devices.

Severity: Trivial

Priority: Low

Type of Testing: Compatibility Testing

Test Case Approach: Positive

-----

**Data - 55**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Live Classes and Webinars

**Feature description -**

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A

opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

**Userstory title -**

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

**Acceptance criteria -**

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

**Generated Test Cases:**

Here are five detailed test cases for the "Live Classes and Webinars" feature:

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

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

**\*\*Test Case Title:\*\*** Verify that live classes and webinars can be conducted through video conferencing tools.

**\*\*Test Case Description:\*\*** This test case ensures that the platform allows instructors to conduct live classes and webinars through seamless video conferencing tools, enabling dynamic engagement and real-time interaction between instructors and learners.

**\*\*Test Suite:\*\*** Live Classes and Webinars

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

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

\* Instructor account is created and logged in.

\* Learner account is created and logged in.

\* Video conferencing tools are integrated with the platform.

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

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

1. Log in as an instructor and create a new live class or webinar session.
2. Invite learners to join the session.
3. Start the session and verify that video conferencing tools are enabled.
4. Verify that learners can join the session and participate in real-time.
5. Verify that instructors can share their screen and resources during the session.

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

- \* Session is recorded and available for later viewing.
- \* Learners receive notifications before the session starts.

**\*\*Expected Result:\*\*** Live classes and webinars can be conducted successfully through video conferencing tools, enabling real-time interaction between instructors and learners.

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

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

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that learners can ask questions in real-time during live classes and webinars.

**\*\*Test Case Description:\*\*** This test case ensures that learners can ask questions in real-time during live classes and webinars, enabling dynamic engagement and interaction with instructors.

**\*\*Test Suite:\*\*** Live Classes and Webinars

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

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

\* Learner account is created and logged in.

\* Live class or webinar session is created and started by the instructor.

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

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

1. Join a live class or webinar session as a learner.
2. Ask a question in real-time using the chat or Q&A feature.
3. Verify that the instructor receives the question and can respond in real-time.
4. Verify that other learners can view the question and response.

**\*\*Postconditions:\*\* None.**

**\*\*Expected Result:\*\* Learners can ask questions in real-time during live classes and webinars, and instructors can respond accordingly.**

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

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

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

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

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

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

**\*\*Test Case Title:\*\* Verify that sessions are recorded and available for later viewing.**

**\*\*Test Case Description:\*\* This test case ensures that live classes and webinars are recorded and available for**

later viewing, allowing learners to review the content at their convenience.

**\*\*Test Suite:\*\*** Live Classes and Webinars

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

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

\* Instructor account is created and logged in.

\* Live class or webinar session is created and started by the instructor.

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

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

1. Start a live class or webinar session as an instructor.
2. Verify that the session is being recorded.
3. End the session and verify that the recording is available for later viewing.
4. Log in as a learner and verify that the recording can be accessed and played back.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** Live classes and webinars are recorded and available for later viewing, allowing learners to review the content at their convenience.

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

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

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

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

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

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

**\*\*Test Case Title:\*\*** Verify that instructors can share screen and resources during sessions.

**\*\*Test Case Description:\*\*** This test case ensures that instructors can share their screen and resources during live classes and webinars, enhancing the learning experience and enabling dynamic engagement.

**\*\*Test Suite:\*\*** Live Classes and Webinars

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

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

\* Instructor account is created and logged in.

\* Live class or webinar session is created and started by the instructor.

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

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

1. Start a live class or webinar session as an instructor.
2. Share the screen and resources (e.g., PowerPoint presentation) with learners.
3. Verify that learners can view the shared screen and resources in real-time.
4. Verify that instructors can control the shared screen and resources.

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** Instructors can share their screen and resources during live classes and webinars, enhancing the learning experience and enabling dynamic engagement.

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

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

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

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



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

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

**\*\*Test Case Title:\*\*** Verify that notifications are sent before sessions start.

**\*\*Test Case Description:\*\*** This test case ensures that learners receive notifications before live classes and webinars start, reminding them of upcoming sessions and enabling them to prepare accordingly.

**\*\*Test Suite:\*\*** Live Classes and Webinars

**\*\*Test Priority:\*\*** Low

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

\* Learner account is created and logged in.

\* Live class or webinar session is created and scheduled by the instructor.

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

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

1. Schedule a live class or webinar session with a start time in the future.
2. Verify that learners receive notifications (e.g., email, in-app notification) before the session starts.
3. Verify that the notification includes the session details (e.g., date, time, topic).

**\*\*Postconditions:\*\*** None.

**\*\*Expected Result:\*\*** Learners receive notifications before live classes and webinars start, reminding them of upcoming sessions and enabling them to prepare accordingly.

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

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

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

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

---

## **Data - 56**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Mobile Accessibility

### **Feature description -**

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

### **Userstory title -**

As a learner, I want mobile access to the platform so that I can learn on the go.

### **Acceptance criteria -**

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

### **Generated Test Cases:**

Here are five detailed test cases for the Mobile Accessibility feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that the platform is accessible on smartphones\*\***

Test Case ID: MOB-001

Test Case Title: Verify that the platform is accessible on smartphones

Test Case Description: This test case verifies that learners can access the online learning platform using their smartphones.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- \* A valid learner account with a username and password
- \* A compatible smartphone with a stable internet connection

Test Data: No test data needed

Test Steps:

1. Open the online learning platform on a smartphone using a compatible browser.
2. Enter valid login credentials and click the "Login" button.
3. Verify that the dashboard is displayed correctly and accessible on the smartphone screen.

Postconditions:

- \* Log out of the platform
- \* Ensure the smartphone is disconnected from the internet

Expected Result: The online learning platform is accessible and functional on a smartphone.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the mobile UI is optimized for touch navigation\*\***

Test Case ID: MOB-002

Test Case Title: Verify that the mobile UI is optimized for touch navigation

Test Case Description: This test case verifies that the mobile UI is optimized for touch navigation, ensuring a seamless learning experience.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- \* A valid learner account with a username and password
- \* A compatible smartphone with a stable internet connection

Test Data: No test data needed

Test Steps:

1. Open the online learning platform on a smartphone using a compatible browser.
2. Navigate to a course page using touch gestures (e.g., swipe, tap).
3. Verify that the UI elements are responsive and adapt to the screen size.
4. Verify that the course materials are accessible and readable using touch gestures.

Postconditions:

- \* Log out of the platform

\* Ensure the smartphone is disconnected from the internet

Expected Result: The mobile UI is optimized for touch navigation, and learners can access course materials easily.

Severity: Critical

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that course materials are accessible and readable on mobile devices\*\***

Test Case ID: MOB-003

Test Case Title: Verify that course materials are accessible and readable on mobile devices

Test Case Description: This test case verifies that course materials are accessible and readable on mobile devices, ensuring that learners can access educational resources on-the-go.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

\* A valid learner account with a username and password

\* A compatible smartphone or tablet with a stable internet connection

\* A course with downloadable resources (e.g., PDF, video)

Test Data: No test data needed

Test Steps:

1. Open the online learning platform on a mobile device using a compatible browser.

2. Navigate to a course page with downloadable resources.
3. Verify that the resources are accessible and can be downloaded or viewed on the mobile device.
4. Verify that the resources are readable and functional on the mobile device.

Postconditions:

- \* Log out of the platform
- \* Ensure the mobile device is disconnected from the internet

Expected Result: Course materials are accessible and readable on mobile devices, allowing learners to access educational resources on-the-go.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that notifications are sent to mobile devices\*\***

Test Case ID: MOB-004

Test Case Title: Verify that notifications are sent to mobile devices

Test Case Description: This test case verifies that notifications are sent to mobile devices, ensuring that learners receive timely updates and reminders.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- \* A valid learner account with a username and password

\* A compatible smartphone with a stable internet connection

\* Notification settings enabled on the platform

Test Data: No test data needed

Test Steps:

1. Open the online learning platform on a smartphone using a compatible browser.
2. Navigate to a course page with notification settings enabled.
3. Verify that notifications are sent to the mobile device (e.g., push notification, in-app notification).
4. Verify that the notifications are readable and functional on the mobile device.

Postconditions:

\* Log out of the platform

\* Ensure the smartphone is disconnected from the internet

Expected Result: Notifications are sent to mobile devices, allowing learners to stay informed and engaged with the platform.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the platform supports offline access to downloaded materials\*\***

Test Case ID: MOB-005

Test Case Title: Verify that the platform supports offline access to downloaded materials

Test Case Description: This test case verifies that the platform supports offline access to downloaded materials, ensuring that learners can access educational resources even without an internet connection.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- \* A valid learner account with a username and password
- \* A compatible smartphone or tablet with a stable internet connection
- \* A course with downloadable resources (e.g., PDF, video)

Test Data: No test data needed

Test Steps:

1. Open the online learning platform on a mobile device using a compatible browser.
2. Navigate to a course page with downloadable resources.
3. Download the resources to the mobile device.
4. Disconnect the mobile device from the internet.
5. Verify that the downloaded resources are accessible and functional offline.

Postconditions:

- \* Reconnect the mobile device to the internet
- \* Log out of the platform

Expected Result: The platform supports offline access to downloaded materials, allowing learners to access educational resources even without an internet connection.

Severity: Blocker



Priority: High

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:

Here are five detailed test cases for the Resource Library feature, ensuring comprehensive coverage of the acceptance criteria:

**\*\*Test Case 1: Verify that the Resource Library includes e-books, articles, and reference guides.\*\***

Test Case ID: RL-001

Test Case Title: Verify that the Resource Library includes e-books, articles, and reference guides.

Test Case Description: This test case ensures that the Resource Library includes various types of supplementary materials, such as e-books, articles, and reference guides, providing learners with a comprehensive repository of learning resources.

Test Suite: Resource Library

Test Priority: High

Preconditions:

\* The Resource Library feature is enabled and accessible.

\* The learner is logged in to the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the Resource Library page.
3. Verify that the library includes e-books, articles, and reference guides.
4. Verify that each resource type has a sufficient number of items.

Postconditions:

- \* The learner can access the Resource Library.
- \* The resources are properly categorized and tagged.

Expected Result: The Resource Library includes e-books, articles, and reference guides, providing learners with a comprehensive repository of learning resources.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that learners can search for specific materials in the Resource Library.\*\***

Test Case ID: RL-002

Test Case Title: Verify that learners can search for specific materials in the Resource Library.

Test Case Description: This test case ensures that learners can search for specific resources within the Resource Library, facilitating easy access to required materials.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- \* The Resource Library feature is enabled and accessible.
- \* The learner is logged in to the platform.

Test Data: Sample search queries (e.g., "Python programming", "Data Science")

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the Resource Library page.
3. Enter a search query in the search bar.
4. Verify that the search results display relevant resources.
5. Verify that the search results include filters and sorting options.

Postconditions:

- \* The search results are updated accordingly.
- \* The learner can access the searched resources.

Expected Result: Learners can search for specific materials in the Resource Library, and relevant resources are displayed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that resources in the Resource Library are categorized and tagged for easy navigation.\*\***

Test Case ID: RL-003

Test Case Title: Verify that resources in the Resource Library are categorized and tagged for easy navigation.

Test Case Description: This test case ensures that resources in the Resource Library are properly categorized and tagged, enabling learners to easily navigate and find relevant materials.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- \* The Resource Library feature is enabled and accessible.
- \* The learner is logged in to the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the Resource Library page.
3. Verify that resources are categorized by type (e.g., e-books, articles, reference guides).
4. Verify that each resource has relevant tags and keywords.
5. Verify that the categorization and tagging facilitate easy navigation and filtering.

Postconditions:

- \* The learner can access the categorized and tagged resources.
- \* The learner can filter and sort resources using categories and tags.

Expected Result: Resources in the Resource Library are categorized and tagged, enabling learners to easily navigate and find relevant materials.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that learners can download resources for offline access.\*\***

Test Case ID: RL-004

Test Case Title: Verify that learners can download resources for offline access.

Test Case Description: This test case ensures that learners can download resources from the Resource Library, allowing them to access materials even when offline.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- \* The Resource Library feature is enabled and accessible.
- \* The learner is logged in to the platform.

Test Data: Sample resources (e.g., e-book, article)

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the Resource Library page.
3. Select a resource to download.
4. Verify that the download option is available.
5. Verify that the resource is successfully downloaded.

Postconditions:

- \* The learner can access the downloaded resource offline.
- \* The downloaded resource is in the correct format (e.g., PDF, EPUB).

Expected Result: Learners can download resources from the Resource Library, enabling offline access to

materials.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that instructors can recommend resources to learners.\*\***

Test Case ID: RL-005

Test Case Title: Verify that instructors can recommend resources to learners.

Test Case Description: This test case ensures that instructors can recommend resources to learners, providing personalized learning recommendations and enhancing the learning experience.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- \* The Resource Library feature is enabled and accessible.
- \* The instructor is logged in to the platform.
- \* The learner is enrolled in a course with the instructor.

Test Data: Sample resource recommendation

Test Steps:

1. Log in to the platform as an instructor.
2. Navigate to the course page.
3. Select a learner to recommend resources to.
4. Verify that the "Recommend Resource" option is available.

5. Verify that the recommended resource is successfully sent to the learner.

Postconditions:

\* The learner receives the recommended resource notification.

\* The recommended resource is added to the learner's resource library.

Expected Result: Instructors can recommend resources to learners, providing personalized learning recommendations and enhancing the learning experience.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 58**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Peer Review

**Feature description -**

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's



work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

### **Userstory title -**

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

### **Acceptance criteria -**

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers remain anonymous to ensure unbiased feedback.']

### **Generated Test Cases:**

Here are five test cases that cover the acceptance criteria for the Peer Review feature:

#### **\*\*Test Case 1: Submitting Assignments for Peer Review\*\***

Test Case ID: PR001

Test Case Title: Verify that learners can submit assignments for peer review.

Test Case Description: This test case ensures that learners can successfully submit their assignments for peer review.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- \* Learner is logged in to the platform
- \* Learner has an active course enrollment
- \* Assignment is available for submission

Test Data: Sample assignment file (e.g., PDF or Word document)

#### Test Steps:

1. Log in to the platform as a learner
2. Navigate to the course page and select the assignment
3. Click the "Submit for Peer Review" button
4. Upload the sample assignment file
5. Click "Submit"

#### Postconditions:

- \* Assignment is listed under "Submitted for Peer Review"
- \* Peer review status is updated to "Assignable"

Expected Result: Learner is able to submit their assignment for peer review successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Providing Feedback on Submitted Assignments\*\***

Test Case ID: PR002

Test Case Title: Verify that peers can provide feedback on submitted assignments.

Test Case Description: This test case ensures that peers can provide feedback on submitted assignments.

Test Suite: Peer Review

Test Priority: High

#### Preconditions:

- \* Learner has submitted an assignment for peer review

- \* Peer reviewer is logged in to the platform
- \* Peer reviewer has been assigned to review the submission

Test Data: Sample feedback text

Test Steps:

1. Log in to the platform as a peer reviewer
2. Navigate to the peer review dashboard
3. Select the assignment to review
4. Enter sample feedback text in the feedback field
5. Click "Submit Feedback"

Postconditions:

- \* Feedback is listed under "Feedback Received"
- \* Peer review status is updated to "Feedback Provided"

Expected Result: Peer reviewer is able to provide feedback on the submitted assignment successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Structured Review Process\*\***

Test Case ID: PR003

Test Case Title: Verify that a structured review process guides the feedback.

Test Case Description: This test case ensures that the platform provides a structured review process to guide peer feedback.

Test Suite: Peer Review

Test Priority: Medium

Preconditions:

- \* Learner has submitted an assignment for peer review
- \* Peer reviewer is logged in to the platform
- \* Peer reviewer has been assigned to review the submission

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a peer reviewer
2. Navigate to the peer review dashboard
3. Select the assignment to review
4. Verify that the review form includes questions or prompts to guide feedback
5. Verify that the review form includes a rating scale or scoring system

Postconditions:

- \* Peer review status remains "Assignable" until feedback is submitted

Expected Result: The platform provides a structured review process to guide peer feedback.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Viewing Feedback on Submitted Assignments\*\***

Test Case ID: PR004

Test Case Title: Verify that learners can view feedback on their work.

Test Case Description: This test case ensures that learners can view feedback received on their submitted assignments.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- \* Learner has submitted an assignment for peer review
- \* Peer reviewer has provided feedback on the submission

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Navigate to the course page and select the assignment
3. Click the "View Feedback" button
4. Verify that feedback is displayed, including reviewer comments and ratings

Postconditions:

- \* Feedback is listed under "Feedback Received"

Expected Result: Learner is able to view feedback on their submitted assignment successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

## **\*\*Test Case 5: Anonymous Reviewers\*\***

Test Case ID: PR005

Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback.

Test Case Description: This test case ensures that the platform maintains reviewer anonymity to ensure unbiased feedback.

Test Suite: Peer Review

Test Priority: Medium

Preconditions:

- \* Learner has submitted an assignment for peer review
- \* Peer reviewer has provided feedback on the submission

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Navigate to the course page and select the assignment
3. Click the "View Feedback" button
4. Verify that reviewer information (e.g., name, username, profile picture) is not displayed

Postconditions:

- \* Reviewer anonymity is maintained

Expected Result: Reviewer remains anonymous to ensure unbiased feedback.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

## **Data - 59**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Certificates and Badges

### **Feature description -**

Upon successful completion of courses or specific milestones, learners receive certificates and badges to recognize their achievements and accomplishments. These credentials provide tangible recognition of learners' efforts, skills, and expertise, enhancing their credentials and demonstrating their commitment to continuous learning and professional development.

### **Userstory title -**

As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

### **Acceptance criteria -**

['Learners earn certificates upon course completion.', 'Badges are awarded for specific achievements and milestones.', 'Certificates and badges are accessible through learner profiles.', 'Learners can share their certificates and badges on social media.', 'Instructors can issue custom badges for special achievements.']

### **Generated Test Cases:**

Here are five detailed test cases for the "Certificates and Badges" feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that learners earn certificates upon course completion\*\***

Test Case ID: CT001

Test Case Title: Verify that learners earn certificates upon course completion

Test Case Description: This test case verifies that learners receive certificates upon completing a course, which can be accessed through their profiles.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- \* A course with at least one module and one quiz is available.
- \* A learner has enrolled in the course and completed all modules and quizzes.

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access the course dashboard.
2. Complete all course modules and quizzes.
3. Verify that the course completion status is updated to "Completed".
4. Access the learner profile and navigate to the Certificates section.
5. Verify that a certificate for the completed course is available for download.

Postconditions:

- \* The learner profile is updated with the earned certificate.
- \* The certificate is downloadable in a suitable format (e.g., PDF).



Expected Result: The learner earns a certificate upon completing a course, which is accessible through their profile.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that badges are awarded for specific achievements and milestones\*\***

Test Case ID: CT002

Test Case Title: Verify that badges are awarded for specific achievements and milestones

Test Case Description: This test case verifies that learners receive badges for achieving specific milestones or completing certain tasks, which can be accessed through their profiles.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

\* A course with at least one module and one quiz is available.

\* A learner has enrolled in the course and completed a specific task or milestone (e.g., completed a certain number of quizzes).

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access the course dashboard.
2. Complete the specific task or milestone (e.g., complete 5 quizzes).

3. Verify that the badge is awarded and displayed on the learner profile.
4. Access the learner profile and navigate to the Badges section.
5. Verify that the earned badge is displayed with a description and date earned.

Postconditions:

- \* The learner profile is updated with the earned badge.
- \* The badge is displayed with a description and date earned.

Expected Result: The learner earns a badge for achieving a specific milestone or completing a task, which is accessible through their profile.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that certificates and badges are accessible through learner profiles\*\***

Test Case ID: CT003

Test Case Title: Verify that certificates and badges are accessible through learner profiles

Test Case Description: This test case verifies that learners can access their earned certificates and badges through their profiles.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

- \* A learner has earned at least one certificate and one badge.

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access the learner profile.
2. Navigate to the Certificates section and verify that all earned certificates are listed.
3. Navigate to the Badges section and verify that all earned badges are listed.
4. Verify that each certificate and badge has a description, date earned, and a download option (for certificates).

Postconditions:

- \* The learner profile is updated with the earned certificates and badges.
- \* All certificates and badges are accessible through the learner profile.

Expected Result: Learners can access their earned certificates and badges through their profiles.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that learners can share their certificates and badges on social media\*\***

Test Case ID: CT004

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.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

\* A learner has earned at least one certificate and one badge.

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access the learner profile.
2. Navigate to the Certificates section and select a certificate to share.
3. Verify that the "Share" button is available and functional.
4. Select a social media platform (e.g., LinkedIn, Twitter, Facebook) and verify that the certificate is shared successfully.
5. Repeat steps 2-4 for a badge.

Postconditions:

- \* The certificate or badge is shared on the selected social media platform.
- \* The shareable content includes a link to the learner profile or a downloadable link for the certificate.

Expected Result: Learners can share their earned certificates and badges on social media platforms.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that instructors can issue custom badges for special achievements\*\***

Test Case ID: CT005

Test Case Title: Verify that instructors can issue custom badges for special achievements

Test Case Description: This test case verifies that instructors can create and award custom badges for learners who achieve specific milestones or complete special tasks.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

\* An instructor has created a course with at least one module and one quiz.

\* A learner has enrolled in the course and completed a specific task or milestone (e.g., completed a project).

Test Data: No test data needed

Test Steps:

1. Log in as an instructor and access the course dashboard.
2. Create a custom badge for the specific task or milestone.
3. Award the custom badge to the learner who completed the task or milestone.
4. Log in as the learner and access the learner profile.
5. Verify that the custom badge is displayed in the Badges section.

Postconditions:

\* The custom badge is created and available for awarding.

\* The learner profile is updated with the earned custom badge.

Expected Result: Instructors can create and award custom badges for learners who achieve specific milestones or complete special tasks.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

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:**

Here are five test cases for the Gamification Elements feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that learners earn points for completing activities\*\***

Test Case ID: Gamification-001

Test Case Title: Verify that learners earn points for completing activities

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- \* Learner account created and logged in
- \* A course with activities (e.g., quizzes, videos, discussions) available
- \* No points earned previously

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access a course with available activities
2. Complete an activity (e.g., quiz, video, discussion) within the course
3. Verify that points are displayed on the learner's profile or gamification dashboard
4. Verify that the points earned are correctly calculated based on the activity type and difficulty

Postconditions:

- \* Verify that the points are updated in real-time
- \* Verify that the points are correctly reflected on the leaderboard

Expected Result: Learner earns points for completing activities, and points are displayed correctly on the profile and leaderboard.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that levels are achieved based on accumulated points\*\***

Test Case ID: Gamification-002

Test Case Title: Verify that levels are achieved based on accumulated points

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- \* Learner account created and logged in
- \* Points earned previously through completing activities
- \* Level thresholds defined in the system (e.g., 100 points for Level 1, 500 points for Level 2)

Test Data: No test data needed

Test Steps:



1. Log in as a learner and access the gamification dashboard
2. Verify that the current level is displayed based on the accumulated points
3. Earn additional points by completing more activities
4. Verify that the level is updated accordingly based on the new accumulated points

Postconditions:

- \* Verify that the level update is reflected on the leaderboard
- \* Verify that the level is correctly displayed on the learner's profile

Expected Result: Learner achieves levels based on accumulated points, and levels are updated correctly on the profile and leaderboard.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that leaderboards display top-performing learners\*\***

Test Case ID: Gamification-003

Test Case Title: Verify that leaderboards display top-performing learners

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- \* Multiple learner accounts created and logged in

\* Points earned by learners through completing activities

\* Leaderboard feature enabled

Test Data: No test data needed

Test Steps:

1. Log in as an administrator and access the leaderboard feature
2. Verify that the leaderboard displays the top-performing learners based on their accumulated points
3. Verify that the leaderboard is updated in real-time as learners earn new points
4. Verify that the leaderboard displays the correct ranking and points for each learner

Postconditions:

\* Verify that the leaderboard is accessible to learners and displays their ranking

\* Verify that the leaderboard is correctly sorted by points earned

Expected Result: Leaderboard displays top-performing learners based on their accumulated points, and is updated in real-time.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that badges are awarded for reaching milestones\*\***

Test Case ID: Gamification-004

Test Case Title: Verify that badges are awarded for reaching milestones

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- \* Learner account created and logged in
- \* Milestones defined in the system (e.g., completing a certain number of courses)
- \* Badges created and configured for milestones

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access the gamification dashboard
2. Verify that the badges are displayed on the dashboard with corresponding milestones
3. Reach a milestone (e.g., complete a course)
4. Verify that the corresponding badge is awarded and displayed on the dashboard

Postconditions:

- \* Verify that the badge is correctly displayed on the learner's profile
- \* Verify that the badge is correctly reflected on the leaderboard

Expected Result: Learner earns badges for reaching milestones, and badges are displayed correctly on the profile and leaderboard.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that learners can view their progress in a gamification dashboard\*\***

Test Case ID: Gamification-005

Test Case Title: Verify that learners can view their progress in a gamification dashboard

Test Suite: Gamification Elements

Test Priority: Low

Preconditions:

- \* Learner account created and logged in
- \* Gamification dashboard feature enabled

Test Data: No test data needed

Test Steps:

1. Log in as a learner and access the gamification dashboard
2. Verify that the dashboard displays the learner's progress (e.g., points, levels, badges)
3. Verify that the dashboard allows learners to view their progress over time (e.g., via a graph or chart)
4. Verify that the dashboard provides clear and concise information about the learner's progress

Postconditions:

- \* Verify that the dashboard is accessible and user-friendly
- \* Verify that the dashboard is correctly updated in real-time

Expected Result: Learner can view their progress in a gamification dashboard, and the dashboard provides clear and concise information about their progress.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 61**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Multilingual Support

**Feature description -**

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

**Userstory title -**

As a learner, I want multilingual support so that I can access the platform in my preferred language.

**Acceptance criteria -**

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated help and support resources.', 'Instructors can provide content in multiple languages.']

### **Generated Test Cases:**

Here are five detailed test cases for the Multilingual Support feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that the platform interface is available in multiple languages\*\***

Test Case ID: ML-001

Test Case Title: Verify that the platform interface is available in multiple languages

Test Case Description: The test case verifies that the online learning platform is accessible in multiple languages, allowing learners to select their preferred language for the interface.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- \* The platform is configured to support multiple languages.
- \* A user account is created with a default language set.

Test Data: No test data needed

Test Steps:

1. Log in to the platform with a user account.
2. Click on the language selection dropdown menu.
3. Select a language other than the default language (e.g., Spanish).
4. Verify that the platform interface is updated to the selected language.

Postconditions:

- \* The language selection dropdown menu is updated to reflect the new language.
- \* Any changes made to the language preference are saved for future sessions.

Expected Result: The platform interface is available in multiple languages, and learners can select their preferred language.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that course materials are available in different languages\*\***

Test Case ID: ML-002

Test Case Title: Verify that course materials are available in different languages

Test Case Description: The test case verifies that course materials, including video lectures, downloadable resources, and quizzes, are available in multiple languages.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- \* A course is created with materials in multiple languages.
- \* A user account is created with a language preference set.

Test Data:

\* Course materials in multiple languages (e.g., English, French, Mandarin).

Test Steps:

1. Log in to the platform with a user account.
2. Enroll in a course with materials available in multiple languages.
3. Verify that the course materials, including video lectures, downloadable resources, and quizzes, are available in the language preferred by the user.
4. Switch the language preference to a different language (e.g., from English to French).
5. Verify that the course materials are updated to the new language.

Postconditions:

\* The course materials are updated to reflect the new language preference.

Expected Result: Course materials are available in multiple languages, and learners can access them in their preferred language.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

\*\*Test Case 3: Verify that learners can select their preferred language at any time\*\*



Test Case ID: ML-003

Test Case Title: Verify that learners can select their preferred language at any time

Test Case Description: The test case verifies that learners can change their language preference at any point during their learning journey.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

\* A user account is created with a default language set.

Test Data: No test data needed

Test Steps:

1. Log in to the platform with a user account.
2. Click on the profile settings icon.
3. Select the language preference option.
4. Choose a different language (e.g., from English to Spanish).
5. Verify that the language preference is updated.

Postconditions:

\* The language preference is updated, and the platform interface reflects the new language.

\* Any changes made to the language preference are saved for future sessions.

Expected Result: Learners can select their preferred language at any time, and the platform updates accordingly.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that multilingual support includes translated help and support resources\*\***

Test Case ID: ML-004

Test Case Title: Verify that multilingual support includes translated help and support resources

Test Case Description: The test case verifies that help and support resources, including FAQs, tutorials, and contact information, are available in multiple languages.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

\* Help and support resources are created in multiple languages.

Test Data:

\* Help and support resources in multiple languages (e.g., English, French, Mandarin).

Test Steps:

1. Log in to the platform with a user account.
2. Click on the help and support icon.
3. Verify that the help and support resources, including FAQs, tutorials, and contact information, are available in the language preferred by the user.

4. Switch the language preference to a different language (e.g., from English to French).
5. Verify that the help and support resources are updated to the new language.

Postconditions:

\* The help and support resources are updated to reflect the new language preference.

Expected Result: Multilingual support includes translated help and support resources, ensuring that learners can access assistance in their preferred language.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that instructors can provide content in multiple languages\*\***

Test Case ID: ML-005

Test Case Title: Verify that instructors can provide content in multiple languages

Test Case Description: The test case verifies that instructors can create and upload course materials in multiple languages.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

\* An instructor account is created with multiple language options.

\* A course is created with materials in multiple languages.

Test Data:

\* Course materials in multiple languages (e.g., English, French, Mandarin).

Test Steps:

1. Log in to the platform with an instructor account.
2. Create a new course or edit an existing one.
3. Upload course materials in multiple languages.
4. Verify that the course materials are displayed in the correct language based on the learner's language preference.

Postconditions:

\* The course materials are updated to reflect the multiple languages.

Expected Result: Instructors can provide content in multiple languages, ensuring that learners can access course materials in their preferred language.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Accessibility Features

**Feature description -**

The platform prioritizes accessibility by incorporating features such as screen reader compatibility, adjustable font sizes, and color contrast options. These accessibility features ensure that learners with disabilities can access and navigate the platform with ease, promoting inclusivity and equitable access to educational opportunities.

**Userstory title -**

As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

**Acceptance criteria -**

['The platform is compatible with screen readers.', 'Learners can adjust font sizes for better readability.', 'Color contrast options are available for improved visibility.', 'Accessibility features are easy to activate and customize.', 'Support resources are available to assist learners with disabilities.']

**Generated Test Cases:**

Here are five detailed test cases for the Accessibility Features of the Online Learning Platform:

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

Test Case ID: ATC-001

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

Test Case Description: This test case ensures that the platform's content is accessible to learners who use

screen readers.

Test Suite: Accessibility

Test Priority: High

Preconditions:

- \* Screen reader software is installed and configured on the test device.

- \* A course with audio and visual content is available on the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform with a learner account.
2. Navigate to a course with audio and visual content.
3. Enable the screen reader software.
4. Verify that the screen reader reads out the course content, including audio descriptions and visual elements.
5. Verify that the learner can navigate through the course using the screen reader.

Postconditions:

- \* The screen reader software is disabled.

- \* The learner account is logged out.

Expected Result: The platform is compatible with screen readers, and learners can access course content using assistive technology.

Severity: Blocker

Priority: High

Type of Testing: Accessibility Testing

Test Case Approach: Positive

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

Test Case ID: ATC-002

Test Case Title: Verify that learners can adjust font sizes for better readability.

Test Case Description: This test case ensures that learners can customize font sizes to improve readability.

Test Suite: Accessibility

Test Priority: Medium

Preconditions:

\* A course with text-based content is available on the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform with a learner account.
2. Navigate to a course with text-based content.
3. Click on the font size adjustment feature.
4. Verify that the font size can be increased and decreased using the adjustment feature.
5. Verify that the font size change is applied to all text-based content on the platform.

Postconditions:

\* The learner account is logged out.

Expected Result: Learners can adjust font sizes for better readability, and the adjustment is applied platform-wide.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: ATC-003

Test Case Title: Verify that color contrast options are available for improved visibility.

Test Case Description: This test case ensures that learners can customize color contrast options to improve visibility.

Test Suite: Accessibility

Test Priority: Medium

Preconditions:

\* A course with visual content is available on the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform with a learner account.
2. Navigate to a course with visual content.
3. Click on the color contrast option feature.
4. Verify that different color contrast options are available (e.g., high contrast, low contrast, inverted colors).
5. Verify that the selected color contrast option is applied to all visual content on the platform.

Postconditions:



\* The learner account is logged out.

Expected Result: Learners can adjust color contrast options for improved visibility, and the adjustment is applied platform-wide.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: ATC-004

Test Case Title: Verify that accessibility features are easy to activate and customize.

Test Case Description: This test case ensures that learners can easily activate and customize accessibility features.

Test Suite: Accessibility

Test Priority: High

Preconditions:

\* A course with accessibility features is available on the platform.

Test Data: No test data needed

Test Steps:

1. Log in to the platform with a learner account.
2. Navigate to the accessibility feature settings.
3. Verify that the accessibility features can be easily activated and customized.
4. Verify that the customizations are saved and applied to the learner's account.

5. Verify that the learner can access a tutorial or support resources for using accessibility features.

Postconditions:

\* The learner account is logged out.

Expected Result: Learners can easily activate and customize accessibility features, and the customizations are saved and applied to their account.

Severity: Blocker

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

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

Test Case ID: ATC-005

Test Case Title: Verify that support resources are available to assist learners with disabilities.

Test Case Description: This test case ensures that learners with disabilities have access to support resources.

Test Suite: Accessibility

Test Priority: High

Preconditions:

\* A course with accessibility features is available on the platform.

Test Data: No test data needed

Test Steps:

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

2. Navigate to the support resources section.
3. Verify that support resources for learners with disabilities are available (e.g., tutorials, FAQs, contact information).
4. Verify that the support resources are accessible and usable for learners with disabilities.
5. Verify that the support resources provide clear instructions for using accessibility features.

Postconditions:

\* The learner account is logged out.

Expected Result: Learners with disabilities have access to support resources that are accessible and usable, and provide clear instructions for using accessibility features.

Severity: Blocker

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

-----

**Data - 63**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

## Discussion Moderation

### **Feature description -**

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

### **Userstory title -**

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

### **Acceptance criteria -**

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

### **Generated Test Cases:**

Here are five detailed test cases for the Discussion Moderation feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that instructors can edit posts in discussions\*\***

Test Case ID: DM-001

Test Case Title: Verify that instructors can edit posts in discussions

Test Case Description: This test case verifies that instructors can edit posts in discussions to maintain a positive and respectful learning environment.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in
- \* A discussion forum is created for a course
- \* A post is created in the discussion forum

Test Data:

- \* Post content to be edited
- \* Edited post content

Test Steps:

1. Log in as an instructor and navigate to the discussion forum.
2. Create a new post or select an existing post to edit.
3. Click the "Edit" button next to the post.
4. Make changes to the post content and click "Save".
5. Verify that the edited post content is updated in the discussion forum.

Postconditions:

- \* The edited post is updated in the discussion forum
- \* Other participants can view the updated post

Expected Result: The instructor can successfully edit a post in the discussion forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that instructors can mute or ban disruptive users\*\***

Test Case ID: DM-002

Test Case Title: Verify that instructors can mute or ban disruptive users

Test Case Description: This test case verifies that instructors can mute or ban users who engage in disruptive behavior to maintain a positive and respectful learning environment.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in
- \* A discussion forum is created for a course
- \* A user account is created and is participating in the discussion forum

Test Data:

- \* User account details
- \* Disruptive post content

Test Steps:

1. Log in as an instructor and navigate to the discussion forum.
2. Identify a user who has posted disruptive content.
3. Click on the user's name and select "Mute" or "Ban" from the dropdown menu.
4. Verify that the user is muted or banned from the discussion forum.
5. Verify that the user receives a notification of the moderation action.

Postconditions:

- \* The user is muted or banned from the discussion forum
- \* Other participants are not affected by the moderation action

Expected Result: The instructor can successfully mute or ban a disruptive user from the discussion forum.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that moderation actions are logged and reported\*\***

Test Case ID: DM-003

Test Case Title: Verify that moderation actions are logged and reported

Test Case Description: This test case verifies that all moderation actions taken by instructors are logged and reported to maintain transparency and accountability.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in
- \* A discussion forum is created for a course
- \* A moderation action is taken (e.g., editing, deleting, highlighting a post)

Test Data:

\* Moderation action details

\* Report format requirements

Test Steps:

1. Log in as an instructor and navigate to the discussion forum.
2. Take a moderation action (e.g., edit a post).
3. Navigate to the moderation action log or report.
4. Verify that the moderation action is logged and reported with details (e.g., date, time, action taken).
5. Verify that the report can be exported or printed.

Postconditions:

\* The moderation action is logged and reported

\* The report is accessible to instructors and administrators

Expected Result: All moderation actions taken by instructors are logged and reported.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that community guidelines are visible to all participants\*\***

Test Case ID: DM-004

Test Case Title: Verify that community guidelines are visible to all participants



Test Case Description: This test case verifies that community guidelines are clearly visible to all participants in the discussion forum to promote respectful and constructive interactions.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- \* A discussion forum is created for a course
- \* Community guidelines are created and configured

Test Data:

- \* Community guidelines content

Test Steps:

1. Navigate to the discussion forum.
2. Verify that the community guidelines are displayed prominently at the top of the discussion forum.
3. Verify that the community guidelines are easily accessible from the discussion forum.
4. Verify that the community guidelines are up-to-date and reflect the latest changes.

Postconditions:

- \* The community guidelines are visible to all participants
- \* The community guidelines are easily accessible

Expected Result: Community guidelines are clearly visible to all participants in the discussion forum.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that notifications are sent for moderated actions\*\***

Test Case ID: DM-005

Test Case Title: Verify that notifications are sent for moderated actions

Test Case Description: This test case verifies that notifications are sent to affected users when a moderation action is taken to keep them informed of changes to their posts or accounts.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in
- \* A discussion forum is created for a course
- \* A moderation action is taken (e.g., editing, deleting, highlighting a post)
- \* The affected user has notifications enabled

Test Data:

- \* Notification content
- \* Affected user account details

Test Steps:

1. Log in as an instructor and navigate to the discussion forum.

2. Take a moderation action (e.g., edit a post).
3. Verify that a notification is sent to the affected user.
4. Verify that the notification contains relevant details (e.g., moderation action, post content).
5. Verify that the notification is received by the affected user.

Postconditions:

- \* The notification is sent to the affected user
- \* The notification contains relevant details

Expected Result: Notifications are sent to affected users when a moderation action is taken.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 64**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Learning Analytics

## **Feature description -**

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement, performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning experience to meet learners' needs effectively.

## **Userstory title -**

As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

## **Acceptance criteria -**

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.', 'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.', 'Instructors can export analytics reports for further analysis.']

## **Generated Test Cases:**

Here are five detailed test cases to ensure that the Learning Analytics feature meets the acceptance criteria:

**\*\*Test Case 1: Verify that engagement metrics are displayed on the learning analytics dashboard\*\***

Test Case ID: LA-001

Test Case Description: This test case verifies that the learning analytics dashboard displays engagement metrics, allowing instructors to gauge learner participation.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

\* Instructor account created and logged in

\* At least one course with learner interactions available

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the Learning Analytics dashboard.
3. Select a course with learner interactions.
4. Verify that engagement metrics (e.g., time spent on course, pages viewed, interactions) are displayed on the dashboard.

Postconditions:

- \* No error messages are displayed.
- \* The dashboard refreshes without issues.

Expected Result: Engagement metrics are accurately displayed on the learning analytics dashboard.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that performance trends are visualized over time on the learning analytics dashboard\*\***

Test Case ID: LA-002

Test Case Description: This test case ensures that performance trends are visualized over time, enabling instructors to identify areas of improvement.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- \* Instructor account created and logged in
- \* At least one course with learner interactions available

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the Learning Analytics dashboard.
3. Select a course with learner interactions.
4. Verify that performance trends (e.g., quiz scores, assignment submissions) are visualized over time on the dashboard.

Postconditions:

- \* No error messages are displayed.
- \* The dashboard refreshes without issues.

Expected Result: Performance trends are accurately visualized over time on the learning analytics dashboard.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that instructors can access detailed reports on learner progress\*\***

Test Case ID: LA-003

Test Case Description: This test case verifies that instructors can access detailed reports on learner progress, providing insights into individual learner performance.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

- \* Instructor account created and logged in
- \* At least one course with learner interactions available

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the Learning Analytics dashboard.
3. Select a course with learner interactions.
4. Click on the "Learner Progress" report.
5. Verify that detailed reports on learner progress (e.g., quiz scores, assignment submissions) are displayed.

Postconditions:

- \* No error messages are displayed.
- \* The report is generated without issues.

Expected Result: Instructors can access detailed reports on learner progress.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that analytics data is updated in real-time on the learning analytics dashboard\*\***

Test Case ID: LA-004

Test Case Description: This test case ensures that analytics data is updated in real-time, allowing instructors to track learner interactions and progress in real-time.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- \* Instructor account created and logged in
- \* At least one course with learner interactions available

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the Learning Analytics dashboard.
3. Select a course with learner interactions.
4. Perform an action as a learner (e.g., submit an assignment, take a quiz).
5. Verify that the analytics data is updated in real-time on the dashboard.

Postconditions:



\* No error messages are displayed.

\* The dashboard refreshes without issues.

Expected Result: Analytics data is updated in real-time on the learning analytics dashboard.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that instructors can export analytics reports for further analysis\*\***

Test Case ID: LA-005

Test Case Description: This test case verifies that instructors can export analytics reports, enabling them to analyze learner data offline.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

\* Instructor account created and logged in

\* At least one course with learner interactions available

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor.
2. Navigate to the Learning Analytics dashboard.
3. Select a course with learner interactions.

4. Click on the "Export Report" button.
5. Verify that the analytics report is exported successfully (e.g., CSV, PDF).

Postconditions:

- \* No error messages are displayed.
- \* The report is exported without issues.

Expected Result: Instructors can export analytics reports for further analysis.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 65**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Social Learning Features

**Feature description -**

The platform integrates social learning features such as group projects, peer collaboration tools, and social

sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

**Userstory title -**

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

**Acceptance criteria -**

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

**Generated Test Cases:**

Here are five detailed test cases for the Social Learning Features of the Online Learning Platform:

**\*\*Test Case 1: Verify that learners can create and join groups for projects\*\***

Test Case ID: SLF-001

Test Case Title: Verify that learners can create and join groups for projects

Test Case Description: This test case validates that learners can successfully create and join groups for projects, enabling collaboration and social learning.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- \* Learner is logged in to the platform
- \* Course is available with group project assignment
- \* Learner has not created or joined a group before

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Navigate to the course page with a group project assignment
3. Click on "Create Group" button
4. Fill in group details (name, description, etc.)
5. Click "Create Group"
6. Verify that the group is created successfully
7. Join the created group
8. Verify that the learner is successfully added to the group

Postconditions:

- \* Group is created and visible in the course page
- \* Learner is listed as a member of the group

Expected Result: Learner can create and join a group for a project successfully

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that collaboration tools are available for group communication\*\***

Test Case ID: SLF-002

Test Case Title: Verify that collaboration tools are available for group communication

Test Case Description: This test case ensures that collaboration tools, such as chat, video conferencing, and file sharing, are available for group communication and facilitation.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- \* Learner is logged in to the platform
- \* Learner is a member of a group
- \* Group project assignment is available

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Navigate to the group page
3. Verify that collaboration tools (chat, video conferencing, file sharing) are available
4. Test each collaboration tool to ensure it functions as expected
5. Verify that other group members can see and participate in the collaboration tools

Postconditions:

- \* Collaboration tools are visible and functional in the group page

Expected Result: Collaboration tools are available and functional for group communication and facilitation

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that learners can share resources and feedback within groups\*\***

Test Case ID: SLF-003

Test Case Title: Verify that learners can share resources and feedback within groups

Test Case Description: This test case validates that learners can share resources, such as files and links, and provide feedback to each other within a group.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- \* Learner is logged in to the platform
- \* Learner is a member of a group
- \* Group project assignment is available

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Navigate to the group page
3. Upload a resource (file or link) to the group page
4. Verify that the resource is visible to other group members
5. Provide feedback on a resource shared by another group member
6. Verify that the feedback is visible to the resource owner and other group members

Postconditions:

- \* Resources are shared and visible to group members

- \* Feedback is visible to the resource owner and other group members

Expected Result: Learners can share resources and feedback within a group successfully

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that instructors can assign group projects and monitor progress\*\***

Test Case ID: SLF-004

Test Case Title: Verify that instructors can assign group projects and monitor progress

Test Case Description: This test case ensures that instructors can assign group projects and monitor progress, enabling them to facilitate peer-to-peer learning and collaboration.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- \* Instructor is logged in to the platform

- \* Course is available with group project assignment

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an instructor

2. Navigate to the course page
3. Assign a group project to learners
4. Verify that the group project is visible to learners
5. Monitor progress of the group project
6. Verify that instructor can view learner contributions and progress

Postconditions:

- \* Group project is assigned and visible to learners
- \* Instructor can view learner contributions and progress

Expected Result: Instructors can assign group projects and monitor progress successfully

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that social sharing features allow learners to share achievements\*\***

Test Case ID: SLF-005

Test Case Title: Verify that social sharing features allow learners to share achievements

Test Case Description: This test case validates that learners can share their achievements, such as completed courses or earned certifications, on social media platforms.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:



\* Learner is logged in to the platform

\* Learner has completed a course or earned a certification

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner
2. Navigate to the course completion or certification page
3. Click on the "Share" button
4. Verify that social media platforms (e.g., LinkedIn, Twitter, Facebook) are available for sharing
5. Share the achievement on a social media platform
6. Verify that the achievement is shared successfully

Postconditions:

\* Achievement is shared on the selected social media platform

Expected Result: Learners can share their achievements on social media platforms successfully

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 66**

**Project name -**

## Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Content Authoring Tools

### **Feature description -**

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

### **Userstory title -**

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

### **Acceptance criteria -**

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

### **Generated Test Cases:**

Here are five detailed test cases for the Content Authoring Tools feature, ensuring that each acceptance criterion is covered:

**\*\*Test Case 1: Verify that instructors can create multimedia presentations\*\***

Test Case ID: CATT-01

Test Case Title: Verify that instructors can create multimedia presentations

Test Case Description: This test case ensures that instructors can create multimedia presentations using the content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- \* Instructors have a valid account on the online learning platform
- \* Instructors have access to the content authoring tools
- \* Instructors have a stable internet connection

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor
2. Navigate to the content authoring tools section
3. Click on the "Create New Presentation" button
4. Enter a title and description for the presentation
5. Add multimedia elements such as images, videos, and audio files
6. Click on the "Save" button

Postconditions:

- \* The multimedia presentation is created and saved
- \* The presentation is displayed in the instructor's dashboard

Expected Result: Instructors can successfully create multimedia presentations using the content authoring

tools.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that interactive modules can be developed and integrated\*\***

Test Case ID: CATT-02

Test Case Title: Verify that interactive modules can be developed and integrated

Test Case Description: This test case ensures that instructors can develop and integrate interactive modules using the content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- \* Instructors have a valid account on the online learning platform
- \* Instructors have access to the content authoring tools
- \* Instructors have a stable internet connection

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor
2. Navigate to the content authoring tools section
3. Click on the "Create New Module" button

4. Select the type of interactive module to create (e.g., quiz, game, simulation)
5. Configure the module settings and add content
6. Click on the "Save" button

Postconditions:

- \* The interactive module is created and saved
- \* The module is displayed in the instructor's dashboard

Expected Result: Instructors can successfully develop and integrate interactive modules using the content authoring tools.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that rich media resources are supported\*\***

Test Case ID: CATT-03

Test Case Title: Verify that rich media resources are supported

Test Case Description: This test case ensures that the content authoring tools support rich media resources such as videos and animations.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

- \* Instructors have a valid account on the online learning platform
- \* Instructors have access to the content authoring tools
- \* Instructors have a stable internet connection

Test Data: Sample video and animation files

Test Steps:

1. Log in to the online learning platform as an instructor
2. Navigate to the content authoring tools section
3. Click on the "Add Media" button
4. Upload a video file and an animation file
5. Verify that the files are uploaded successfully
6. Click on the "Save" button

Postconditions:

- \* The rich media resources are uploaded and saved
- \* The resources are displayed in the instructor's dashboard

Expected Result: The content authoring tools support rich media resources such as videos and animations.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that content authoring tools are user-friendly and intuitive\*\***

Test Case ID: CATT-04

Test Case Title: Verify that content authoring tools are user-friendly and intuitive

Test Case Description: This test case ensures that the content authoring tools are user-friendly and intuitive for instructors.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

- \* Instructors have a valid account on the online learning platform
- \* Instructors have access to the content authoring tools
- \* Instructors have a stable internet connection

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as an instructor
2. Navigate to the content authoring tools section
3. Verify that the interface is clean and organized
4. Verify that the toolbars and menus are easy to understand
5. Verify that the help documentation is available and accessible

Postconditions:

- \* The content authoring tools are user-friendly and intuitive
- \* Instructors can easily navigate the interface

Expected Result: The content authoring tools are user-friendly and intuitive, making it easy for instructors to create engaging and interactive course materials.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that instructors can preview and edit content before publishing\*\***

Test Case ID: CATT-05

Test Case Title: Verify that instructors can preview and edit content before publishing

Test Case Description: This test case ensures that instructors can preview and edit content before publishing using the content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

\* Instructors have a valid account on the online learning platform

\* Instructors have access to the content authoring tools

\* Instructors have a stable internet connection

Test Data: Sample course content

Test Steps:

1. Log in to the online learning platform as an instructor
2. Navigate to the content authoring tools section



3. Create a new course content
4. Click on the "Preview" button
5. Verify that the content is displayed correctly
6. Make changes to the content and click on the "Save" button
7. Verify that the changes are saved successfully

Postconditions:

- \* The content is previewed correctly
- \* The content can be edited and saved successfully

Expected Result: Instructors can successfully preview and edit content before publishing using the content authoring tools.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 67**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Learning Paths and Curriculum

### **Feature description -**

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence and continuity in learners' educational journeys.

### **Userstory title -**

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

### **Acceptance criteria -**

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

### **Generated Test Cases:**

Here are five detailed test cases for the Learning Paths and Curriculum feature:

**\*\*Test Case 1: Verify that learning paths include sequential courses and modules\*\***

Test Case ID: LP-001

Test Case Title: Verify that learning paths include sequential courses and modules

Test Case Description: This test case verifies that the learning path includes a series of courses and modules that are sequentially arranged to achieve a specific learning objective.

Test Suite: Learning Paths

Test Priority: High

Preconditions:

- \* The learner is logged in to the platform.
- \* A learning path with multiple courses and modules is created.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the learning path section.
3. Select a learning path with multiple courses and modules.
4. Verify that the courses and modules are listed in a sequential order.
5. Click on each course and module to verify that they are accessible and in the correct order.

Postconditions:

- \* The learner can access the next course or module in the sequence.
- \* The learner can track their progress within the learning path.

Expected Result: The learning path includes a series of courses and modules that are sequentially arranged to achieve a specific learning objective.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that curricula are aligned with specific learning objectives\*\***

Test Case ID: LP-002

Test Case Title: Verify that curricula are aligned with specific learning objectives

Test Case Description: This test case verifies that the curriculum is designed to achieve specific learning objectives and that the learning objectives are clearly stated.

Test Suite: Learning Paths

Test Priority: High

Preconditions:

- \* The learner is logged in to the platform.
- \* A curriculum with specific learning objectives is created.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the curriculum section.
3. Select a curriculum with specific learning objectives.
4. Verify that the learning objectives are clearly stated and align with the curriculum content.
5. Review the curriculum content to ensure it aligns with the learning objectives.

Postconditions:

- \* The learner can understand the learning objectives of the curriculum.
- \* The curriculum content is aligned with the learning objectives.

Expected Result: The curriculum is designed to achieve specific learning objectives and the learning

objectives are clearly stated.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that learners can track their progress within a learning path\*\***

Test Case ID: LP-003

Test Case Title: Verify that learners can track their progress within a learning path

Test Case Description: This test case verifies that the learner can track their progress within a learning path, including completed courses and modules.

Test Suite: Learning Paths

Test Priority: Medium

Preconditions:

- \* The learner is logged in to the platform.
- \* A learning path with multiple courses and modules is created.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the learning path section.
3. Select a learning path with multiple courses and modules.
4. Complete a course or module within the learning path.
5. Verify that the completed course or module is tracked and visible in the learner's progress.

Postconditions:

- \* The learner can view their progress within the learning path.
- \* The completed courses and modules are tracked and updated.

Expected Result: The learner can track their progress within a learning path, including completed courses and modules.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that instructors can customize learning paths for different needs\*\***

Test Case ID: LP-004

Test Case Title: Verify that instructors can customize learning paths for different needs

Test Case Description: This test case verifies that instructors can customize learning paths for different learner needs, including adjusting the course sequence or adding/removing modules.

Test Suite: Learning Paths

Test Priority: High

Preconditions:

- \* The instructor is logged in to the platform.
- \* A learning path with multiple courses and modules is created.

Test Data: No test data needed

#### Test Steps:

1. Log in to the platform as an instructor.
2. Navigate to the learning path section.
3. Select a learning path with multiple courses and modules.
4. Customize the learning path by adjusting the course sequence or adding/removing modules.
5. Verify that the customized learning path is saved and accessible to learners.

#### Postconditions:

- \* The customized learning path is saved and accessible to learners.
- \* The instructor can track learner progress within the customized learning path.

Expected Result: Instructors can customize learning paths for different learner needs, including adjusting the course sequence or adding/removing modules.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that notifications are sent for upcoming courses or milestones\*\***

Test Case ID: LP-005

Test Case Title: Verify that notifications are sent for upcoming courses or milestones

Test Case Description: This test case verifies that the platform sends notifications to learners about upcoming courses or milestones within a learning path.

Test Suite: Learning Paths

Test Priority: Medium

Preconditions:

- \* The learner is logged in to the platform.
- \* A learning path with multiple courses and modules is created.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Navigate to the learning path section.
3. Select a learning path with multiple courses and modules.
4. Set up notifications for upcoming courses or milestones.
5. Verify that notifications are sent to the learner about upcoming courses or milestones.

Postconditions:

- \* The learner receives notifications about upcoming courses or milestones.
- \* The notifications are timely and accurate.

Expected Result: The platform sends notifications to learners about upcoming courses or milestones within a learning path.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive



-----

**Data - 68**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Attendance Tracking

**Feature description -**

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

**Userstory title -**

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

**Acceptance criteria -**

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

**Generated Test Cases:**

Here are five detailed test cases for the Attendance Tracking feature:

**\*\*Test Case 1: Verify that attendance is automatically recorded during live sessions\*\***

Test Case ID: AT01

Test Case Title: Verify that attendance is automatically recorded during live sessions

Test Case Description: This test case verifies that the attendance tracking system automatically records learner attendance during live sessions.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

? Live session is scheduled and active

? Learner is registered for the course and has access to the live session

? Learner joins the live session

Test Data: No test data needed

Test Steps:

1. Instructor schedules a live session and starts it.
2. Learner joins the live session using their credentials.
3. Wait for the live session to end.
4. Check the attendance report for the live session.

Postconditions:

? Attendance report is generated for the live session

? Learner's attendance is recorded as "Present" in the attendance report

Expected Result: Attendance is automatically recorded during live sessions.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that instructors can view attendance reports for each session\*\***

Test Case ID: AT02

Test Case Title: Verify that instructors can view attendance reports for each session

Test Case Description: This test case verifies that instructors can view attendance reports for each live session.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

? Live session has ended

? Instructor has access to the attendance tracking feature

Test Data: No test data needed

Test Steps:

1. Instructor logs in to the platform and navigates to the attendance tracking feature.
2. Instructor selects the live session for which they want to view the attendance report.
3. Check that the attendance report is displayed with learner names and attendance status.

Postconditions:

? Attendance report is displayed with learner names and attendance status

? Instructor can view attendance reports for each live session

Expected Result: Instructors can view attendance reports for each live session.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that learners can see their attendance records\*\***

Test Case ID: AT03

Test Case Title: Verify that learners can see their attendance records

Test Case Description: This test case verifies that learners can view their attendance records for live sessions.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

? Learner has attended at least one live session

? Learner has access to their attendance records

Test Data: No test data needed

Test Steps:

1. Learner logs in to the platform and navigates to their attendance records.
2. Check that the attendance records are displayed with live session names and attendance status.
3. Verify that learner's attendance records are accurate.

Postconditions:

? Attendance records are displayed with live session names and attendance status

? Learner can view their attendance records

Expected Result: Learners can see their attendance records.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that attendance data is integrated with participation metrics\*\***

Test Case ID: AT04

Test Case Title: Verify that attendance data is integrated with participation metrics

Test Case Description: This test case verifies that attendance data is integrated with participation metrics, providing a comprehensive picture of learner engagement.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

? Live session has ended

? Learner has participated in the live session

Test Data: No test data needed

Test Steps:

1. Instructor logs in to the platform and navigates to the attendance tracking feature.
2. Check that the attendance data is integrated with participation metrics, such as engagement scores or quiz results.
3. Verify that the integrated data provides a comprehensive picture of learner engagement.

Postconditions:

? Attendance data is integrated with participation metrics

? Integrated data provides a comprehensive picture of learner engagement

Expected Result: Attendance data is integrated with participation metrics.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that notifications are sent for missed sessions\*\***

Test Case ID: AT05

Test Case Title: Verify that notifications are sent for missed sessions

Test Case Description: This test case verifies that learners receive notifications when they miss a live session.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

? Live session has ended

? Learner was registered for the live session but did not attend

Test Data: No test data needed

Test Steps:

1. Verify that learner did not attend the live session.
2. Check that a notification is sent to the learner's email or in-platform messaging system.
3. Verify that the notification includes details about the missed live session.

Postconditions:

? Notification is sent to the learner's email or in-platform messaging system

? Notification includes details about the missed live session

Expected Result: Notifications are sent for missed sessions.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **Data - 69**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Content Licensing and Rights Management

### **Feature description -**

The platform implements robust content licensing and rights management features to protect intellectual property and ensure compliance with copyright laws. By managing content licenses, usage rights, and permissions, the platform safeguards the interests of content creators and providers, enabling secure and lawful distribution of educational materials.

### **Userstory title -**

As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

## Acceptance criteria -

['Content licenses are assigned to each resource.', 'Usage rights and permissions are clearly defined.', 'Content creators can set restrictions on usage and distribution.', 'Instructors and learners are informed of content licensing terms.', 'Notifications are sent for license expiration or renewal.']

## Generated Test Cases:

Here are five detailed test cases for the Content Licensing and Rights Management feature:

**\*\*Test Case 1: Verify that content licenses are assigned to each resource\*\***

Test Case ID: CLRM-001

Test Case Title: Verify that content licenses are assigned to each resource

Test Case Description: This test case verifies that each resource on the platform has an assigned content license, ensuring protection of intellectual property.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

\* Content creator has uploaded a new resource to the platform

\* Resource is in a published state

Test Data: No test data needed

Test Steps:

1. Log in as a content creator and upload a new resource to the platform
2. Verify that the resource has an assigned content license
3. Check the license details, such as license type, duration, and restrictions



4. Verify that the license is correctly displayed on the resource page

Postconditions:

- \* Resource is available for use on the platform

- \* Content license is visible on the resource page

Expected Result: Each resource has an assigned content license with correct details displayed.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that usage rights and permissions are clearly defined\*\***

Test Case ID: CLRM-002

Test Case Title: Verify that usage rights and permissions are clearly defined

Test Case Description: This test case verifies that usage rights and permissions for each resource are clearly defined and accessible to instructors and learners.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- \* Content creator has set usage rights and permissions for a resource

- \* Instructor or learner has access to the resource

Test Data: No test data needed

Test Steps:

1. Log in as an instructor or learner and access a resource
2. Verify that usage rights and permissions are clearly displayed on the resource page
3. Check that the permissions are correct based on the content creator's settings
4. Verify that the usage rights and permissions are accessible from the resource page

Postconditions:

- \* Instructor or learner can view usage rights and permissions
- \* Correct permissions are applied based on the content creator's settings

Expected Result: Usage rights and permissions are clearly defined and accessible to instructors and learners.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that content creators can set restrictions on usage and distribution\*\***

Test Case ID: CLRM-003

Test Case Title: Verify that content creators can set restrictions on usage and distribution

Test Case Description: This test case verifies that content creators can set restrictions on usage and distribution of their resources, such as limiting downloads or specifying specific use cases.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

- \* Content creator has uploaded a resource to the platform

\* Resource is in a published state

Test Data: No test data needed

Test Steps:

1. Log in as a content creator and access a published resource
2. Verify that the content creator can set restrictions on usage and distribution
3. Set a restriction on downloads, such as limiting the number of downloads
4. Verify that the restriction is correctly applied and displayed on the resource page

Postconditions:

\* Restriction is correctly applied and displayed on the resource page

\* Content creator can modify or remove restrictions

Expected Result: Content creators can set restrictions on usage and distribution of their resources.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that instructors and learners are informed of content licensing terms\*\***

Test Case ID: CLRM-004

Test Case Title: Verify that instructors and learners are informed of content licensing terms

Test Case Description: This test case verifies that instructors and learners are informed of content licensing terms and conditions when accessing a resource.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

- \* Instructor or learner has accessed a resource with licensing terms

- \* Resource is in a published state

Test Data: No test data needed

Test Steps:

1. Log in as an instructor or learner and access a resource with licensing terms
2. Verify that the licensing terms and conditions are displayed
3. Check that the terms are clear and easy to understand
4. Verify that the terms are accessible from the resource page

Postconditions:

- \* Instructors and learners are aware of content licensing terms and conditions

- \* Correct licensing terms are displayed on the resource page

Expected Result: Instructors and learners are informed of content licensing terms and conditions.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that notifications are sent for license expiration or renewal\*\***

Test Case ID: CLRM-005

Test Case Title: Verify that notifications are sent for license expiration or renewal

Test Case Description: This test case verifies that notifications are sent to content creators, instructors, and learners when a license is about to expire or has been renewed.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

- \* A resource license is approaching expiration or has been renewed
- \* Notification settings are enabled for content creators, instructors, and learners

Test Data: No test data needed

Test Steps:

1. Set up a resource license to expire within a specified timeframe
2. Verify that a notification is sent to the content creator, instructor, and learner before the license expires
3. Renew the license and verify that a notification is sent to the content creator, instructor, and learner
4. Check that the notifications contain correct information and are sent at the correct time

Postconditions:

- \* Notifications are sent to content creators, instructors, and learners
- \* Correct information is included in the notifications

Expected Result: Notifications are sent for license expiration or renewal to content creators, instructors, and learners.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

## **Data - 70**

### **Project name -**

Online Learning Platform

### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

Content Versioning

### **Feature description -**

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

### **Userstory title -**

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

### **Acceptance criteria -**

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.', 'Notifications are sent for significant content updates.']

### **Generated Test Cases:**

Here are five detailed test cases for the Content Versioning feature:

**\*\*Test Case 1: Verify that instructors can create and save multiple versions of content\*\***

Test Case ID: CV-001

Test Case Title: Verify that instructors can create and save multiple versions of content

Test Case Description: This test case ensures that instructors can create and save multiple versions of content, allowing them to manage and track changes to their course materials.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in
- \* A course is created with at least one lecture
- \* The lecture contains a content item (e.g., video, PDF, or text)

Test Data: No test data needed

Test Steps:

1. Login as an instructor and navigate to the course dashboard
2. Select a lecture and click on the "Edit" button
3. Make changes to the content item (e.g., add a new video or update the PDF)
4. Click on the "Save as new version" button
5. Verify that a new version of the content item is created and saved
6. Repeat steps 3-5 to create multiple versions of the content item

Postconditions:

- \* The instructor can view a list of all versions of the content item
- \* Each version is dated and timestamped

Expected Result: Instructors can create and save multiple versions of content, and each version is tracked and recorded.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that version history is maintained and accessible\*\***

Test Case ID: CV-002

Test Case Title: Verify that version history is maintained and accessible

Test Case Description: This test case ensures that the platform maintains a version history of all changes made to content items and that instructors can access this history.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- \* Instructor account is created and logged in
- \* A course is created with at least one lecture
- \* The lecture contains a content item with multiple versions

Test Data: No test data needed



## Test Steps:

1. Login as an instructor and navigate to the course dashboard
2. Select a lecture and click on the "Version History" button
3. Verify that a list of all versions of the content item is displayed, including the current version
4. Click on a previous version to view its details
5. Verify that the version history page displays the changes made between versions

## Postconditions:

- \* The version history page displays all changes made to the content item
- \* Instructors can easily access and view previous versions of the content item

Expected Result: The platform maintains a version history of all changes made to content items, and instructors can access and view previous versions.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that instructors can compare different versions of content\*\***

Test Case ID: CV-003

Test Case Title: Verify that instructors can compare different versions of content

Test Case Description: This test case ensures that instructors can compare different versions of content to identify changes made between versions.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in
- \* A course is created with at least one lecture
- \* The lecture contains a content item with multiple versions

Test Data: No test data needed

Test Steps:

1. Login as an instructor and navigate to the course dashboard
2. Select a lecture and click on the "Compare Versions" button
3. Select two versions of the content item to compare
4. Verify that a comparison page is displayed, highlighting the differences between the two versions
5. Verify that instructors can switch between the two versions to view the changes

Postconditions:

- \* The comparison page displays the differences between the two versions
- \* Instructors can easily identify changes made between versions

Expected Result: Instructors can compare different versions of content to identify changes made between versions.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that previous versions can be restored if needed\*\***

Test Case ID: CV-004

Test Case Title: Verify that previous versions can be restored if needed

Test Case Description: This test case ensures that instructors can restore previous versions of content if needed.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in
- \* A course is created with at least one lecture
- \* The lecture contains a content item with multiple versions

Test Data: No test data needed

Test Steps:

1. Login as an instructor and navigate to the course dashboard
2. Select a lecture and click on the "Version History" button
3. Select a previous version of the content item to restore
4. Click on the "Restore" button to restore the previous version
5. Verify that the previous version is restored as the current version

Postconditions:

- \* The previous version is restored as the current version
- \* The version history page is updated to reflect the restored version

Expected Result: Instructors can restore previous versions of content if needed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that notifications are sent for significant content updates\*\***

Test Case ID: CV-005

Test Case Title: Verify that notifications are sent for significant content updates

Test Case Description: This test case ensures that the platform sends notifications to instructors when significant content updates are made.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- \* Instructor account is created and logged in
- \* A course is created with at least one lecture
- \* The lecture contains a content item with multiple versions
- \* Notification settings are configured to send notifications for significant content updates

Test Data: No test data needed

#### Test Steps:

1. Login as an instructor and navigate to the course dashboard
2. Select a lecture and make a significant update to the content item (e.g., add a new video or update the PDF)
3. Click on the "Save as new version" button
4. Verify that a notification is sent to the instructor's email or in-app notification
5. Verify that the notification includes details about the changes made to the content item

#### Postconditions:

- \* The notification is sent to the instructor's email or in-app notification
- \* The notification includes details about the changes made to the content item

Expected Result: The platform sends notifications to instructors when significant content updates are made.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 71**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

API Integration

**Feature description -**

The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

**Userstory title -**

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

**Acceptance criteria -**

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

**Generated Test Cases:**

Here are five detailed test cases for the API Integration feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that APIs are available for key platform functionalities\*\***

Test Case ID: API-001

Test Case Title: Verify that APIs are available for key platform functionalities

Test Case Description: This test case checks if the platform provides APIs for key functionalities such as course creation, user management, and content retrieval.

Test Suite: API Integration

Test Priority: High

Preconditions:

- \* The API documentation is available
- \* A valid API key is obtained

Test Data: No test data needed

Test Steps:

1. Review the API documentation to identify the available APIs for key platform functionalities.
2. Use a REST client (e.g., Postman) to send API requests to the platform for key functionalities (e.g., creating a new course, retrieving course content, managing user profiles).
3. Verify that the API responses are successful (200 OK) and contain the expected data.

Postconditions:

- \* The API requests are logged for further analysis.
- \* The API responses are reviewed for any errors or inconsistencies.

Expected Result: The platform provides APIs for key functionalities, and the API responses are successful with expected data.

Severity: Blocker

Priority: High

Type of Testing: API Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that comprehensive API documentation is provided\*\***

Test Case ID: API-002

Test Case Title: Verify that comprehensive API documentation is provided

Test Case Description: This test case checks if the platform provides comprehensive API documentation that covers all available APIs, parameters, and response formats.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

\* The API documentation is available

Test Data: No test data needed

Test Steps:

1. Review the API documentation to ensure it covers all available APIs, parameters, and response formats.
2. Verify that the documentation is easy to understand, well-organized, and includes code samples and examples.
3. Check if the documentation provides information on error handling, rate limiting, and authentication.

Postconditions:

\* The API documentation is reviewed for any inconsistencies or updates.

Expected Result: The platform provides comprehensive API documentation that covers all available APIs, parameters, and response formats.

Severity: Major

Priority: Medium

Type of Testing: API Testing



Test Case Approach: Positive

**\*\*Test Case 3: Verify that secure access protocols are implemented\*\***

Test Case ID: API-003

Test Case Title: Verify that secure access protocols are implemented

Test Case Description: This test case checks if the platform implements secure access protocols (e.g., HTTPS, OAuth, SSL/TLS) to protect API requests and data.

Test Suite: API Integration

Test Priority: High

Preconditions:

\* A valid API key is obtained

Test Data: No test data needed

Test Steps:

1. Verify that the platform uses HTTPS protocol for all API requests.
2. Check if the platform supports OAuth or other secure authentication mechanisms.
3. Test if the platform encrypts sensitive data (e.g., passwords, credit card numbers) in transit.

Postconditions:

- \* The API requests are monitored for any security breaches.
- \* The platform is reviewed for any security vulnerabilities.

Expected Result: The platform implements secure access protocols to protect API requests and data.

Severity: Critical

Priority: High

Type of Testing: Security Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that third-party applications can authenticate and access APIs\*\***

Test Case ID: API-004

Test Case Title: Verify that third-party applications can authenticate and access APIs

Test Case Description: This test case checks if third-party applications can authenticate and access the platform's APIs using valid credentials.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

\* A valid API key is obtained

\* A third-party application is set up with valid credentials

Test Data: No test data needed

Test Steps:

1. Use a third-party application to send an API request to the platform using valid credentials.
2. Verify that the API request is successful (200 OK) and the platform returns the expected data.
3. Test if the platform authenticates and authorizes the third-party application correctly.

Postconditions:

- \* The API requests are monitored for any unauthorized access.
- \* The platform is reviewed for any authentication or authorization issues.

Expected Result: Third-party applications can authenticate and access the platform's APIs using valid credentials.

Severity: Major

Priority: Medium

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that rate limiting and monitoring features are in place\*\***

Test Case ID: API-005

Test Case Title: Verify that rate limiting and monitoring features are in place

Test Case Description: This test case checks if the platform implements rate limiting and monitoring features to prevent abuse and ensure API performance.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

- \* The API documentation is available

Test Data: No test data needed

Test Steps:

1. Review the API documentation to identify the rate limiting policies and thresholds.
2. Use a REST client (e.g., Postman) to send API requests to the platform at a high frequency (e.g., 100

requests per minute).

3. Verify that the platform enforces rate limiting and returns an error response (e.g., 429 Too Many Requests) when the threshold is exceeded.

Postconditions:

- \* The API requests are monitored for any rate limiting issues.

- \* The platform is reviewed for any performance issues.

Expected Result: The platform implements rate limiting and monitoring features to prevent abuse and ensure API performance.

Severity: Major

Priority: Medium

Type of Testing: Performance Testing

Test Case Approach: Negative

-----

**Data - 72**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Data Analytics and Insights

## **Feature description -**

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

## **Userstory title -**

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

## **Acceptance criteria -**

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

## **Generated Test Cases:**

Here are five detailed test cases for the Data Analytics and Insights feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that data visualization tools are available for analysis.\*\***

Test Case ID: DA-001

Test Case Title: Verify that data visualization tools are available for analysis.

Test Case Description: This test case ensures that data visualization tools are available for administrators to analyze learner performance and engagement.

Test Suite: Data Analytics

Test Priority: High

Preconditions:

\* Administrator account is created and logged in.

\* At least one course is created with learner data.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an administrator.
2. Navigate to the Data Analytics dashboard.
3. Verify that data visualization tools such as charts, graphs, and tables are available.
4. Verify that the visualization tools display learner performance and engagement data.
5. Verify that the visualization tools are interactive, allowing administrators to filter and drill down into the data.

Postconditions:

- \* No changes are made to the learner data.
- \* The administrator can log out of the platform.

Expected Result: Data visualization tools are available and display learner performance and engagement data correctly.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that reports can be generated for learner performance metrics.\*\***

Test Case ID: DA-002

Test Case Title: Verify that reports can be generated for learner performance metrics.

Test Case Description: This test case ensures that administrators can generate reports for learner performance metrics.

Test Suite: Data Analytics

Test Priority: High

Preconditions:

- \* Administrator account is created and logged in.
- \* At least one course is created with learner data.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an administrator.
2. Navigate to the Reports section.
3. Select a course and choose a report type (e.g., learner progress, course completion).
4. Verify that the report is generated correctly, displaying learner performance metrics.
5. Verify that the report can be exported in various formats (e.g., PDF, CSV).

Postconditions:

- \* No changes are made to the learner data.
- \* The administrator can log out of the platform.

Expected Result: Reports can be generated correctly, displaying learner performance metrics.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that real-time analytics provide insights into engagement and outcomes.\*\***

Test Case ID: DA-003

Test Case Title: Verify that real-time analytics provide insights into engagement and outcomes.

Test Case Description: This test case ensures that real-time analytics provide insights into learner engagement and outcomes.

Test Suite: Data Analytics

Test Priority: High

Preconditions:

- \* Administrator account is created and logged in.

- \* At least one course is created with learner data.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an administrator.
2. Navigate to the Real-Time Analytics dashboard.
3. Verify that real-time data is displayed for learner engagement and outcomes.
4. Verify that the real-time data is updated correctly as learners interact with the course.
5. Verify that the real-time analytics provide insights into learner behavior and course effectiveness.

Postconditions:

- \* No changes are made to the learner data.



\* The administrator can log out of the platform.

Expected Result: Real-time analytics provide insights into learner engagement and outcomes correctly.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that data can be exported for further analysis.\*\***

Test Case ID: DA-004

Test Case Title: Verify that data can be exported for further analysis.

Test Case Description: This test case ensures that administrators can export data for further analysis.

Test Suite: Data Analytics

Test Priority: Medium

Preconditions:

\* Administrator account is created and logged in.

\* At least one course is created with learner data.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an administrator.
2. Navigate to the Data Analytics dashboard.
3. Verify that data can be exported in various formats (e.g., CSV, Excel).
4. Verify that the exported data is correct and matches the data displayed in the platform.

5. Verify that the exported data can be opened and analyzed in external tools (e.g., Excel, Tableau).

Postconditions:

- \* No changes are made to the learner data.
- \* The administrator can log out of the platform.

Expected Result: Data can be exported correctly for further analysis.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that notifications are sent for significant data trends or anomalies.\*\***

Test Case ID: DA-005

Test Case Title: Verify that notifications are sent for significant data trends or anomalies.

Test Case Description: This test case ensures that notifications are sent to administrators for significant data trends or anomalies.

Test Suite: Data Analytics

Test Priority: Medium

Preconditions:

- \* Administrator account is created and logged in.
- \* At least one course is created with learner data.

Test Data: No test data needed

#### Test Steps:

1. Log in to the platform as an administrator.
2. Navigate to the Data Analytics dashboard.
3. Verify that notification settings are available for significant data trends or anomalies.
4. Simulate a significant data trend or anomaly (e.g., sudden increase in learner engagement).
5. Verify that a notification is sent to the administrator correctly.

#### Postconditions:

- \* No changes are made to the learner data.
- \* The administrator can log out of the platform.

Expected Result: Notifications are sent correctly for significant data trends or anomalies.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

#### **Data - 73**

##### **Project name -**

Online Learning Platform

##### **Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

### **Feature name -**

User Roles and Permissions

### **Feature description -**

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

### **Userstory title -**

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

### **Acceptance criteria -**

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

### **Generated Test Cases:**

Here are five detailed test cases for the User Roles and Permissions feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that administrators can create and assign user roles\*\***

Test Case ID: TC-URP-001

Test Case Title: Verify that administrators can create and assign user roles

Test Case Description: This test case verifies that administrators can create new user roles and assign them to users.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- \* Administrator is logged in to the platform
- \* There are no existing user roles with the same name

Test Data: Role name, role description, and user credentials

Test Steps:

1. Log in as an administrator and navigate to the User Roles page
2. Click on the "Create New Role" button and enter role details (name, description)
3. Click on the "Save" button to create the new role
4. Navigate to the Users page and select a user
5. Click on the "Assign Role" button and select the newly created role
6. Click on the "Save" button to assign the role to the user

Postconditions:

- \* The new role is created and visible in the User Roles page
- \* The assigned user is updated with the new role

Expected Result: Administrators can create and assign user roles successfully

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that permissions are customizable for each role\*\***

Test Case ID: TC-URP-002

Test Case Title: Verify that permissions are customizable for each role

Test Case Description: This test case verifies that administrators can customize permissions for each user role.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- \* A user role is created and assigned to a user

- \* The user is logged out

Test Data: Role name, permission details (e.g., view courses, edit profiles)

Test Steps:

1. Log in as an administrator and navigate to the User Roles page
2. Select the existing role and click on the "Edit" button
3. Click on the "Permissions" tab and select/deselect permissions accordingly
4. Click on the "Save" button to update the role permissions
5. Log in as the user with the updated role and verify permissions

Postconditions:

- \* The updated role permissions are reflected in the user's account

Expected Result: Permissions are customizable for each user role

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users are restricted to functionalities based on their roles\*\***

Test Case ID: TC-URP-003

Test Case Title: Verify that users are restricted to functionalities based on their roles

Test Case Description: This test case verifies that users can only access features and functionality based on their assigned roles.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- \* A user role is created and assigned to a user

- \* The user is logged in

Test Data: None

Test Steps:

1. Log in as the user and attempt to access a feature not permitted by their role
2. Verify that the feature is not accessible or displays an access denied error
3. Attempt to access a feature permitted by their role
4. Verify that the feature is accessible and functional

Postconditions:

- \* The user's account is updated with the correct access levels

Expected Result: Users are restricted to functionalities based on their roles

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that changes to roles and permissions are logged\*\***

Test Case ID: TC-URP-004

Test Case Title: Verify that changes to roles and permissions are logged

Test Case Description: This test case verifies that changes to user roles and permissions are logged for auditing and tracking purposes.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- \* A user role is created and assigned to a user
- \* The administrator has made changes to the role permissions

Test Data: Role name, permission changes, and administrator credentials

Test Steps:

1. Log in as an administrator and navigate to the User Roles page
2. Select the existing role and click on the "Edit" button
3. Make changes to the role permissions
4. Click on the "Save" button to update the role



5. Navigate to the Audit Logs page and verify the change is logged

Postconditions:

- \* The updated role permissions are reflected in the user's account

- \* The change is logged in the Audit Logs page

Expected Result: Changes to roles and permissions are logged

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users are notified of their assigned roles and permissions\*\***

Test Case ID: TC-URP-005

Test Case Title: Verify that users are notified of their assigned roles and permissions

Test Case Description: This test case verifies that users receive notifications when their roles or permissions are updated.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- \* A user role is created and assigned to a user

- \* The user is logged out

Test Data: Role name, permission details, and user credentials

#### Test Steps:

1. Log in as an administrator and navigate to the User Roles page
2. Select the existing role and click on the "Edit" button
3. Update the role permissions and click on the "Save" button
4. Log in as the user and verify the notification of updated role and permissions

#### Postconditions:

\* The user receives a notification of the updated role and permissions

Expected Result: Users are notified of their assigned roles and permissions

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Note: These test cases cover all the acceptance criteria specified in the user story. Additional test cases may be needed to cover edge cases, negative scenarios, and other aspects of the feature.

-----

**Data - 74**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

User Feedback and Rating System

**Feature description -**

The platform includes a user feedback and rating system, allowing learners to provide feedback on courses, instructors, and learning materials. This feature enables continuous improvement by gathering user insights and ratings, helping instructors refine their content and approach.

**Userstory title -**

As a learner, I want to provide feedback and ratings so that I can share my experience and help improve the platform.

**Acceptance criteria -**

['Learners can rate courses and instructors on a scale.', 'Feedback forms are available for detailed comments.', 'Instructors can view and respond to feedback.', 'Aggregated ratings are displayed publicly.', 'Anonymous feedback options are available.']

**Generated Test Cases:**

Here are five detailed test cases for the User Feedback and Rating System feature:

**\*\*Test Case 1: Rating Course\*\***

Test Case ID: UFC-001

Test Case Title: Verify that learners can rate courses on a scale.

Test Case Description: This test case verifies that learners can rate courses using a rating scale.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- \* The learner is logged in to the platform.
- \* The learner has completed a course.
- \* The course has a rating feature enabled.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Access a completed course with a rating feature enabled.
3. Click on the "Rate this course" button.
4. Select a rating from the rating scale (e.g., 1-5).
5. Submit the rating.

Postconditions:

- \* The rating is saved and displayed on the course page.
- \* The rating is reflected in the aggregated ratings.

Expected Result: The learner can successfully rate a course on a scale, and the rating is displayed on the course page.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Providing Feedback\*\***

Test Case ID: UFC-002

Test Case Title: Verify that learners can provide feedback on courses through a feedback form.

Test Case Description: This test case verifies that learners can provide detailed feedback on courses using a feedback form.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- \* The learner is logged in to the platform.
- \* The learner has completed a course.
- \* The course has a feedback feature enabled.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.
2. Access a completed course with a feedback feature enabled.
3. Click on the "Provide feedback" button.
4. Fill in the feedback form with valid data (e.g., rating, comments).
5. Submit the feedback.

Postconditions:

- \* The feedback is saved and displayed on the course page.
- \* The instructor can view and respond to the feedback.

Expected Result: The learner can successfully provide feedback on a course through a feedback form, and the feedback is displayed on the course page.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Instructor Viewing Feedback\*\***

Test Case ID: UFC-003

Test Case Title: Verify that instructors can view and respond to feedback on courses.

Test Case Description: This test case verifies that instructors can view and respond to feedback on courses.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

- \* The instructor is logged in to the platform.
- \* The course has feedback feature enabled.
- \* There is feedback available on the course.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as an instructor.
2. Access a course with feedback available.
3. Click on the "View feedback" button.
4. Verify that the feedback is displayed.
5. Respond to the feedback.

Postconditions:

\* The response is saved and displayed on the course page.

Expected Result: The instructor can successfully view and respond to feedback on a course, and the response is displayed on the course page.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Anonymous Feedback\*\***

Test Case ID: UFC-004

Test Case Title: Verify that learners can provide anonymous feedback on courses.

Test Case Description: This test case verifies that learners can provide anonymous feedback on courses.

Test Suite: User Feedback and Rating System

Test Priority: Low

Preconditions:

\* The learner is logged in to the platform.

\* The learner has completed a course.

\* The course has anonymous feedback feature enabled.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.

2. Access a completed course with anonymous feedback feature enabled.

3. Click on the "Provide anonymous feedback" button.
4. Fill in the feedback form with valid data (e.g., rating, comments).
5. Submit the feedback.

Postconditions:

- \* The feedback is saved and displayed on the course page without learner's identity.

Expected Result: The learner can successfully provide anonymous feedback on a course, and the feedback is displayed on the course page without learner's identity.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Aggregated Ratings\*\***

Test Case ID: UFC-005

Test Case Title: Verify that aggregated ratings are displayed publicly on course pages.

Test Case Description: This test case verifies that aggregated ratings are displayed publicly on course pages.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- \* The course has a rating feature enabled.
- \* There are multiple ratings available on the course.

Test Data: No test data needed



### Test Steps:

1. Access a course page with multiple ratings.
2. Verify that the aggregated rating is displayed.
3. Verify that the aggregated rating is calculated correctly.

### Postconditions:

\* The aggregated rating is updated when new ratings are submitted.

Expected Result: The aggregated rating is correctly displayed on the course page, and it is updated when new ratings are submitted.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

### Data - 75

#### Project name -

Online Learning Platform

#### Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

#### Feature name -

## Social Media Integration

### **Feature description -**

The platform supports social media integration, allowing learners and instructors to share their achievements, courses, and updates on social media platforms. This feature promotes engagement and community building, enhancing the visibility of the platform's offerings.

### **Userstory title -**

As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

### **Acceptance criteria -**

['Learners can connect their social media accounts to the platform.', 'Achievements and course completions can be shared on social media.', 'Privacy settings control the visibility of shared content.', 'Instructors can promote courses and updates through social media.', 'Analytics track the reach and engagement of shared content.']

### **Generated Test Cases:**

Here are five test cases for the Social Media Integration feature of the Online Learning Platform:

**\*\*Test Case 1: Verify that learners can connect their social media accounts to the platform\*\***

Test Case ID: SMIT-001

Test Case Title: Verify that learners can connect their social media accounts to the platform

Test Case Description: This test case verifies that learners can successfully connect their social media accounts to the platform, enabling them to share their achievements and course completions.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- \* Learner has a valid social media account (e.g., Facebook, Twitter, LinkedIn)
- \* Learner is logged in to the Online Learning Platform
- \* Learner has not previously connected their social media account to the platform

Test Data: No test data needed

Test Steps:

1. Log in to the Online Learning Platform as a learner.
2. Navigate to the Account Settings page.
3. Click on the "Connect Social Media" button.
4. Select the desired social media platform (e.g., Facebook).
5. Enter the social media account credentials and authorize the platform to connect.
6. Verify that the social media account is successfully connected.

Postconditions:

- \* The learner's social media account is connected to the platform.
- \* The learner can view their connected social media account on the Account Settings page.

Expected Result: The learner can successfully connect their social media account to the platform.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that achievements and course completions can be shared on social media\*\***

Test Case ID: SMIT-002

Test Case Title: Verify that achievements and course completions can be shared on social media

Test Case Description: This test case verifies that learners can share their achievements and course completions on their connected social media accounts.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- \* Learner has a connected social media account (e.g., Facebook)
- \* Learner has completed a course or achieved a milestone
- \* Learner is logged in to the Online Learning Platform

Test Data: Course completion data, achievement data

Test Steps:

1. Log in to the Online Learning Platform as a learner.
2. Navigate to the Course Dashboard or Achievement page.
3. Click on the "Share on Social Media" button.
4. Select the desired social media platform (e.g., Facebook).
5. Verify that the shared content appears on the learner's social media timeline.

Postconditions:

- \* The shared content is visible on the learner's social media timeline.
- \* The shared content includes the learner's name, course/achievement details, and a link to the platform.

Expected Result: Learners can successfully share their achievements and course completions on social media.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that privacy settings control the visibility of shared content\*\***

Test Case ID: SMIT-003

Test Case Title: Verify that privacy settings control the visibility of shared content

Test Case Description: This test case verifies that learners can control the visibility of their shared content on social media using privacy settings.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

\* Learner has a connected social media account (e.g., Facebook)

\* Learner has shared content on social media

\* Learner is logged in to the Online Learning Platform

Test Data: Privacy setting data (public, friends, only me)

Test Steps:

1. Log in to the Online Learning Platform as a learner.
2. Navigate to the Account Settings page.
3. Update the privacy settings to "Friends" or "Only Me".
4. Navigate to the social media platform and verify the visibility of the shared content.

Postconditions:

- \* The shared content is only visible to the specified audience (friends or only the learner).
- \* The learner can update the privacy settings to control the visibility of shared content.

Expected Result: Learners can control the visibility of their shared content on social media using privacy settings.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that instructors can promote courses and updates through social media\*\***

Test Case ID: SMIT-004

Test Case Title: Verify that instructors can promote courses and updates through social media

Test Case Description: This test case verifies that instructors can share courses and updates on their connected social media accounts.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- \* Instructor has a connected social media account (e.g., Twitter)
- \* Instructor has created a course or update
- \* Instructor is logged in to the Online Learning Platform

Test Data: Course data, update data

#### Test Steps:

1. Log in to the Online Learning Platform as an instructor.
2. Navigate to the Course Dashboard or Update page.
3. Click on the "Share on Social Media" button.
4. Select the desired social media platform (e.g., Twitter).
5. Verify that the shared content appears on the instructor's social media timeline.

#### Postconditions:

- \* The shared content is visible on the instructor's social media timeline.
- \* The shared content includes the course/update details and a link to the platform.

Expected Result: Instructors can successfully share courses and updates on social media.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that analytics track the reach and engagement of shared content\*\***

Test Case ID: SMIT-005

Test Case Title: Verify that analytics track the reach and engagement of shared content

Test Case Description: This test case verifies that the platform tracks the reach and engagement of shared content on social media.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- \* Learner or instructor has shared content on social media
- \* Analytics tracking is enabled on the platform

Test Data: Analytics data (reach, engagement metrics)

Test Steps:

1. Log in to the Online Learning Platform as an administrator.
2. Navigate to the Analytics dashboard.
3. Verify that the reach and engagement metrics are displayed for shared content.
4. Verify that the metrics are updated in real-time.

Postconditions:

- \* The analytics dashboard displays the reach and engagement metrics for shared content.
- \* The metrics are accurate and up-to-date.

Expected Result: The platform accurately tracks the reach and engagement of shared content on social media.

Severity: Major

Priority: Medium

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 -**

Activity Tracking

**Feature description -**

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity levels and progress towards fitness goals.

**Userstory title -**

As a user, I want to track my activities so that I can monitor my daily physical activity levels.

**Acceptance criteria -**

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']

**Generated Test Cases:**

Here are five detailed test cases for the Activity Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that the app records data on distance, duration, pace, and calories burned\*\***

Test Case ID: AT-001

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 captures data on distance, duration, pace, and calories burned for various activities.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

- \* The user has enabled location services and has a compatible wearable device or fitness equipment connected to the app.
- \* The user has selected an activity type (e.g., running, walking, cycling).
- \* The user has started the activity tracking feature.

Test Data: No test data needed

Test Steps:

1. Start the activity tracking feature for a specific activity type.
2. Perform the activity for a set duration (e.g., 30 minutes).
3. Stop the activity tracking feature.
4. Review the recorded data in the app.

Postconditions:

- \* The app displays the recorded data on distance, duration, pace, and calories burned.
- \* The recorded data is accurate and corresponds to the performed activity.

Expected Result: The app successfully records and displays data on distance, duration, pace, and calories burned for the performed activity.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can view activity summaries\*\***

Test Case ID: AT-002

Test Case Title: Verify that users can view activity summaries

Test Case Description: This test case verifies that users can view summaries of their past activities, including key metrics such as total distance, calories burned, and average pace.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- \* The user has recorded multiple activities using the app.
- \* The user has accessed the activity summary feature.

Test Data: No test data needed

Test Steps:

1. Access the activity summary feature in the app.
2. Review the summary data, including total distance, calories burned, and average pace.
3. Filter the summary data by date range or activity type.

Postconditions:

- \* The app displays an accurate and comprehensive summary of the user's past activities.
- \* The summary data is easily accessible and navigable.

Expected Result: The app successfully displays activity summaries, including key metrics, and allows filtering by date range or activity type.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that data is stored for historical analysis\*\***

Test Case ID: AT-003

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 progress over time.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- \* The user has recorded multiple activities using the app.
- \* The user has accessed the activity history feature.

Test Data: No test data needed

Test Steps:

1. Access the activity history feature in the app.
2. Review the stored activity data, including dates, activity types, and key metrics.
3. Filter the activity history data by date range or activity type.

Postconditions:

- \* The app successfully stores and displays historical activity data.
- \* The stored data is accurate and comprehensive.

Expected Result: The app successfully stores and displays activity data for historical analysis, allowing users to track progress over time.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that activity tracking can be paused and resumed\*\***

Test Case ID: AT-004

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, ensuring accurate data capture and minimizing data loss.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- \* The user has started the activity tracking feature for a specific activity type.
- \* The user has paused the activity tracking feature.

Test Data: No test data needed

Test Steps:

1. Pause the activity tracking feature during an activity.
2. Perform additional activity (e.g., 10 minutes).
3. Resume the activity tracking feature.
4. Review the recorded data in the app.

Postconditions:

- \* The app successfully pauses and resumes activity tracking.
- \* The recorded data is accurate and comprehensive.

Expected Result: The app successfully pauses and resumes activity tracking, ensuring accurate data capture and minimizing data loss.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users receive notifications for milestones\*\***

Test Case ID: AT-005

Test Case Title: Verify that users receive notifications for milestones

Test Case Description: This test case verifies that users receive notifications when they reach specific milestones (e.g., 100km run, 500 calories burned).

Test Suite: Activity Tracking

Test Priority: Low

Preconditions:

- \* The user has set up milestone notifications in the app settings.

\* The user has reached a specific milestone (e.g., 100km run).

Test Data: No test data needed

Test Steps:

1. Set up milestone notifications in the app settings.
2. Perform an activity that reaches the set milestone.
3. Review the notification received.

Postconditions:

- \* The app sends a notification when the user reaches a milestone.
- \* The notification is clear and informative.

Expected Result: The app successfully sends notifications when users reach specific milestones, motivating them to continue their fitness journey.

Severity: Minor

Type of Testing: Functional Testing

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:**

Here are five detailed test cases for the GPS Tracking feature of the Fitness Tracker App, covering all the acceptance criteria:

**\*\*Test Case 1: Verify that GPS tracking accurately records route data\*\***

Test Case ID: FT-GPS-001

Test Case Title: Verify that GPS tracking accurately records route data

Test Suite: GPS Tracking

Test Priority: High

Preconditions:



- \* User has enabled GPS tracking on their device
- \* User has started a new workout session
- \* User has moved to an outdoor location with a clear GPS signal

Test Data: No test data needed

Test Steps:

1. Start a new workout session and enable GPS tracking
2. Perform a 10-minute outdoor activity (e.g., jogging, cycling)
3. Stop the workout session
4. Review the recorded route data on the app
5. Verify that the route data is accurate, including distance traveled, time elapsed, and route taken

Postconditions:

- \* Route data is saved for future reference
- \* Route data is correctly synced with the user's account

Expected Result: The GPS tracking feature accurately records route data, including distance traveled, time elapsed, and route taken.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can view detailed maps and elevation profiles\*\***

Test Case ID: FT-GPS-002

Test Case Title: Verify that users can view detailed maps and elevation profiles

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

\* User has completed a workout session with GPS tracking enabled

\* User has saved the route data

Test Data: No test data needed

Test Steps:

1. Open the workout session details page
2. Tap on the "View Map" button
3. Verify that a detailed map of the route is displayed
4. Tap on the "Elevation Profile" button
5. Verify that an elevation profile of the route is displayed

Postconditions:

\* Map and elevation profile are correctly rendered

\* Map and elevation profile are correctly scaled

Expected Result: Users can view detailed maps and elevation profiles of their workout routes.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that performance data includes location-based metrics\*\***

Test Case ID: FT-GPS-003

Test Case Title: Verify that performance data includes location-based metrics

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- \* User has completed a workout session with GPS tracking enabled

- \* User has saved the route data

Test Data: No test data needed

Test Steps:

1. Open the workout session details page
2. Tap on the "Performance Data" button
3. Verify that location-based metrics, such as pace, speed, and distance, are displayed
4. Verify that the metrics are accurately calculated based on the route data

Postconditions:

- \* Performance data is correctly calculated

\* Performance data is correctly displayed

Expected Result: Performance data includes location-based metrics, such as pace, speed, and distance.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that routes can be saved and revisited\*\***

Test Case ID: FT-GPS-004

Test Case Title: Verify that routes can be saved and revisited

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

\* User has completed a workout session with GPS tracking enabled

\* User has saved the route data

Test Data: No test data needed

Test Steps:

1. Open the workout session details page
2. Tap on the "Save Route" button
3. Verify that the route is saved to the user's route library
4. Tap on the "Route Library" button

5. Verify that the saved route is listed and can be selected for revisiting

Postconditions:

- \* Route is correctly saved to the user's route library

- \* Route is correctly listed in the route library

Expected Result: Routes can be saved and revisited for future reference.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that notifications are provided for deviations from planned routes\*\***

Test Case ID: FT-GPS-005

Test Case Title: Verify that notifications are provided for deviations from planned routes

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- \* User has planned a workout route with GPS tracking enabled

- \* User has deviated from the planned route during a workout session

Test Data: No test data needed

Test Steps:

1. Plan a workout route with GPS tracking enabled
2. Start a new workout session and deviate from the planned route
3. Verify that a notification is displayed indicating the deviation
4. Verify that the notification provides an option to adjust the route or continue with the current route

Postconditions:

\* Notification is correctly displayed

\* Notification provides an option to adjust the route or continue with the current route

Expected Result: Notifications are provided for deviations from planned routes, allowing users to adjust their route or continue as desired.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Here are five test cases for the Heart Rate Monitoring feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that heart rate data is recorded in real-time\*\***

Test Case ID: HRM-001

Test Case Title: Verify that heart rate data is recorded in real-time

Test Case Description: This test case verifies that the app records heart rate data in real-time using built-in sensors or compatible wearable devices.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- \* User has granted permission for heart rate monitoring
- \* User is wearing a compatible wearable device or has enabled built-in sensors

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the heart rate monitoring feature
2. Start a new workout session
3. Perform physical activity to stimulate heart rate changes
4. Observe the heart rate data being recorded in real-time
5. Verify that the heart rate data is updating correctly

Postconditions:

- \* Heart rate data is stored in the app's database
- \* User can view historical heart rate data

Expected Result: Heart rate data is recorded in real-time and updates correctly during the workout session.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can set and monitor target heart rate zones\*\***



Test Case ID: HRM-002

Test Case Title: Verify that users can set and monitor target heart rate zones

Test Case Description: This test case verifies that users can set target heart rate zones and receive alerts when they are outside of these zones during a workout.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- \* User has created an account and logged in
- \* User has enabled heart rate monitoring

Test Data:

- \* Target heart rate zone: 120-140 bpm

Test Steps:

1. Launch the app and navigate to the heart rate monitoring feature
2. Set a target heart rate zone (e.g., 120-140 bpm)
3. Start a new workout session
4. Perform physical activity to stimulate heart rate changes
5. Verify that alerts are triggered when the user's heart rate falls outside of the set target zone

Postconditions:

- \* Target heart rate zone is stored in the app's database
- \* User can view historical heart rate data with target zone indicators

Expected Result: User can set target heart rate zones and receive alerts when they are outside of these zones during a workout.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that historical heart rate data is stored and viewable\*\***

Test Case ID: HRM-003

Test Case Title: Verify that historical heart rate data is stored and viewable

Test Case Description: This test case verifies that historical heart rate data is stored and can be viewed by the user.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

\* User has created an account and logged in

\* User has enabled heart rate monitoring

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the heart rate monitoring feature
2. View historical heart rate data for a previous workout session
3. Verify that the data is displayed correctly, including date, time, and heart rate values

Postconditions:

- \* Historical heart rate data is stored in the app's database

- \* User can view historical heart rate data

Expected Result: Historical heart rate data is stored and can be viewed by the user.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that alerts are provided for abnormal heart rate readings\*\***

Test Case ID: HRM-004

Test Case Title: Verify that alerts are provided for abnormal heart rate readings

Test Case Description: This test case verifies that the app provides alerts for abnormal heart rate readings, such as extremely high or low heart rates.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- \* User has created an account and logged in

- \* User has enabled heart rate monitoring

Test Data:

\* Abnormal heart rate reading: 180 bpm

Test Steps:

1. Launch the app and navigate to the heart rate monitoring feature
2. Simulate an abnormal heart rate reading (e.g., 180 bpm)
3. Verify that an alert is triggered and displayed to the user
4. Verify that the alert includes a warning message and recommendations for action

Postconditions:

\* Alert is stored in the app's database

\* User can view alert history

Expected Result: Alerts are triggered for abnormal heart rate readings and displayed to the user.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative

**\*\*Test Case 5: Verify that data is compatible with various wearable devices\*\***

Test Case ID: HRM-005

Test Case Title: Verify that data is compatible with various wearable devices

Test Case Description: This test case verifies that the app is compatible with various wearable devices and can sync heart rate data correctly.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

- \* User has created an account and logged in
- \* User has enabled heart rate monitoring on a compatible wearable device

Test Data:

- \* Compatible wearable devices: Apple Watch, Fitbit, Garmin

Test Steps:

1. Launch the app and navigate to the heart rate monitoring feature
2. Connect a compatible wearable device (e.g., Apple Watch)
3. Verify that heart rate data is synced correctly from the wearable device
4. Repeat steps 2-3 for multiple wearable devices (e.g., Fitbit, Garmin)

Postconditions:

- \* Heart rate data is stored in the app's database
- \* User can view heart rate data from various wearable devices

Expected Result: The app is compatible with various wearable devices and can sync heart rate data correctly.

Severity: Major

Priority: Medium

Type of Testing: Integration Testing

Test Case Approach: Positive

---

## **Data - 79**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Calorie Tracking

### **Feature description -**

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

### **Userstory title -**

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

### **Acceptance criteria -**

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

### **Generated Test Cases:**

Here are five detailed test cases for the Calorie Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can log daily food intake and calorie consumption\*\***

Test Case ID: CT-001

Test Case Title: Verify that users can log daily food intake and calorie consumption

Test Case Description: This test case ensures that users can successfully log their daily food intake and calorie consumption in the app.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- \* User is logged in to the app
- \* User has access to the Calorie Tracking feature
- \* No existing food logs for the current day

Test Data: Sample food item with calorie information (e.g., Apple, 95 calories)

Test Steps:

1. Launch the app and navigate to the Calorie Tracking feature
2. Click on the "Log Food" button
3. Search for a food item (e.g., Apple) and select it from the database
4. Enter the serving size and quantity consumed
5. Click on the "Save" button to log the food item
6. Verify that the food item is displayed in the daily food log with the correct calorie information

Postconditions:

- \* Food item is saved to the user's daily food log

\* Calorie information is updated in the user's profile

Expected Result: The user can successfully log their daily food intake and calorie consumption.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the app calculates and displays total calories consumed and burned\*\***

Test Case ID: CT-002

Test Case Title: Verify that the app calculates and displays total calories consumed and burned

Test Case Description: This test case ensures that the app accurately calculates and displays the total calories consumed and burned based on user input.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

\* User has logged food items for the current day

\* User has tracked physical activities for the current day

Test Data: Sample food items and physical activities with calorie information

Test Steps:

1. Launch the app and navigate to the Calorie Tracking feature
2. Verify that the total calories consumed and burned are displayed on the dashboard
3. Log additional food items and physical activities



4. Verify that the total calories consumed and burned are updated accurately

Postconditions:

- \* Total calories consumed and burned are updated in the user's profile
- \* Accurate calculation of calorie balance (consumed - burned)

Expected Result: The app accurately calculates and displays the total calories consumed and burned.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can set daily calorie goals and track progress\*\***

Test Case ID: CT-003

Test Case Title: Verify that users can set daily calorie goals and track progress

Test Case Description: This test case ensures that users can set daily calorie goals and track their progress towards achieving those goals.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has access to the Calorie Tracking feature

Test Data: Sample daily calorie goal (e.g., 2000 calories)

### Test Steps:

1. Launch the app and navigate to the Calorie Tracking feature
2. Click on the "Set Goal" button
3. Enter the daily calorie goal and click on the "Save" button
4. Verify that the daily calorie goal is displayed on the dashboard
5. Log food items and physical activities
6. Verify that the progress towards the daily calorie goal is updated accurately

### Postconditions:

- \* Daily calorie goal is saved to the user's profile
- \* Accurate tracking of progress towards the daily calorie goal

Expected Result: Users can set daily calorie goals and track their progress towards achieving those goals.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that food items and calorie information are searchable in the database\*\***

Test Case ID: CT-004

Test Case Title: Verify that food items and calorie information are searchable in the database

Test Case Description: This test case ensures that users can search for food items and retrieve accurate calorie information from the database.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has access to the Calorie Tracking feature

Test Data: Sample food items with calorie information (e.g., Banana, 105 calories)

Test Steps:

1. Launch the app and navigate to the Calorie Tracking feature
2. Click on the "Search Food" button
3. Enter a search term (e.g., Banana)
4. Verify that the search results display the correct food item with calorie information
5. Select a food item from the search results
6. Verify that the food item is displayed with accurate calorie information

Postconditions:

- \* Accurate search results for food items and calorie information
- \* Correct calorie information is displayed for the selected food item

Expected Result: Users can search for food items and retrieve accurate calorie information from the database.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users receive alerts for exceeding calorie goals\*\***

Test Case ID: CT-005

Test Case Title: Verify that users receive alerts for exceeding calorie goals

Test Case Description: This test case ensures that users receive alerts when they exceed their daily calorie goals.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- \* User has set a daily calorie goal
- \* User has logged food items and physical activities

Test Data: Sample daily calorie goal (e.g., 2000 calories) and sample food items exceeding the goal

Test Steps:

1. Launch the app and navigate to the Calorie Tracking feature
2. Log food items that exceed the daily calorie goal
3. Verify that an alert is displayed notifications
4. Verify that the alert includes a message indicating that the daily calorie goal has been exceeded
5. Verify that the alert provides guidance on how to get back on track

Postconditions:

- \* Alert is sent to the user for exceeding the daily calorie goal
- \* Accurate calculation of calorie balance (consumed - burned)

Expected Result: Users receive alerts when they exceed their daily calorie goals.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 80**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Sleep Tracking

**Feature description -**

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

**Userstory title -**

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

**Acceptance criteria -**

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

### **Generated Test Cases:**

Here are five detailed test cases for the Sleep Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that sleep data is recorded including duration and quality\*\***

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 records sleep data, including duration and quality, using accelerometer and heart rate sensors.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- \* User has enabled sleep tracking feature
- \* User has worn the wearable device or fitness equipment during sleep
- \* User has synced the device with the app

Test Data: No test data needed

Test Steps:

1. Enable sleep tracking feature in the app
2. Wear the wearable device or fitness equipment during sleep
3. Sync the device with the app in the morning

4. Check the app for recorded sleep data, including duration and quality

Postconditions:

- \* Sleep data is synced to the app
- \* Sleep data includes duration and quality

Expected Result: Sleep data is recorded and displayed in the app, including duration and quality.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can view detailed sleep stages and patterns\*\***

Test Case ID: ST-002

Test Case Title: Verify that users can view detailed sleep stages and patterns

Test Case Description: This test case verifies that the app displays detailed sleep stages and patterns, including light, deep, and REM sleep.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- \* User has enabled sleep tracking feature
- \* User has worn the wearable device or fitness equipment during sleep
- \* User has synced the device with the app

Test Data: No test data needed

Test Steps:

1. Enable sleep tracking feature in the app
2. Wear the wearable device or fitness equipment during sleep
3. Sync the device with the app in the morning
4. Check the app for detailed sleep stages and patterns
5. Verify that the app displays light, deep, and REM sleep stages

Postconditions:

- \* Sleep stages and patterns are displayed in the app
- \* Sleep stages include light, deep, and REM sleep

Expected Result: The app displays detailed sleep stages and patterns, including light, deep, and REM sleep.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the app provides insights into sleep disturbances\*\***

Test Case ID: ST-003

Test Case Title: Verify that the app provides insights into sleep disturbances

Test Case Description: This test case verifies that the app provides insights into sleep disturbances, such as sleep fragmentation and awakenings.

Test Suite: Sleep Tracking



Test Priority: Medium

Preconditions:

- \* User has enabled sleep tracking feature
- \* User has worn the wearable device or fitness equipment during sleep
- \* User has synced the device with the app

Test Data: No test data needed

Test Steps:

1. Enable sleep tracking feature in the app
2. Wear the wearable device or fitness equipment during sleep
3. Sync the device with the app in the morning
4. Check the app for insights into sleep disturbances
5. Verify that the app displays sleep fragmentation and awakenings

Postconditions:

- \* Sleep disturbances are displayed in the app
- \* Sleep disturbances include sleep fragmentation and awakenings

Expected Result: The app provides insights into sleep disturbances, including sleep fragmentation and awakenings.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users receive recommendations for improving sleep quality\*\***

Test Case ID: ST-004

Test Case Title: Verify that users receive recommendations for improving sleep quality

Test Case Description: This test case verifies that the app provides personalized recommendations for improving sleep quality based on user's sleep data.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- \* User has enabled sleep tracking feature
- \* User has worn the wearable device or fitness equipment during sleep
- \* User has synced the device with the app

Test Data: No test data needed

Test Steps:

1. Enable sleep tracking feature in the app
2. Wear the wearable device or fitness equipment during sleep
3. Sync the device with the app in the morning
4. Check the app for personalized recommendations for improving sleep quality

Postconditions:

- \* Recommendations for improving sleep quality are displayed in the app
- \* Recommendations are personalized based on user's sleep data

Expected Result: The app provides personalized recommendations for improving sleep quality based on user's sleep data.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that sleep data can be viewed in daily, weekly, and monthly summaries\*\***

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 verifies that the app displays sleep data in daily, weekly, and monthly summaries.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- \* User has enabled sleep tracking feature
- \* User has worn the wearable device or fitness equipment during sleep
- \* User has synced the device with the app

Test Data: No test data needed

Test Steps:

1. Enable sleep tracking feature in the app
2. Wear the wearable device or fitness equipment during sleep

3. Sync the device with the app in the morning
4. Check the app for daily, weekly, and monthly summaries of sleep data
5. Verify that sleep data is displayed in each summary

Postconditions:

- \* Sleep data is displayed in daily, weekly, and monthly summaries
- \* Summaries display sleep duration, quality, and other relevant metrics

Expected Result: The app displays sleep data in daily, weekly, and monthly summaries, allowing users to track their sleep patterns over time.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 81**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

## Workout Logging

### Feature description -

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

### Userstory title -

As a user, I want to log my workouts so that I can track my progress and improve my performance.

### Acceptance criteria -

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

### Generated Test Cases:

Here are five detailed test cases for the Workout Logging feature of the Fitness Tracker App:

#### **\*\*Test Case 1: Logging a New Workout\*\***

Test Case ID: WKT-001

Test Case Title: Verify that users can log a new workout with details.

Test Case Description: This test case ensures that users can create a new workout log entry with exercise type, duration, intensity, and notes.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

\* User is logged in to the app

\* User has access to the Workout Logging feature

- \* There are no existing workout logs for the current user

#### Test Data:

- \* Exercise type: "Running"

- \* Duration: 30 minutes

- \* Intensity: Medium

- \* Notes: "First run of the week"

#### Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Click the "Add New Workout" button
3. Enter the exercise type, duration, intensity, and notes
4. Click the "Save" button
5. Verify that the new workout log entry is displayed in the workout log list

#### Postconditions:

- \* The new workout log entry is saved in the database

- \* The workout log list is updated to display the new entry

Expected Result: The app successfully creates a new workout log entry with the provided details.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Editing an Existing Workout\*\***

Test Case ID: WKT-002

Test Case Title: Verify that users can edit an existing workout log entry.

Test Case Description: This test case ensures that users can edit an existing workout log entry and update the exercise type, duration, intensity, and notes.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has access to the Workout Logging feature
- \* There is an existing workout log entry for the current user

Test Data:

- \* Existing workout log entry: "Walking" exercise type, 20 minutes duration, Low intensity, "Morning walk" notes
- \* Updated exercise type: "Jogging"
- \* Updated duration: 25 minutes
- \* Updated intensity: Medium
- \* Updated notes: "Morning jog"

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Select an existing workout log entry
3. Click the "Edit" button
4. Update the exercise type, duration, intensity, and notes
5. Click the "Save" button
6. Verify that the updated workout log entry is displayed in the workout log list

Postconditions:

- \* The updated workout log entry is saved in the database
- \* The workout log list is updated to display the updated entry

Expected Result: The app successfully updates the existing workout log entry with the provided changes.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Deleting a Workout Log Entry\*\***

Test Case ID: WKT-003

Test Case Title: Verify that users can delete a workout log entry.

Test Case Description: This test case ensures that users can delete a workout log entry and remove it from the workout log list.

Test Suite: Workout Logging

Test Priority: Medium



Preconditions:

- \* User is logged in to the app
- \* User has access to the Workout Logging feature
- \* There is an existing workout log entry for the current user

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Select an existing workout log entry
3. Click the "Delete" button
4. Confirm the deletion prompt
5. Verify that the workout log entry is removed from the workout log list

Postconditions:

- \* The workout log entry is removed from the database
- \* The workout log list is updated to remove the deleted entry

Expected Result: The app successfully deletes the selected workout log entry.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

## **\*\*Test Case 4: Filtering Workout Logs by Date\*\***

Test Case ID: WKT-004

Test Case Title: Verify that users can filter workout logs by date.

Test Case Description: This test case ensures that users can filter workout logs by date and view only the logs that match the selected date range.

Test Suite: Workout Logging

Test Priority: Low

Preconditions:

- \* User is logged in to the app
- \* User has access to the Workout Logging feature
- \* There are multiple workout log entries for the current user with different dates

Test Data:

- \* Date range: Yesterday and today
- \* Expected workout log entries: 2 entries, one from yesterday and one from today

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Select the date range filter option
3. Choose a date range (e.g., yesterday and today)
4. Verify that only the workout log entries that match the selected date range are displayed
5. Verify that the workout log list is updated to display only the filtered entries

Postconditions:

- \* The workout log list is updated to display only the filtered entries
- \* The date range filter option is reset to default

Expected Result: The app successfully filters workout logs by date and displays only the relevant entries.

Severity: Trivial

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Workout Log Data Storage and Retrieval\*\***

Test Case ID: WKT-005

Test Case Title: Verify that workout log data is stored and retrievable.

Test Case Description: This test case ensures that workout log data is stored in the database and can be retrieved and displayed in the workout log list.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- \* User is logged in to the app
- \* User has access to the Workout Logging feature
- \* There are existing workout log entries for the current user

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Verify that the workout log list displays existing workout log entries
3. Select an existing workout log entry
4. Verify that the workout log entry details are displayed correctly
5. Verify that the workout log list is updated to display the selected entry

Postconditions:

- \* The workout log data is stored in the database
- \* The workout log list is updated to display the selected entry

Expected Result: The app successfully stores and retrieves workout log data and displays it in the workout log list.

Severity: Blocker

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

-----

**Data - 82**

**Project name -**

## Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Goal Setting

### **Feature description -**

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

### **Userstory title -**

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

### **Acceptance criteria -**

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

### **Generated Test Cases:**

Here are five detailed test cases for the Goal Setting feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can set goals for various fitness metrics\*\***

Test Case ID: FT-001

Test Case Title: Verify that users can set goals for various fitness metrics

Test Case Description: This test case ensures that users can set goals for different fitness metrics such as steps

taken, calories burned, distance traveled, and heart rate.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- \* User is logged in to the app

- \* User has access to the goal setting feature

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the goal setting feature
2. Select a fitness metric (e.g., steps taken) and click on "Set Goal"
3. Enter a goal value and select a goal type (e.g., daily, weekly, monthly)
4. Click "Save Goal"
5. Verify that the goal is saved and displayed in the goal list

Postconditions:

- \* The goal is saved and displayed in the goal list

- \* The goal value and type are correctly displayed

Expected Result: The user can set goals for various fitness metrics, and the goal is saved and displayed correctly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that progress tracking is available for each goal\*\***

Test Case ID: FT-002

Test Case Title: Verify that progress tracking is available for each goal

Test Case Description: This test case ensures that the app provides progress tracking for each goal set by the user.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- \* User has set a goal for a fitness metric

- \* User has performed an activity that contributes to the goal (e.g., taken steps)

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the goal setting feature
2. Select a goal and click on "View Progress"
3. Verify that the progress tracking graph/chart is displayed
4. Verify that the current progress is correctly calculated and displayed
5. Verify that the remaining progress to achieve the goal is correctly calculated and displayed

Postconditions:

- \* The progress tracking graph/chart is displayed

- \* The current progress is correctly calculated and displayed

- \* The remaining progress to achieve the goal is correctly calculated and displayed

Expected Result: The app provides progress tracking for each goal, and the current progress and remaining progress are correctly calculated and displayed.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users receive motivational feedback and reminders\*\***

Test Case ID: FT-003

Test Case Title: Verify that users receive motivational feedback and reminders

Test Case Description: This test case ensures that the app provides motivational feedback and reminders to users to help them stay focused and committed to achieving their goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- \* User has set a goal for a fitness metric

- \* User has performed an activity that contributes to the goal (e.g., taken steps)

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the goal setting feature



2. Select a goal and click on "View Progress"
3. Verify that motivational feedback is displayed (e.g., "You're getting closer to your goal!")
4. Verify that reminders are sent to the user at regular intervals (e.g., daily, weekly)
5. Verify that the reminders are correctly displayed in the app's notification center

Postconditions:

- \* Motivational feedback is displayed
- \* Reminders are sent to the user at regular intervals
- \* Reminders are correctly displayed in the app's notification center

Expected Result: The app provides motivational feedback and reminders to users to help them stay focused and committed to achieving their goals.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that goals can be adjusted or updated as needed\*\***

Test Case ID: FT-004

Test Case Title: Verify that goals can be adjusted or updated as needed

Test Case Description: This test case ensures that users can adjust or update their goals as needed.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

\* User has set a goal for a fitness metric

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the goal setting feature
2. Select a goal and click on "Edit Goal"
3. Make changes to the goal value or type
4. Click "Save Changes"
5. Verify that the goal is updated correctly

Postconditions:

- \* The goal is updated correctly
- \* The updated goal is displayed in the goal list

Expected Result: Users can adjust or update their goals as needed, and the changes are correctly saved and reflected in the goal list.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the app provides a summary of goal attainment\*\***

Test Case ID: FT-005

Test Case Title: Verify that the app provides a summary of goal attainment

Test Case Description: This test case ensures that the app provides a summary of goal attainment to users.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- \* User has set multiple goals for different fitness metrics
- \* User has performed activities that contribute to the goals (e.g., taken steps, burned calories)

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the goal setting feature
2. Click on "Goal Summary"
3. Verify that a summary of goal attainment is displayed (e.g., "You've achieved 3 out of 5 goals this week")
4. Verify that the summary includes a breakdown of progress for each goal
5. Verify that the summary is correctly calculated and displayed

Postconditions:

- \* The goal summary is displayed
- \* The summary includes a breakdown of progress for each goal
- \* The summary is correctly calculated and displayed

Expected Result: The app provides a summary of goal attainment, including a breakdown of progress for each goal, and the summary is correctly calculated and displayed.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 83**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Workout Plans

**Feature description -**

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

**Userstory title -**

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

**Acceptance criteria -**

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

**Generated Test Cases:**

Here are five detailed test cases for the Workout Plans feature of the Fitness Tracker App, using the provided Test Case Creation Template:

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

Test Case ID: WKP-001

Test Case Title: Verify that users can select workout plans based on fitness levels.

Test Case Description: This test case ensures that users can choose workout plans corresponding to their fitness levels.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- \* The user has a registered account and is logged in to the app.
- \* The user has completed a fitness assessment to determine their fitness level.

Test Data: No test data needed

Test Steps:

1. Log in to the app as a registered user.
2. Navigate to the Workout Plans section.
3. Select a fitness level (e.g., beginner, intermediate, advanced).
4. Verify that the app displays a list of workout plans suitable for the selected fitness level.
5. Select a workout plan from the list.

Postconditions:

- \* The selected workout plan is displayed with its details (e.g., name, description, exercises, schedule).

Expected Result: The app allows users to select workout plans based on their fitness levels.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: WKP-002

Test Case Title: Verify that workout plans include detailed exercise routines and schedules.

Test Case Description: This test case ensures that workout plans provide detailed information about exercises and schedules.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- \* The user has selected a workout plan.
- \* The workout plan is displayed with its details.

Test Data: No test data needed

Test Steps:

1. Review the selected workout plan's details.
2. Verify that the plan includes a list of exercises with descriptions, sets, reps, and weights.
3. Verify that the plan includes a schedule with workout days, times, and durations.
4. Verify that the plan provides instructions for each exercise.

Postconditions:

\* The workout plan's details are updated in the user's profile.

Expected Result: Workout plans include detailed exercise routines and schedules.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: WKP-003

Test Case Title: Verify that users can track adherence and progress within the workout plan.

Test Case Description: This test case ensures that users can log their workouts and track their progress within the selected workout plan.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

\* The user has selected a workout plan.

\* The workout plan is displayed with its details.

Test Data: No test data needed

Test Steps:

1. Log a workout for a particular day and exercise.
2. Verify that the app records the workout and updates the user's progress.
3. Verify that the app displays a dashboard with progress metrics (e.g., completed workouts, remaining workouts, progress percentage).

4. Verify that the app provides feedback and suggestions based on the user's progress.

Postconditions:

- \* The user's progress is updated in their profile.

Expected Result: Users can track adherence and progress within the workout plan.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: WKP-004

Test Case Title: Verify that the app provides notifications and reminders for workouts.

Test Case Description: This test case ensures that the app sends notifications and reminders to users about their upcoming workouts.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- \* The user has selected a workout plan.

- \* The workout plan includes a schedule with upcoming workouts.

Test Data: No test data needed

Test Steps:

1. Set up notifications and reminders for upcoming workouts.



2. Verify that the app sends notifications and reminders at the scheduled times.
3. Verify that the notifications and reminders include relevant information (e.g., workout name, exercise, time).

Postconditions:

\* The user receives notifications and reminders for their upcoming workouts.

Expected Result: The app provides notifications and reminders for workouts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

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

Test Case ID: WKP-005

Test Case Title: Verify that users can adjust plans based on progress and preferences.

Test Case Description: This test case ensures that users can modify their workout plans based on their progress and preferences.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

\* The user has selected a workout plan.

\* The user has logged workouts and tracked their progress.

Test Data: No test data needed

Test Steps:

1. Access the workout plan's settings.
2. Modify the workout plan's settings (e.g., change the exercise, adjust the schedule).
3. Verify that the app updates the workout plan accordingly.
4. Verify that the app provides recommendations for adjustments based on the user's progress and preferences.

Postconditions:

\* The user's workout plan is updated in their profile.

Expected Result: Users can adjust plans based on progress and preferences.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 84**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Community Support

### **Feature description -**

Users can connect with a community of like-minded individuals, share progress updates, participate in challenges, and offer support and encouragement to fellow users. Community support fosters accountability, motivation, and a sense of camaraderie among users striving towards similar fitness goals.

### **Userstory title -**

As a user, I want to connect with a community so that I can share my progress and stay motivated.

### **Acceptance criteria -**

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.', 'The app facilitates participation in community challenges.', 'Users can offer and receive support and encouragement.', 'Community activity and engagement are tracked.']

### **Generated Test Cases:**

Here are five detailed test cases for the Community Support feature of the Fitness Tracker App, using the provided Test Case Creation Template:

**\*\*Test Case 1: Verify that users can join and participate in community groups\*\***

Test Case ID: CT001

Test Case Title: Verify that users can join and participate in community groups

Test Case Description: This test case verifies that users can successfully join and participate in community groups, enabling them to connect with like-minded individuals and share their fitness journey.

Test Suite: Community Support

Test Priority: High

Preconditions:

\* User is logged in to the app

\* Community groups are available and accessible

Test Data: No test data needed

Test Steps:

1. Navigate to the Community tab
2. Search for a community group (e.g., "Running Enthusiasts")
3. Click on the "Join" button for the selected group
4. Verify that the user is added to the group and can view group members and posts
5. Post a comment or update in the group to verify participation

Postconditions:

- \* User is added to the community group
- \* User can view group members and posts
- \* User can post comments or updates in the group

Expected Result: User can successfully join and participate in community groups

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can share progress updates and achievements\*\***

Test Case ID: CT002

Test Case Title: Verify that users can share progress updates and achievements

Test Case Description: This test case verifies that users can share their progress updates and achievements

with the community, fostering accountability and motivation.

Test Suite: Community Support

Test Priority: High

Preconditions:

- \* User has logged at least one workout or activity

- \* User is a member of a community group

Test Data: Sample workout data (e.g., 5km run, 30-minute yoga session)

Test Steps:

1. Navigate to the Community tab
2. Click on the "Share Update" button
3. Select the workout or activity to share
4. Add a description or comment to the update
5. Click on the "Post" button to share the update with the community

Postconditions:

- \* Update is posted successfully

- \* Update is visible to community group members

- \* Update includes user's name, workout details, and optional comment

Expected Result: User can successfully share progress updates and achievements with the community

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the app facilitates participation in community challenges\*\***

Test Case ID: CT003

Test Case Title: Verify that the app facilitates participation in community challenges

Test Case Description: This test case verifies that the app enables users to participate in community challenges, promoting healthy competition and motivation among users.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- \* A community challenge is available and active
- \* User is a member of the community group hosting the challenge

Test Data: Sample challenge details (e.g., "Summer Running Challenge", "Walk 10,000 steps daily for 7 days")

Test Steps:

1. Navigate to the Community tab
2. Click on the "Challenges" section
3. Select the available challenge to participate in
4. Click on the "Join" button to participate in the challenge
5. Verify that user is added to the challenge participants list

Postconditions:

- \* User is added to the challenge participants list
- \* User can view challenge details and rules
- \* User can track progress and compete with other participants

Expected Result: User can successfully participate in community challenges

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users can offer and receive support and encouragement\*\***

Test Case ID: CT004

Test Case Title: Verify that users can offer and receive support and encouragement

Test Case Description: This test case verifies that users can offer and receive support and encouragement from fellow community members, fostering a sense of camaraderie and motivation.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- \* User is a member of a community group
- \* Another user has posted an update or achievement

Test Data: Sample update or achievement (e.g., "just completed my first 5K!")

Test Steps:

1. Navigate to the Community tab

2. View the update or achievement posted by another user
3. Click on the "Comment" or "Like" button to offer support and encouragement
4. Verify that the comment or like is successfully posted
5. Verify that the original poster receives a notification about the comment or like

Postconditions:

- \* User can offer support and encouragement to fellow community members
- \* User receives notifications about comments or likes on their updates
- \* User can view comments and likes on their updates

Expected Result: User can successfully offer and receive support and encouragement

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that community activity and engagement are tracked\*\***

Test Case ID: CT005

Test Case Title: Verify that community activity and engagement are tracked

Test Case Description: This test case verifies that the app tracks community activity and engagement, providing insights into user behavior and community health.

Test Suite: Community Support

Test Priority: Low

Preconditions:



\* Community groups are available and accessible

\* Users have participated in community activities (e.g., posted updates, commented on posts)

Test Data: No test data needed

Test Steps:

1. Navigate to the Community tab
2. Click on the "Insights" or "Analytics" section
3. Verify that community activity and engagement metrics are displayed (e.g., number of posts, comments, likes)
4. Verify that user engagement metrics are displayed (e.g., most active users, top contributors)

Postconditions:

\* Community activity and engagement metrics are displayed

\* User engagement metrics are displayed

Expected Result: Community activity and engagement are successfully tracked

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 85**

**Project name -**

## Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Integration with Wearable Devices

### **Feature description -**

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

### **Userstory title -**

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

### **Acceptance criteria -**

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

### **Generated Test Cases:**

Here are the test cases for the feature "Integration with Wearable Devices":

**\*\*Test Case 1: Verify that the app supports integration with multiple wearable devices\*\***

Test Case ID: FT-IWD-001

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: This test case ensures that the app supports integration with multiple wearable devices,

allowing users to connect and sync data from various devices.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- \* The app is installed and running on a compatible device.
- \* Multiple wearable devices (e.g., fitness trackers, smartwatches, heart rate monitors) are available for testing.

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the wearable device integration settings.
2. Select multiple wearable devices to integrate (e.g., Fitbit, Apple Watch, Garmin).
3. Authenticate and authorize each device for integration.
4. Verify that all selected devices are successfully integrated with the app.

Postconditions:

- \* The app should display a list of integrated wearable devices.
- \* Each device should be listed with its corresponding settings and data sync status.

Expected Result: The app successfully integrates with multiple wearable devices, and each device is listed with its settings and data sync status.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can sync activity data from their wearables\*\***

Test Case ID: FT-IWD-002

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: This test case ensures that users can sync activity data from their wearable devices to the app, allowing them to track their fitness activities.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- \* The app is installed and running on a compatible device.
- \* A wearable device is paired and integrated with the app.

Test Data: No test data needed

Test Steps:

1. Perform a workout or physical activity using the wearable device.
2. Launch the app and navigate to the wearable device integration settings.
3. Trigger a data sync from the wearable device to the app.
4. Verify that the app receives and displays the synced activity data (e.g., steps taken, calories burned, distance traveled).

Postconditions:

- \* The app should display the synced activity data from the wearable device.

Expected Result: The app successfully syncs activity data from the wearable device, and the data is displayed correctly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the app provides real-time data updates from wearables\*\***

Test Case ID: FT-IWD-003

Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: This test case ensures that the app receives and displays real-time data updates from wearable devices, providing users with up-to-date fitness tracking information.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- \* The app is installed and running on a compatible device.
- \* A wearable device is paired and integrated with the app.

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the wearable device integration settings.
2. Perform a physical activity using the wearable device (e.g., take a few steps).
3. Verify that the app receives and displays the updated activity data in real-time.
4. Repeat step 2 and 3 multiple times to ensure consistent real-time updates.

Postconditions:

- \* The app should display real-time updates of activity data from the wearable device.

Expected Result: The app successfully receives and displays real-time data updates from the wearable device.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that wearable data includes steps, heart rate, and calories burned\*\***

Test Case ID: FT-IWD-004

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: This test case ensures that the app receives and displays the expected wearable device data, including steps taken, heart rate, and calories burned.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions:

- \* The app is installed and running on a compatible device.

- \* A wearable device is paired and integrated with the app.

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the wearable device integration settings.
2. Trigger a data sync from the wearable device to the app.
3. Verify that the app displays the following wearable device data:
  - \* Steps taken
  - \* Heart rate
  - \* Calories burned
4. Verify that the data is displayed correctly and accurately.

Postconditions:

- \* The app should display the expected wearable device data.

Expected Result: The app successfully receives and displays the expected wearable device data, including steps taken, heart rate, and calories burned.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can troubleshoot connectivity issues with wearables\*\***

Test Case ID: FT-IWD-005

Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: This test case ensures that the app provides adequate troubleshooting options for users experiencing connectivity issues with their wearable devices.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions:

- \* The app is installed and running on a compatible device.
- \* A wearable device is paired but not integrated with the app (e.g., Bluetooth connection issues).

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the wearable device integration settings.
2. Attempt to connect the wearable device to the app.
3. Simulate a connectivity issue (e.g., Bluetooth connection failure).
4. Verify that the app provides troubleshooting options or error messages to assist the user in resolving the issue.

Postconditions:

- \* The app should display troubleshooting options or error messages to assist the user.

Expected Result: The app provides adequate troubleshooting options or error messages to assist the user in resolving connectivity issues with their wearable devices.

Severity: Minor

Priority: Medium

Type of Testing: Functional 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:**

Here are five detailed test cases for the Water Intake Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can log their daily water intake\*\***

Test Case ID: FT-WIT-001

Test Case Title: Verify that users can log their daily water intake

Test Case Description: This test case ensures that users can successfully log their daily water intake in the app.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- \* The user is logged in to the app
- \* The user has access to the water intake tracking feature

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the water intake tracking feature
2. Enter a valid water intake amount (e.g., 8 oz)
3. Tap the "Log Water Intake" button
4. Verify that the water intake amount is successfully logged and displayed on the dashboard

Postconditions:

- \* The user's water intake history is updated
- \* The dashboard displays the updated water intake amount

Expected Result: The user can successfully log their daily water intake, and the app accurately records and displays the data.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the app calculates total water consumption for the day\*\***

Test Case ID: FT-WIT-002

Test Case Title: Verify that the app calculates total water consumption for the day

Test Case Description: This test case ensures that the app accurately calculates the total water consumption for the day based on user-inputted water intake amounts.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- \* The user is logged in to the app
- \* The user has access to the water intake tracking feature
- \* The user has logged multiple water intake amounts throughout the day

Test Data:

- \* Multiple water intake amounts (e.g., 8 oz, 16 oz, 24 oz)

Test Steps:

1. Launch the app and navigate to the water intake tracking feature
2. Log multiple water intake amounts throughout the day
3. Verify that the app calculates the total water consumption for the day
4. Compare the calculated total water consumption with the expected result

Postconditions:

\* The dashboard displays the calculated total water consumption for the day

Expected Result: The app accurately calculates the total water consumption for the day based on user-inputted water intake amounts.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can set daily hydration goals\*\***

Test Case ID: FT-WIT-003

Test Case Title: Verify that users can set daily hydration goals

Test Case Description: This test case ensures that users can set daily hydration goals, and the app stores and displays the goal correctly.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

\* The user is logged in to the app

\* The user has access to the water intake tracking feature

Test Data:

\* A daily hydration goal (e.g., 64 oz)

### Test Steps:

1. Launch the app and navigate to the water intake tracking feature
2. Tap the "Set Hydration Goal" button
3. Enter a valid daily hydration goal
4. Tap the "Save" button
5. Verify that the app stores and displays the daily hydration goal correctly

### Postconditions:

- \* The user's daily hydration goal is updated
- \* The dashboard displays the updated daily hydration goal

Expected Result: The user can set a daily hydration goal, and the app accurately stores and displays the goal.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that reminders are provided to encourage regular water intake\*\***

Test Case ID: FT-WIT-004

Test Case Title: Verify that reminders are provided to encourage regular water intake

Test Case Description: This test case ensures that the app provides reminders to encourage regular water intake, and users can customize the reminder settings.

Test Suite: Water Intake Tracking

Test Priority: Medium

#### Preconditions:

- \* The user is logged in to the app
- \* The user has access to the water intake tracking feature
- \* The user has set a daily hydration goal

#### Test Data:

- \* Reminder settings (e.g., notification frequency, time of day)

#### Test Steps:

1. Launch the app and navigate to the water intake tracking feature
2. Tap the "Reminder Settings" button
3. Configure the reminder settings (e.g., set notification frequency to "Every 2 hours")
4. Verify that the app sends reminders to drink water at the configured frequency
5. Verify that users can customize the reminder settings

#### Postconditions:

- \* The user receives reminders to drink water at the configured frequency
- \* The reminder settings are updated

Expected Result: The app provides reminders to encourage regular water intake, and users can customize the reminder settings.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can view hydration progress and trends\*\***

Test Case ID: FT-WIT-005

Test Case Title: Verify that users can view hydration progress and trends

Test Case Description: This test case ensures that users can view their hydration progress and trends, and the app displays the data accurately.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- \* The user is logged in to the app
- \* The user has access to the water intake tracking feature
- \* The user has logged multiple water intake amounts

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the water intake tracking feature
2. Verify that the app displays the user's hydration progress and trends
3. Verify that the app displays the data accurately, including:
  - \* Total water consumption for the day
  - \* Daily hydration goal progress
  - \* Hydration trend over time (e.g., last 7 days, last 30 days)

Postconditions:

- \* The user can view their hydration progress and trends
- \* The app displays the data accurately

Expected Result: The user can view their hydration progress and trends, and the app displays the data accurately.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 87**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Nutrition Tracking

**Feature description -**

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food



choices, and helps users align their nutrition with their fitness goals.

### **Userstory title -**

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

### **Acceptance criteria -**

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

### **Generated Test Cases:**

Here are five detailed test cases for the Nutrition Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can log their daily meals and snacks\*\***

Test Case ID: NT-01

Test Case Title: Verify that users can log their daily meals and snacks

Test Case Description: Verify that the app allows users to log their daily meals and snacks, including breakfast, lunch, dinner, and snacks.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- \* User is logged in to the app
- \* User has access to the food diary feature
- \* No meals or snacks are previously logged for the current day

Test Data: No test data needed

## Test Steps:

1. Launch the app and navigate to the food diary feature
2. Click on the "Add Meal" button
3. Enter meal details (e.g., breakfast, lunch, dinner, snack) and food item (e.g., apple, sandwich, pasta)
4. Click on the "Save" button
5. Verify that the meal is successfully logged and displayed in the food diary

## Postconditions:

- \* Meal is logged and displayed in the food diary
- \* Meal details are correct (e.g., meal type, food item)

Expected Result: The app allows users to log their daily meals and snacks successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the app calculates macronutrient intake (carbs, fats, proteins)\*\***

Test Case ID: NT-02

Test Case Title: Verify that the app calculates macronutrient intake (carbs, fats, proteins)

Test Case Description: Verify that the app accurately calculates the macronutrient intake (carbs, fats, proteins) based on the user's logged meals and snacks.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- \* User has logged meals and snacks for the current day
- \* Meal and snack details include macronutrient information (e.g., carbs, fats, proteins)

Test Data: Sample meal and snack data with macronutrient information

Test Steps:

1. Launch the app and navigate to the nutrition tracking feature
2. Verify that the app displays the calculated macronutrient intake (carbs, fats, proteins) for the current day
3. Compare the calculated macronutrient intake with the expected values based on the logged meals and snacks

Postconditions:

- \* Macronutrient intake is calculated accurately
- \* Macronutrient intake is displayed correctly in the nutrition tracking feature

Expected Result: The app accurately calculates the macronutrient intake (carbs, fats, proteins) based on the user's logged meals and snacks.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can set nutritional goals and track progress\*\***

Test Case ID: NT-03

Test Case Title: Verify that users can set nutritional goals and track progress

Test Case Description: Verify that the app allows users to set nutritional goals (e.g., daily caloric intake, macronutrient targets) and track their progress towards achieving those goals.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has access to the nutrition tracking feature

Test Data: Sample nutritional goal data (e.g., daily caloric intake, macronutrient targets)

Test Steps:

1. Launch the app and navigate to the nutrition tracking feature
2. Click on the "Set Goals" button
3. Enter nutritional goal details (e.g., daily caloric intake, macronutrient targets)
4. Click on the "Save" button
5. Verify that the app displays the user's progress towards achieving their nutritional goals

Postconditions:

- \* Nutritional goals are set successfully

\* Progress towards achieving nutritional goals is tracked and displayed correctly

Expected Result: The app allows users to set nutritional goals and track their progress successfully.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the app provides nutritional information and suggestions\*\***

Test Case ID: NT-04

Test Case Title: Verify that the app provides nutritional information and suggestions

Test Case Description: Verify that the app provides users with nutritional information and suggestions based on their logged meals and snacks, and nutritional goals.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

\* User has logged meals and snacks for the current day

\* User has set nutritional goals

Test Data: Sample meal and snack data with nutritional information

Test Steps:

1. Launch the app and navigate to the nutrition tracking feature

2. Verify that the app displays nutritional information (e.g., calorie intake, macronutrient breakdown) for each meal and snack
3. Verify that the app provides suggestions for improving nutritional balance and achieving nutritional goals

Postconditions:

- \* Nutritional information is displayed correctly for each meal and snack
- \* Suggestions for improving nutritional balance and achieving nutritional goals are provided

Expected Result: The app provides users with nutritional information and suggestions based on their logged meals and snacks, and nutritional goals.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can view their nutritional history and trends\*\***

Test Case ID: NT-05

Test Case Title: Verify that users can view their nutritional history and trends

Test Case Description: Verify that the app allows users to view their nutritional history and trends, including daily, weekly, and monthly summaries of their macronutrient intake and progress towards achieving nutritional goals.

Test Suite: Nutrition Tracking

Test Priority: Low

Preconditions:

\* User has logged meals and snacks for at least one week

\* User has set nutritional goals

Test Data: Sample meal and snack data with nutritional information for at least one week

Test Steps:

1. Launch the app and navigate to the nutrition tracking feature
2. Click on the "History" button
3. Verify that the app displays daily, weekly, and monthly summaries of the user's macronutrient intake
4. Verify that the app displays progress towards achieving nutritional goals over time

Postconditions:

\* Nutritional history and trends are displayed correctly

\* Progress towards achieving nutritional goals is displayed correctly over time

Expected Result: The app allows users to view their nutritional history and trends successfully.

Severity: Minor

Priority: Low

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 -**

Progress Reports

**Feature description -**

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

**Userstory title -**

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

**Acceptance criteria -**

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

**Generated Test Cases:**

Here are five detailed test cases for the Progress Reports feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that weekly progress reports are generated correctly\*\***

Test Case ID: TR-PR-001



Test Case Title: Verify that weekly progress reports are generated correctly

Test Case Description: This test case ensures that the app generates weekly progress reports accurately, including activity levels, workout performance, and goal attainment.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

- \* User has tracked activity data for at least one week

- \* User has set goals and workout routines in the app

Test Data: No test data needed

Test Steps:

1. Log in to the app with a test user account
2. Navigate to the Progress Reports section
3. Verify that a weekly progress report is generated and displayed
4. Check that the report includes activity levels, workout performance, and goal attainment data
5. Verify that the report is accurate and up-to-date

Postconditions:

- \* Weekly progress report is generated and displayed

- \* Report data is accurate and up-to-date

Expected Result: The app generates a weekly progress report correctly, including activity levels, workout performance, and goal attainment data.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that monthly progress reports include detailed breakdowns of achievements and trends\*\***

Test Case ID: TR-PR-002

Test Case Title: Verify that monthly progress reports include detailed breakdowns of achievements and trends

Test Case Description: This test case ensures that the app provides detailed breakdowns of achievements and trends in monthly progress reports, empowering users to make informed decisions.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

\* User has tracked activity data for at least one month

\* User has set goals and workout routines in the app

Test Data: No test data needed

Test Steps:

1. Log in to the app with a test user account
2. Navigate to the Progress Reports section
3. Verify that a monthly progress report is generated and displayed
4. Check that the report includes detailed breakdowns of achievements and trends
5. Verify that the breakdowns are accurate and relevant to the user's fitness goals

Postconditions:

- \* Monthly progress report is generated and displayed
- \* Report includes detailed breakdowns of achievements and trends

Expected Result: The app generates a monthly progress report that includes detailed breakdowns of achievements and trends, empowering users to make informed decisions.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can customize the frequency and content of progress reports\*\***

Test Case ID: TR-PR-003

Test Case Title: Verify that users can customize the frequency and content of progress reports

Test Case Description: This test case ensures that users can customize the frequency and content of progress reports to suit their fitness goals and preferences.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- \* User has logged in to the app with a test user account
- \* User has set goals and workout routines in the app

Test Data: No test data needed

Test Steps:

1. Log in to the app with a test user account
2. Navigate to the Progress Reports section
3. Verify that the app allows users to customize the frequency and content of progress reports
4. Change the report frequency to daily and add/remove certain metrics (e.g., calories burned, distance traveled)
5. Verify that the app reflects the changes in the next generated report

Postconditions:

- \* User can customize the frequency and content of progress reports
- \* Changes are reflected in the next generated report

Expected Result: The app allows users to customize the frequency and content of progress reports, and reflects the changes in the next generated report.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that progress reports provide recommendations based on report findings\*\***

Test Case ID: TR-PR-004

Test Case Title: Verify that progress reports provide recommendations based on report findings

Test Case Description: This test case ensures that the app provides personalized recommendations in progress reports, helping users to make informed decisions and adjustments to their fitness routines.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

- \* User has tracked activity data for at least one week
- \* User has set goals and workout routines in the app

Test Data: No test data needed

Test Steps:

1. Log in to the app with a test user account
2. Navigate to the Progress Reports section
3. Verify that the app provides personalized recommendations in the progress report
4. Check that the recommendations are relevant to the user's fitness goals and report findings
5. Verify that the recommendations are actionable and easy to understand

Postconditions:

- \* Progress report includes personalized recommendations
- \* Recommendations are relevant and actionable

Expected Result: The app provides personalized recommendations in progress reports, helping users to make informed decisions and adjustments to their fitness routines.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that progress reports are displayed correctly on different devices and platforms\*\***

Test Case ID: TR-PR-005

Test Case Title: Verify that progress reports are displayed correctly on different devices and platforms

Test Case Description: This test case ensures that progress reports are displayed correctly and consistently across different devices and platforms, ensuring a seamless user experience.

Test Suite: Progress Reports

Test Priority: Low

Preconditions:

- \* User has logged in to the app with a test user account
- \* User has set goals and workout routines in the app

Test Data: No test data needed

Test Steps:

1. Log in to the app with a test user account on different devices (e.g., iOS, Android, web)
2. Navigate to the Progress Reports section on each device
3. Verify that the progress report is displayed correctly and consistently across all devices
4. Check that the report layout, formatting, and content are identical across all devices
5. Verify that the app provides a seamless user experience across different devices and platforms

Postconditions:

- \* Progress reports are displayed correctly on different devices and platforms
- \* App provides a seamless user experience across different devices and platforms

Expected Result: The app displays progress reports correctly and consistently across different devices and platforms, ensuring a seamless user experience.

Severity: Minor

Priority: Low

Type of Testing: Compatibility Testing

Test Case Approach: Positive

-----

**Data - 89**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

In-App Challenges

**Feature description -**

The app offers in-app challenges and competitions, encouraging users to set personal records, compete with friends, and earn rewards for achieving fitness milestones. In-app challenges add an element of fun, motivation, and friendly competition to the fitness tracking experience.

**Userstory title -**

As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals.

**Acceptance criteria -**

['Users can join various in-app challenges.', 'Challenges have clear objectives and rules.', 'Users can track their

progress within the challenge.', 'Rewards are provided for achieving challenge milestones.', 'Users can compete with friends and view leaderboards.']

## **Generated Test Cases:**

Here are five detailed test cases for the In-App Challenges feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can join various in-app challenges\*\***

Test Case ID: FT-001

Test Case Title: Verify that users can join various in-app challenges

Test Case Description: This test case verifies that users can successfully join multiple in-app challenges, ensuring that the feature allows users to participate in different challenges.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- \* User has a registered account on the Fitness Tracker App
- \* User has completed the onboarding process
- \* At least two in-app challenges are available

Test Data: No test data needed

Test Steps:

1. Log in to the Fitness Tracker App as a registered user
2. Navigate to the In-App Challenges section
3. Browse through the available challenges and select two challenges to join
4. Click on the "Join" button for each challenge



5. Verify that the user is successfully added to each challenge

Postconditions:

- \* User is displayed as a participant in each joined challenge

- \* Challenges are updated in the user's challenge dashboard

Expected Result: User can successfully join multiple in-app challenges

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that challenges have clear objectives and rules\*\***

Test Case ID: FT-002

Test Case Title: Verify that challenges have clear objectives and rules

Test Case Description: This test case verifies that each challenge has clear objectives, rules, and requirements, ensuring that users understand what is expected to achieve the challenge.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

- \* User has joined an in-app challenge

Test Data: No test data needed

Test Steps:

1. Navigate to the In-App Challenges section
2. Select a joined challenge and view the challenge details
3. Verify that the challenge has a clear objective, rules, and requirements
4. Verify that the challenge details are easily understandable and well-structured

Postconditions: None

Expected Result: Challenges have clear objectives, rules, and requirements

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can track their progress within the challenge\*\***

Test Case ID: FT-003

Test Case Title: Verify that users can track their progress within the challenge

Test Case Description: This test case verifies that users can view their progress within a challenge, ensuring that users can monitor their progress and stay motivated.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

\* User has joined an in-app challenge

\* User has completed some progress within the challenge

Test Data: No test data needed

## Test Steps:

1. Navigate to the In-App Challenges section
2. Select a joined challenge and view the challenge dashboard
3. Verify that the user's progress is displayed accurately
4. Verify that the progress tracking is updated in real-time

## Postconditions:

- \* User's progress is updated in the challenge dashboard
- \* Progress tracking is accurate and up-to-date

Expected Result: Users can track their progress within the challenge

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that rewards are provided for achieving challenge milestones\*\***

Test Case ID: FT-004

Test Case Title: Verify that rewards are provided for achieving challenge milestones

Test Case Description: This test case verifies that users receive rewards for achieving specific milestones within a challenge, ensuring that users are motivated to complete the challenge.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

- \* User has joined an in-app challenge
- \* User has completed a challenge milestone

Test Data: No test data needed

Test Steps:

1. Navigate to the In-App Challenges section
2. Select a joined challenge and view the challenge dashboard
3. Verify that the user has received a reward for completing the milestone
4. Verify that the reward is displayed in the user's profile or challenge dashboard

Postconditions:

- \* User's reward is displayed in the profile or challenge dashboard
- \* Reward is accurately reflected in the user's account

Expected Result: Users receive rewards for achieving challenge milestones

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can compete with friends and view leaderboards\*\***

Test Case ID: FT-005

Test Case Title: Verify that users can compete with friends and view leaderboards

Test Case Description: This test case verifies that users can compete with friends and view leaderboards, ensuring that users are motivated to compete and stay engaged with the app.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

\* User has joined an in-app challenge with friends

Test Data: No test data needed

Test Steps:

1. Navigate to the In-App Challenges section
2. Select a joined challenge and view the challenge dashboard
3. Verify that the user can view the leaderboard and compete with friends
4. Verify that the leaderboard is updated in real-time

Postconditions:

\* Leaderboard is updated in real-time

\* User can view their ranking and progress compared to friends

Expected Result: Users can compete with friends and view leaderboards

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

---

**Data - 90**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Customizable Dashboards

**Feature description -**

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

**Userstory title -**

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

**Acceptance criteria -**

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

**Generated Test Cases:**

Here are five detailed test cases for the Customizable Dashboards feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can select which metrics to display on their dashboard\*\***

Test Case ID: CD-001

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 the fitness metrics they want to display on their dashboard.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions:

- \* The user is logged in to the app.
- \* The user has access to the dashboard customization feature.

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the dashboard customization page.
2. Select a few metrics (e.g., steps taken, calories burned, distance traveled) to display on the dashboard.
3. Click the "Save" button to save the changes.
4. Verify that the selected metrics are displayed on the dashboard.

Postconditions:

- \* The dashboard displays the selected metrics.
- \* The user can view the metrics in real-time.

Expected Result: The user can successfully select and display their preferred metrics on the dashboard.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the app allows customization of dashboard layout and theme\*\***

Test Case ID: CD-002

Test Case Title: Verify that the app allows customization of dashboard layout and theme

Test Case Description: This test case verifies that users can customize the layout and theme of their dashboard.

Test Suite: Dashboard Customization

Test Priority: Medium

Preconditions:

- \* The user is logged in to the app.
- \* The user has access to the dashboard customization feature.

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the dashboard customization page.
2. Select a different layout (e.g., grid, list, compact) and theme (e.g., light, dark, custom) for the dashboard.
3. Click the "Save" button to save the changes.
4. Verify that the dashboard layout and theme have changed accordingly.

Postconditions:

- \* The dashboard layout and theme have changed as selected by the user.
- \* The dashboard still displays the selected metrics.



Expected Result: The user can successfully customize the layout and theme of their dashboard.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can save and update their dashboard settings\*\***

Test Case ID: CD-003

Test Case Title: Verify that users can save and update their dashboard settings

Test Case Description: This test case verifies that users can save and update their dashboard settings.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions:

\* The user is logged in to the app.

\* The user has access to the dashboard customization feature.

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the dashboard customization page.
2. Select a few metrics and customize the layout and theme of the dashboard.
3. Click the "Save" button to save the changes.
4. Log out of the app and log back in.
5. Verify that the dashboard settings are retained and the selected metrics are displayed.

Postconditions:

- \* The dashboard settings are saved and retained.
- \* The user can update their dashboard settings at any time.

Expected Result: The user can successfully save and update their dashboard settings.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the dashboard displays are dynamic and update in real-time\*\***

Test Case ID: CD-004

Test Case Title: Verify that the dashboard displays are dynamic and update in real-time

Test Case Description: This test case verifies that the dashboard displays update in real-time.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions:

- \* The user is logged in to the app.
- \* The user has access to the dashboard customization feature.

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the dashboard customization page.
2. Select a few metrics (e.g., steps taken, calories burned) to display on the dashboard.

3. Perform an action that updates the metric data (e.g., take a few steps, burn some calories).
4. Verify that the dashboard metrics update in real-time to reflect the changes.

Postconditions:

\* The dashboard displays the updated metrics in real-time.

Expected Result: The dashboard displays update dynamically and in real-time to reflect changes in the user's fitness data.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can reset the dashboard to default settings\*\***

Test Case ID: CD-005

Test Case Title: Verify that users can reset the dashboard to default settings

Test Case Description: This test case verifies that users can reset the dashboard to its default settings.

Test Suite: Dashboard Customization

Test Priority: Medium

Preconditions:

\* The user is logged in to the app.

\* The user has access to the dashboard customization feature.

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the dashboard customization page.
2. Customize the dashboard layout, theme, and metrics.
3. Click the "Reset to Default" button.
4. Verify that the dashboard resets to its default settings.

Postconditions:

- \* The dashboard resets to its default settings.
- \* The user can re-customize the dashboard as needed.

Expected Result: The user can successfully reset the dashboard to its default settings.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

These test cases cover all the acceptance criteria for the Customizable Dashboards feature, ensuring that users can customize their dashboard to display the metrics that matter most to them, with a personalized layout and theme, and that the dashboard updates dynamically in real-time.

-----

**Data - 91**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Social Sharing

**Feature description -**

Users can share workout summaries, achievements, and progress updates on social media platforms, inspiring others and fostering accountability. Social sharing enables users to celebrate their successes, seek support from their social networks, and stay motivated on their fitness journey.

**Userstory title -**

As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

**Acceptance criteria -**

['Users can share workout summaries and achievements.', 'The app supports sharing on multiple social media platforms.', 'Users can customize the content and format of shared posts.', 'Sharing features are integrated seamlessly with social media.', 'Users can track the engagement of their shared posts.']

**Generated Test Cases:**

Here are five detailed test cases for the Social Sharing feature of the Fitness Tracker App, using the provided Test Case Creation Template:

**\*\*Test Case 1: Verify that users can share workout summaries on social media\*\***

Test Case ID: TS-001

Test Case Title: Verify that users can share workout summaries on social media

Test Case Description: Test that users can share a workout summary on social media platforms such as

Facebook, Twitter, and Instagram.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- \* User is logged in to the app
- \* User has completed a workout
- \* Social media accounts are connected to the app

Test Data: No test data needed

Test Steps:

1. Launch the Fitness Tracker App
2. Select a completed workout from the workout log
3. Tap the "Share" button
4. Select the social media platform to share on (e.g., Facebook, Twitter, Instagram)
5. Verify that a workout summary is generated and shared on the selected social media platform

Postconditions:

- \* Verify that the shared post is visible on the user's social media profile
- \* Verify that the post includes relevant workout details (e.g., date, duration, distance traveled)

Expected Result: The user can successfully share a workout summary on social media.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can customize the content and format of shared posts\*\***

Test Case ID: TS-002

Test Case Title: Verify that users can customize the content and format of shared posts

Test Case Description: Test that users can customize the content and format of shared posts on social media platforms.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has completed a workout
- \* Social media accounts are connected to the app

Test Data: No test data needed

Test Steps:

1. Launch the Fitness Tracker App
2. Select a completed workout from the workout log
3. Tap the "Share" button
4. Select the social media platform to share on (e.g., Facebook, Twitter, Instagram)
5. Edit the default post content and format (e.g., add hashtags, change font style)
6. Verify that the customized post is shared on the selected social media platform

Postconditions:

- \* Verify that the customized post is visible on the user's social media profile
- \* Verify that the post includes the customized content and format

Expected Result: The user can successfully customize the content and format of shared posts on social media.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that sharing features are integrated seamlessly with social media\*\***

Test Case ID: TS-003

Test Case Title: Verify that sharing features are integrated seamlessly with social media

Test Case Description: Test that the app's sharing features are integrated seamlessly with social media platforms, without errors or issues.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- \* User is logged in to the app
- \* User has completed a workout
- \* Social media accounts are connected to the app

Test Data: No test data needed

Test Steps:

1. Launch the Fitness Tracker App
2. Select a completed workout from the workout log
3. Tap the "Share" button
4. Select the social media platform to share on (e.g., Facebook, Twitter, Instagram)
5. Verify that the app redirects to the social media platform's authorization page (if necessary)
6. Verify that the sharing process is completed successfully without errors or issues

Postconditions:

- \* Verify that the shared post is visible on the user's social media profile
- \* Verify that the app's sharing features are functioning as expected

Expected Result: The app's sharing features are integrated seamlessly with social media platforms, without errors or issues.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive



**\*\*Test Case 4: Verify that users can track the engagement of their shared posts\*\***

Test Case ID: TS-004

Test Case Title: Verify that users can track the engagement of their shared posts

Test Case Description: Test that users can track the engagement of their shared posts, including likes, comments, and shares.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has completed a workout
- \* Social media accounts are connected to the app
- \* User has shared a post on social media

Test Data: No test data needed

Test Steps:

1. Launch the Fitness Tracker App
2. Select a shared post from the workout log
3. Tap the "Track Engagement" button
4. Verify that the app displays the engagement metrics (e.g., likes, comments, shares)
5. Verify that the engagement metrics are updated in real-time

Postconditions:

- \* Verify that the engagement metrics are accurate and up-to-date
- \* Verify that the app's tracking features are functioning as expected

Expected Result: The user can successfully track the engagement of their shared posts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that the app supports sharing on multiple social media platforms\*\***

Test Case ID: TS-005

Test Case Title: Verify that the app supports sharing on multiple social media platforms

Test Case Description: Test that the app supports sharing on multiple social media platforms, including Facebook, Twitter, Instagram, and LinkedIn.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- \* User is logged in to the app
- \* User has completed a workout
- \* Social media accounts are connected to the app

Test Data: No test data needed

Test Steps:

1. Launch the Fitness Tracker App
2. Select a completed workout from the workout log
3. Tap the "Share" button
4. Verify that the app displays a list of available social media platforms to share on
5. Select each social media platform and verify that the sharing process is completed successfully

Postconditions:

- \* Verify that the shared post is visible on each social media profile
- \* Verify that the app's sharing features are functioning as expected

Expected Result: The app supports sharing on multiple social media platforms, including Facebook, Twitter, Instagram, and LinkedIn.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 92**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Rest and Recovery Tracking

**Feature description -**

The app includes features for tracking rest days, recovery activities, and recovery metrics such as resting heart rate and sleep quality. Rest and recovery tracking helps users prioritize recovery, prevent overtraining, and optimize their overall health and performance.

**Userstory title -**

As a user, I want to track my rest and recovery so that I can optimize my fitness performance and prevent overtraining.

**Acceptance criteria -**

['Users can log rest days and recovery activities.', 'The app tracks metrics such as resting heart rate and sleep quality.', 'Users receive recommendations for recovery based on their activities.', 'The app provides alerts for

insufficient recovery time.', 'Users can view trends and history of their recovery data.']

## **Generated Test Cases:**

Here are five detailed test cases for the Rest and Recovery Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can log rest days and recovery activities\*\***

Test Case ID: FT-001

Test Case Title: Verify that users can log rest days and recovery activities

Test Case Description: This test case verifies that users can successfully log rest days and recovery activities in the app.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- \* The user has a valid account and is logged in to the app.
- \* The user has access to the Rest and Recovery Tracking feature.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the Rest and Recovery Tracking section.
2. Click on the "Log Rest Day" button.
3. Enter a rest day description and select the date.
4. Click on the "Save" button.
5. Verify that the rest day is successfully logged and displayed in the app.

Postconditions:

- \* The rest day is displayed in the app's dashboard.
- \* The rest day is reflected in the user's recovery metrics.

Expected Result: The user can successfully log a rest day and view it in the app.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the app tracks metrics such as resting heart rate and sleep quality\*\***

Test Case ID: FT-002

Test Case Title: Verify that the app tracks metrics such as resting heart rate and sleep quality

Test Case Description: This test case verifies that the app accurately tracks resting heart rate and sleep quality metrics.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- \* The user has a wearable device or fitness equipment connected to the app.
- \* The user has enabled tracking for resting heart rate and sleep quality.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the Rest and Recovery Tracking section.
2. Verify that the app is displaying the user's resting heart rate and sleep quality metrics.
3. Compare the metrics with the actual data from the wearable device or fitness equipment.
4. Verify that the metrics are accurate and up-to-date.

Postconditions:

- \* The resting heart rate and sleep quality metrics are displayed in the app.
- \* The metrics are accurate and reflect the user's actual data.

Expected Result: The app accurately tracks and displays resting heart rate and sleep quality metrics.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users receive recommendations for recovery based on their activities\*\***

Test Case ID: FT-003

Test Case Title: Verify that users receive recommendations for recovery based on their activities

Test Case Description: This test case verifies that the app provides personalized recovery recommendations based on the user's activities.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- \* The user has logged activities and workouts in the app.
- \* The user has enabled personalized recovery recommendations.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the Rest and Recovery Tracking section.
2. Verify that the app is displaying personalized recovery recommendations based on the user's activities.
3. Verify that the recommendations are relevant and tailored to the user's specific needs.

Postconditions:

- \* The app displays personalized recovery recommendations.
- \* The recommendations are relevant and tailored to the user's needs.

Expected Result: The app provides personalized recovery recommendations based on the user's activities.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the app provides alerts for insufficient recovery time\*\***

Test Case ID: FT-004

Test Case Title: Verify that the app provides alerts for insufficient recovery time

Test Case Description: This test case verifies that the app sends alerts to users who have not had sufficient recovery time.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- \* The user has set a recovery goal in the app.
- \* The user has not had sufficient recovery time according to the app's algorithms.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the Rest and Recovery Tracking section.
2. Verify that the app sends an alert to the user when insufficient recovery time is detected.
3. Verify that the alert is clear and actionable.

Postconditions:

- \* The app sends an alert to the user for insufficient recovery time.
- \* The alert is clear and actionable.

Expected Result: The app sends alerts for insufficient recovery time.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive



**\*\*Test Case 5: Verify that users can view trends and history of their recovery data\*\***

Test Case ID: FT-005

Test Case Title: Verify that users can view trends and history of their recovery data

Test Case Description: This test case verifies that users can view trends and history of their recovery data in the app.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- \* The user has logged recovery data in the app.
- \* The user has access to the recovery data trends and history feature.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the Rest and Recovery Tracking section.
2. Click on the "Trends and History" tab.
3. Verify that the app displays trends and history of the user's recovery data.
4. Verify that the data is accurate and up-to-date.

Postconditions:

- \* The app displays trends and history of the user's recovery data.
- \* The data is accurate and up-to-date.

Expected Result: Users can view trends and history of their recovery data.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 93**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Audio Coaching

**Feature description -**

Users can access audio-guided workouts, coaching cues, and motivational messages during their workouts, enhancing engagement and adherence. Audio coaching provides real-time feedback, encouragement, and guidance, simulating the experience of having a personal trainer or workout buddy.

**Userstory title -**

As a user, I want to access audio coaching during workouts so that I can stay motivated and receive real-time guidance.

**Acceptance criteria -**

['Users can access a variety of audio-guided workouts.', 'Audio coaching includes motivational messages and coaching cues.', 'Users can choose coaching based on workout type and intensity.', 'Audio coaching integrates seamlessly with other app features.', 'Users can control audio settings and preferences.']

### **Generated Test Cases:**

Here are five detailed test cases for the Audio Coaching feature in the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can access a variety of audio-guided workouts\*\***

Test Case ID: TC001

Test Case Title: Verify that users can access a variety of audio-guided workouts

Test Case Description: This test case verifies that users can access a range of audio-guided workouts, including cardio, strength training, yoga, and other exercise types.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- \* User is logged in to the app
- \* User has a stable internet connection
- \* Audio coaching feature is enabled

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Click on the Audio Coaching button
3. Select a workout type (e.g., cardio, strength training, yoga)

4. Verify that a list of available audio-guided workouts is displayed
5. Select a workout from the list
6. Verify that the audio coaching begins and provides guidance and cues throughout the workout

Postconditions:

- \* User can resume the workout from where they left off
- \* Workout data is saved and synced with the user's profile

Expected Result: Users can access a variety of audio-guided workouts and receive coaching cues and guidance throughout the workout.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that audio coaching includes motivational messages and coaching cues\*\***

Test Case ID: TC002

Test Case Title: Verify that audio coaching includes motivational messages and coaching cues

Test Case Description: This test case verifies that the audio coaching feature provides motivational messages and coaching cues to keep users engaged and motivated during their workout.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- \* User is logged in to the app

\* User has a stable internet connection

\* Audio coaching feature is enabled

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Click on the Audio Coaching button
3. Select a workout type (e.g., cardio, strength training, yoga)
4. Verify that motivational messages and coaching cues are played during the workout
5. Verify that the audio coaching provides feedback on the user's performance (e.g., pace, distance, calories burned)

Postconditions:

\* User can save and review their workout data

\* Workout data is synced with the user's profile

Expected Result: Audio coaching includes motivational messages and coaching cues to keep users engaged and motivated during their workout.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can choose coaching based on workout type and intensity\*\***

Test Case ID: TC003

Test Case Title: Verify that users can choose coaching based on workout type and intensity

Test Case Description: This test case verifies that users can select audio coaching based on their workout type and intensity level.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has a stable internet connection
- \* Audio coaching feature is enabled

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Click on the Audio Coaching button
3. Select a workout type (e.g., cardio, strength training, yoga)
4. Select an intensity level (e.g., beginner, intermediate, advanced)
5. Verify that the audio coaching is tailored to the selected workout type and intensity level

Postconditions:

- \* User can save and review their workout data
- \* Workout data is synced with the user's profile

Expected Result: Users can choose audio coaching based on their workout type and intensity level.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that audio coaching integrates seamlessly with other app features\*\***

Test Case ID: TC004

Test Case Title: Verify that audio coaching integrates seamlessly with other app features

Test Case Description: This test case verifies that the audio coaching feature integrates with other features in the app, such as workout logging and goal setting.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

- \* User is logged in to the app
- \* User has a stable internet connection
- \* Audio coaching feature is enabled

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Workout Logging feature
2. Click on the Audio Coaching button
3. Select a workout type (e.g., cardio, strength training, yoga)
4. Verify that the audio coaching integrates with the workout logging feature
5. Verify that the audio coaching updates the user's progress towards their goals

Postconditions:

- \* User can view their workout history and progress towards their goals
- \* Workout data is synced with the user's profile

Expected Result: Audio coaching integrates seamlessly with other app features, such as workout logging and goal setting.

Severity: Major

Priority: Medium

Type of Testing: Integration Testing, Acceptance Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can control audio settings and preferences\*\***

Test Case ID: TC005

Test Case Title: Verify that users can control audio settings and preferences

Test Case Description: This test case verifies that users can adjust audio settings and preferences to customize their audio coaching experience.

Test Suite: Audio Coaching

Test Priority: Low

Preconditions:

- \* User is logged in to the app
- \* User has a stable internet connection
- \* Audio coaching feature is enabled



Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Settings feature
2. Click on the Audio Coaching settings
3. Verify that users can adjust audio settings, such as volume and playback speed
4. Verify that users can select preferred audio coaching styles, such as male or female voice

Postconditions:

- \* User can save their audio settings and preferences
- \* Workout data is synced with the user's profile

Expected Result: Users can control audio settings and preferences to customize their audio coaching experience.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

-----

**Data - 94**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

#### **Feature name -**

Integration with Health Apps

#### **Feature description -**

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

#### **Userstory title -**

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

#### **Acceptance criteria -**

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

#### **Generated Test Cases:**

Here are five detailed test cases for the "Integration with Health Apps" feature of the Fitness Tracker App.

**\*\*Test Case 1: Verify that the app integrates with popular health and wellness apps\*\***

Test Case ID: FT-INT-001

Test Case Title: Verify that the app integrates with popular health and wellness apps

Test Case Description: This test case ensures that the app can integrate with popular health and wellness apps, such as calorie counters, meditation apps, and mental health trackers.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- \* The user has a valid account on the fitness tracker app
- \* The user has installed a popular health and wellness app (e.g., MyFitnessPal)
- \* The app is configured to allow integration with other health apps

Test Data: None

Test Steps:

1. Launch the fitness tracker app and log in with a valid account
2. Navigate to the settings or integration section of the app
3. Select the popular health and wellness app (e.g., MyFitnessPal) to integrate with
4. Authenticate with the selected app and grant permission to access data
5. Verify that the app is connected successfully and displays a confirmation message

Postconditions:

- \* The user is redirected to the app's dashboard after integration
- \* The integrated app's data is displayed in the fitness tracker app's dashboard

Expected Result: The app integrates successfully with the popular health and wellness app, and the user can access combined data.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can sync data from connected health apps\*\***

Test Case ID: FT-INT-002

Test Case Title: Verify that users can sync data from connected health apps

Test Case Description: This test case ensures that users can sync data from connected health apps, such as exercise routines, sleep patterns, and nutrition plans.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- \* The user has integrated a health app with the fitness tracker app
- \* The user has generated data on the connected health app (e.g., logged a workout)

Test Data: None

Test Steps:

1. Launch the fitness tracker app and navigate to the dashboard
2. Select the connected health app from which to sync data
3. Click the "Sync" button to initiate data synchronization
4. Verify that the data is synced successfully and displayed in the fitness tracker app
5. Compare the synced data with the original data on the connected health app to ensure accuracy

Postconditions:

- \* The synced data is updated in the fitness tracker app
- \* The user is notified of any errors or discrepancies during sync

Expected Result: The user can sync data from connected health apps, and the data is accurate and up-to-date.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that the app provides a consolidated view of health metrics\*\***

Test Case ID: FT-INT-003

Test Case Title: Verify that the app provides a consolidated view of health metrics

Test Case Description: This test case ensures that the app provides a comprehensive view of the user's health metrics, including data from connected health apps.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- \* The user has integrated multiple health apps with the fitness tracker app
- \* The user has generated data on each connected health app (e.g., logged a workout, tracked nutrition)

Test Data: None

Test Steps:

1. Launch the fitness tracker app and navigate to the dashboard

2. Verify that the app displays a consolidated view of health metrics from all connected health apps
3. Check that the metrics are organized and easily accessible (e.g., graphs, charts, tables)
4. Verify that the user can filter or sort the metrics by date, type, or other criteria
5. Compare the consolidated view with the individual data from each connected health app to ensure accuracy

Postconditions:

- \* The user can access a comprehensive view of their health metrics
- \* The consolidated view is updated in real-time as new data is synced from connected health apps

Expected Result: The app provides a consolidated view of health metrics from connected health apps, enabling the user to track their overall well-being.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that users receive insights based on combined data from multiple apps\*\***

Test Case ID: FT-INT-004

Test Case Title: Verify that users receive insights based on combined data from multiple apps

Test Case Description: This test case ensures that the app provides valuable insights to users based on the combined data from multiple health apps.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- \* The user has integrated multiple health apps with the fitness tracker app
- \* The user has generated data on each connected health app (e.g., logged a workout, tracked nutrition)

Test Data: None

Test Steps:

1. Launch the fitness tracker app and navigate to the insights or recommendations section
2. Verify that the app displays personalized insights based on the combined data from connected health apps
3. Check that the insights are relevant and actionable (e.g., suggestions for improvement, alerts for unusual patterns)
4. Verify that the user can adjust the frequency or type of insights received
5. Compare the insights with the individual data from each connected health app to ensure accuracy and relevance

Postconditions:

- \* The user receives valuable insights based on combined data from multiple health apps
- \* The insights are updated in real-time as new data is synced from connected health apps

Expected Result: The app provides personalized insights based on combined data from multiple health apps, enabling the user to make informed decisions about their health and wellness.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that integration settings are easy to manage and configure\*\***

Test Case ID: FT-INT-005

Test Case Title: Verify that integration settings are easy to manage and configure

Test Case Description: This test case ensures that the app allows users to easily manage and configure integration settings with connected health apps.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- \* The user has integrated one or more health apps with the fitness tracker app
- \* The user has access to the app's settings or integration section

Test Data: None

Test Steps:

1. Launch the fitness tracker app and navigate to the settings or integration section
2. Verify that the app displays a list of connected health apps with configurable settings (e.g., data sharing, notification preferences)
3. Select a connected health app and adjust its integration settings (e.g., toggle data sharing on/off)
4. Verify that the changes are saved successfully and take effect immediately
5. Repeat steps 2-4 for multiple connected health apps to ensure consistency

Postconditions:

- \* The user can easily manage and configure integration settings for connected health apps



\* The changes are reflected in the app's behavior and data syncing

Expected Result: The app provides an intuitive and user-friendly interface for managing integration settings, enabling users to customize their experience.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

-----

**Data - 95**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Body Composition Tracking

**Feature description -**

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and the effectiveness of exercise and nutrition interventions.

**Userstory title -**

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed decisions about my training and nutrition.

### **Acceptance criteria -**

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set goals for body composition metrics.']

### **Generated Test Cases:**

Here are five detailed test cases for the Body Composition Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can log body composition metrics\*\***

Test Case ID: BTC-001

Test Case Title: Verify that users can log body composition metrics

Test Case Description: This test case verifies that users can successfully log body composition metrics such as weight, body fat percentage, and muscle mass.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

\* User is logged in to the app

\* User has access to the body composition tracking feature

Test Data:

\* Weight: 70 kg

\* Body fat percentage: 25%

\* Muscle mass: 35 kg

#### Test Steps:

1. Log in to the app as a registered user
2. Navigate to the body composition tracking feature
3. Enter weight, body fat percentage, and muscle mass data
4. Click the "Save" button
5. Verify that the data is saved successfully

#### Postconditions:

- \* The app displays a confirmation message indicating successful data saving
- \* The logged data is visible in the body composition tracking history

Expected Result: The user can successfully log body composition metrics, and the data is saved and visible in the app.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

\*\*Test Case 2: Verify that the app calculates and displays BMI based on logged data\*\*

Test Case ID: BTC-002

Test Case Title: Verify that the app calculates and displays BMI based on logged data

Test Case Description: This test case verifies that the app calculates and displays BMI based on the logged weight and height data.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- \* User has logged weight and height data
- \* User has access to the body composition tracking feature

Test Data:

- \* Weight: 70 kg
- \* Height: 175 cm

Test Steps:

1. Log in to the app as a registered user
2. Navigate to the body composition tracking feature
3. Verify that the app displays the calculated BMI based on the logged weight and height data

Postconditions:

- \* The app displays the calculated BMI value

Expected Result: The app correctly calculates and displays BMI based on the logged weight and height data.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can view trends and history of body composition changes\*\***

Test Case ID: BTC-003

Test Case Title: Verify that users can view trends and history of body composition changes

Test Case Description: This test case verifies that users can view trends and history of body composition changes over time.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- \* User has logged body composition data multiple times
- \* User has access to the body composition tracking feature

Test Data: No test data needed

Test Steps:

1. Log in to the app as a registered user
2. Navigate to the body composition tracking feature
3. Verify that the app displays a graphical representation of body composition changes over time
4. Verify that the app displays a table or list of historical body composition data

Postconditions:

- \* The app displays a clear and easy-to-understand graphical representation of body composition changes

Expected Result: The user can view trends and history of body composition changes over time.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the app provides insights and recommendations based on body composition data\*\***

Test Case ID: BTC-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 provides users with insights and recommendations based on their body composition data.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- \* User has logged body composition data

- \* User has access to the body composition tracking feature

Test Data: No test data needed

Test Steps:

1. Log in to the app as a registered user
2. Navigate to the body composition tracking feature
3. Verify that the app displays insights and recommendations based on the logged body composition data

Postconditions:

\* The app displays relevant and actionable insights and recommendations

Expected Result: The app provides users with relevant and actionable insights and recommendations based on their body composition data.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can set goals for body composition metrics\*\***

Test Case ID: BTC-005

Test Case Title: Verify that users can set goals for body composition metrics

Test Case Description: This test case verifies that users can set goals for body composition metrics such as weight, body fat percentage, and muscle mass.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

\* User has access to the body composition tracking feature

Test Data:

\* Weight goal: 65 kg

\* Body fat percentage goal: 20%

Test Steps:

1. Log in to the app as a registered user
2. Navigate to the body composition tracking feature
3. Click the "Set Goals" button
4. Enter weight and body fat percentage goal data
5. Click the "Save" button
6. Verify that the app displays a confirmation message indicating successful goal setting

Postconditions:

\* The app displays the set goals in the body composition tracking feature

Expected Result: The user can successfully set goals for body composition metrics, and the app displays the set goals.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive



---

## **Data - 96**

### **Project name -**

Fitness Tracker App

### **Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

### **Feature name -**

Workout Analysis

### **Feature description -**

The app offers detailed workout analysis tools, including performance metrics, trends, and insights into exercise intensity, duration, and calorie burn. Workout analysis helps users assess the effectiveness of their workouts, identify areas for improvement, and optimize training strategies for better results.

### **Userstory title -**

As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

### **Acceptance criteria -**

['Users can view detailed performance metrics for each workout.', 'The app provides trends and insights into workout intensity and duration.', 'Users can track calorie burn and other health metrics.', 'The app offers recommendations based on workout analysis.', 'Users can compare current workouts with past performance.']

### **Generated Test Cases:**

Here are five detailed test cases for the Workout Analysis feature of the Fitness Tracker App, covering each acceptance criterion:

**\*\*Test Case 1: Verify that users can view detailed performance metrics for each workout\*\***

Test Case ID: WT001

Test Case Title: View Workout Performance Metrics

Test Case Description: This test case verifies that users can view detailed performance metrics for each workout, including exercise intensity, duration, and calorie burn.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

\* User has completed a workout and logged it in the app

\* Workout data is synced with the app's database

Test Data: No test data needed

Test Steps:

1. Log in to the app with a valid username and password
2. Navigate to the Workout Analysis section
3. Select a completed workout from the list
4. Verify that the workout details page displays performance metrics, including exercise intensity, duration, and calorie burn
5. Validate that the metrics are accurate and correctly calculated based on the workout data

Postconditions:

\* Workout data is updated in the app's database

\* User can view detailed performance metrics for each workout

Expected Result: The app displays accurate and detailed performance metrics for each workout.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the app provides trends and insights into workout intensity and duration\*\***

Test Case ID: WT002

Test Case Title: Workout Trends and Insights

Test Case Description: This test case verifies that the app provides trends and insights into workout intensity and duration, enabling users to identify patterns and areas for improvement.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

\* User has completed multiple workouts and logged them in the app

\* Workout data is synced with the app's database

Test Data: No test data needed

Test Steps:

1. Log in to the app with a valid username and password
2. Navigate to the Workout Analysis section
3. Select a workout category (e.g., cardio, strength training)

4. Verify that the app displays trends and insights into workout intensity and duration, including average values and progress over time
5. Validate that the trends and insights are accurate and correctly calculated based on the workout data

Postconditions:

- \* Workout data is updated in the app's database
- \* User can view trends and insights into workout intensity and duration

Expected Result: The app provides accurate and useful trends and insights into workout intensity and duration.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can track calorie burn and other health metrics\*\***

Test Case ID: WT003

Test Case Title: Calorie Burn and Health Metrics Tracking

Test Case Description: This test case verifies that users can track calorie burn and other health metrics, enabling them to monitor their progress and adjust their workout routines accordingly.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- \* User has completed a workout and logged it in the app
- \* Workout data is synced with the app's database

Test Data: No test data needed

Test Steps:

1. Log in to the app with a valid username and password
2. Navigate to the Workout Analysis section
3. Select a completed workout from the list
4. Verify that the app displays calorie burn and other health metrics (e.g., distance traveled, heart rate) for the selected workout
5. Validate that the metrics are accurate and correctly calculated based on the workout data

Postconditions:

- \* Workout data is updated in the app's database
- \* User can view calorie burn and other health metrics for each workout

Expected Result: The app tracks calorie burn and other health metrics accurately.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the app offers recommendations based on workout analysis\*\***

Test Case ID: WT004

Test Case Title: Workout Recommendations

Test Case Description: This test case verifies that the app offers personalized recommendations based on

workout analysis, helping users optimize their training strategies.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- \* User has completed multiple workouts and logged them in the app

- \* Workout data is synced with the app's database

Test Data: No test data needed

Test Steps:

1. Log in to the app with a valid username and password
2. Navigate to the Workout Analysis section
3. Select a workout category (e.g., cardio, strength training)
4. Verify that the app displays personalized recommendations based on workout analysis, including suggested exercises, routines, and goals
5. Validate that the recommendations are accurate and relevant to the user's workout history and goals

Postconditions:

- \* Workout data is updated in the app's database

- \* User receives personalized workout recommendations

Expected Result: The app provides accurate and relevant workout recommendations based on analysis.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can compare current workouts with past performance\*\***

Test Case ID: WT005

Test Case Title: Workout Comparison

Test Case Description: This test case verifies that users can compare their current workouts with past performance, enabling them to track progress and identify areas for improvement.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

\* User has completed multiple workouts and logged them in the app

\* Workout data is synced with the app's database

Test Data: No test data needed

Test Steps:

1. Log in to the app with a valid username and password
2. Navigate to the Workout Analysis section
3. Select a workout from the list
4. Verify that the app displays a comparison of the current workout with past performances, including progress charts and statistics
5. Validate that the comparison is accurate and correctly calculated based on the workout data

Postconditions:

- \* Workout data is updated in the app's database
- \* User can compare current workouts with past performance

Expected Result: The app provides an accurate and useful comparison of current workouts with past performance.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 97**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Weather Integration

**Feature description -**

Integrated weather forecasts and conditions allow users to plan outdoor workouts and activities based on current weather conditions and forecasts. Weather integration helps users make informed decisions, stay safe, and optimize their outdoor exercise routines according to weather conditions.

**Userstory title -**



As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

### **Acceptance criteria -**

['The app provides current weather conditions and forecasts.', 'Users can view weather details for different locations.', 'Weather data is updated in real-time.', 'The app offers recommendations based on weather conditions.', 'Users can receive weather alerts for extreme conditions.']

### **Generated Test Cases:**

Here are five detailed test cases for the Weather Integration feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that the app displays current weather conditions\*\***

Test Case ID: WT-001

Test Case Title: Verify that the app displays current weather conditions

Test Case Description: This test case ensures that the app displays the current weather conditions for the user's location.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- \* User has granted location access to the app
- \* User is connected to the internet
- \* Weather API is integrated and functioning correctly

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the weather integration feature

2. Ensure that the app has access to the user's current location
3. Verify that the app displays the current weather conditions (e.g., temperature, humidity, wind speed)
4. Verify that the weather conditions are updated in real-time

Postconditions: None

Expected Result: The app displays the current weather conditions for the user's location

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can view weather details for different locations\*\***

Test Case ID: WT-002

Test Case Title: Verify that users can view weather details for different locations

Test Case Description: This test case ensures that users can view weather details for different locations.

Test Suite: Weather Integration

Test Priority: Medium

Preconditions:

- \* User has granted location access to the app
- \* User is connected to the internet
- \* Weather API is integrated and functioning correctly

Test Data: Multiple locations (e.g., New York, London, Tokyo)

Test Steps:

1. Launch the app and navigate to the weather integration feature
2. Enter a different location (e.g., New York)
3. Verify that the app displays the current weather conditions for the entered location
4. Repeat steps 2-3 for multiple locations

Postconditions: None

Expected Result: The app displays weather details for different locations

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that weather data is updated in real-time\*\***

Test Case ID: WT-003

Test Case Title: Verify that weather data is updated in real-time

Test Case Description: This test case ensures that weather data is updated in real-time.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

\* User has granted location access to the app

\* User is connected to the internet

\* Weather API is integrated and functioning correctly

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the weather integration feature
2. Verify that the app displays the current weather conditions
3. Wait for a few minutes and verify that the weather conditions are updated

Postconditions: None

Expected Result: Weather data is updated in real-time

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the app offers recommendations based on weather conditions\*\***

Test Case ID: WT-004

Test Case Title: Verify that the app offers recommendations based on weather conditions

Test Case Description: This test case ensures that the app offers recommendations based on weather conditions.

Test Suite: Weather Integration

Test Priority: Medium

Preconditions:

\* User has granted location access to the app

\* User is connected to the internet

\* Weather API is integrated and functioning correctly

Test Data: Different weather conditions (e.g., sunny, rainy, windy)

## Test Steps:

1. Launch the app and navigate to the weather integration feature
2. Verify that the app displays the current weather conditions
3. Verify that the app offers recommendations based on the weather conditions (e.g., "It's a great day for a run!" or "Consider indoor exercise today due to rain.")

Postconditions: None

Expected Result: The app offers recommendations based on weather conditions

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users receive weather alerts for extreme conditions\*\***

Test Case ID: WT-005

Test Case Title: Verify that users receive weather alerts for extreme conditions

Test Case Description: This test case ensures that users receive weather alerts for extreme conditions.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

\* User has granted location access to the app

\* User is connected to the internet

\* Weather API is integrated and functioning correctly

Test Data: Extreme weather conditions (e.g., hurricane, thunderstorm, heatwave)

Test Steps:

- 1. Launch the app and navigate to the weather integration feature
- 2. Verify that the app displays the current weather conditions
- 3. Simulate extreme weather conditions and verify that the app sends a weather alert to the user

Postconditions: None

Expected Result: The app sends a weather alert to the user for extreme conditions

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**Data - 98**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Mood Tracking

**Feature description -**

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

#### **Userstory title -**

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

#### **Acceptance criteria -**

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and energy levels.']

#### **Generated Test Cases:**

Here are five detailed test cases for the Mood Tracking feature of the Fitness Tracker App:

**\*\*Test Case 1: Verify that users can log their mood, energy levels, and motivation\*\***

Test Case ID: FT-001

Test Case Title: Verify that users can log their mood, energy levels, and motivation

Test Case Description: This test case ensures that users can successfully log their mood, energy levels, and motivation before and after workouts.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

\* The user has a valid account and is logged in

\* The user has completed a workout

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Mood Tracking feature
2. Select the "Log Mood" option
3. Choose a mood, energy level, and motivation from the provided options
4. Click "Save" to log the data

Postconditions:

\* The logged data is displayed in the user's mood tracking history

Expected Result: The user can successfully log their mood, energy levels, and motivation

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that the app provides insights into mood and energy trends over time\*\***

Test Case ID: FT-002

Test Case Title: Verify that the app provides insights into mood and energy trends over time

Test Case Description: This test case ensures that the app provides users with actionable insights into their mood and energy trends over time.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

\* The user has logged mood and energy data for at least 5 workouts



Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the Mood Tracking feature
2. Click on the "Insights" tab
3. Review the displayed trends and insights

Postconditions:

\* The insights are accurate and reflective of the user's logged data

Expected Result: The app provides users with actionable insights into their mood and energy trends over time

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users receive recommendations based on their mood and energy data\*\***

Test Case ID: FT-003

Test Case Title: Verify that users receive recommendations based on their mood and energy data

Test Case Description: This test case ensures that users receive personalized recommendations based on their mood and energy data.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

\* The user has logged mood and energy data for at least 3 workouts

Test Data: No test data needed

## Test Steps:

1. Launch the app and navigate to the Mood Tracking feature
2. Click on the "Recommendations" tab
3. Review the displayed recommendations

## Postconditions:

\* The recommendations are relevant and actionable for the user

Expected Result: The app provides users with personalized recommendations based on their mood and energy data

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that the app supports journaling for personal reflections on mood and workouts\*\***

Test Case ID: FT-004

Test Case Title: Verify that the app supports journaling for personal reflections on mood and workouts

Test Case Description: This test case ensures that users can reflect on their mood and workouts through journaling.

Test Suite: Mood Tracking

Test Priority: Medium

## Preconditions:

\* The user has completed a workout

Test Data: No test data needed

#### Test Steps:

1. Launch the app and navigate to the Mood Tracking feature
2. Click on the "Journal" tab
3. Write a journal entry reflecting on the user's mood and workout
4. Click "Save" to save the entry

#### Postconditions:

\* The journal entry is saved and displayed in the user's journal history

Expected Result: The app supports journaling for personal reflections on mood and workouts

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can set goals for improving mood and energy levels\*\***

Test Case ID: FT-005

Test Case Title: Verify that users can set goals for improving mood and energy levels

Test Case Description: This test case ensures that users can set and track goals for improving their mood and energy levels.

Test Suite: Mood Tracking

Test Priority: High

#### Preconditions:

\* The user has logged mood and energy data for at least 2 workouts

Test Data: No test data needed

Test Steps:

- 1. Launch the app and navigate to the Mood Tracking feature
- 2. Click on the "Goals" tab
- 3. Set a goal for improving mood and energy levels
- 4. Click "Save" to save the goal

Postconditions:

\* The goal is displayed in the user's goal tracking history

Expected Result: Users can set and track goals for improving their mood and energy levels

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

These test cases comprehensively cover all the acceptance criteria for the Mood Tracking feature, ensuring that the app provides a high-quality user experience and accurately tracks mood, energy levels, and motivation.

**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:**

Here are five detailed test cases for the Recipe Creation feature:

**\*\*Test Case 1: Verify that users can input and save ingredients and cooking instructions.\*\***

Test Case ID: RC001

Test Case Title: Verify that users can input and save ingredients and cooking instructions.

Test Case Description: Test that users can enter and save ingredients and cooking instructions for a new recipe.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- The user is logged in to the platform.
- The user has access to the recipe creation feature.
- No test data needed.

Test Steps:

1. Log in to the platform as a registered user.
2. Navigate to the recipe creation page.
3. Enter a valid recipe name, ingredients, and cooking instructions.
4. Click the "Save Recipe" button.
5. Verify that the recipe is saved successfully.

Postconditions:

- The recipe is saved in the user's profile.
- The recipe is accessible to other users on the platform.

Expected Result: The recipe is saved with the entered ingredients and cooking instructions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that users can upload photos for their recipes.\*\***

Test Case ID: RC002

Test Case Title: Verify that users can upload photos for their recipes.

Test Case Description: Test that users can upload a photo for a new recipe.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- The user is logged in to the platform.

- The user has access to the recipe creation feature.
- A test image file is available.

Test Steps:

1. Log in to the platform as a registered user.
2. Navigate to the recipe creation page.
3. Enter a valid recipe name, ingredients, and cooking instructions.
4. Click the "Browse" button to upload a photo.
5. Select a valid image file and click "Upload".
6. Verify that the photo is uploaded successfully.

Postconditions:

- The photo is displayed on the recipe page.
- The photo is associated with the saved recipe.

Expected Result: The photo is uploaded and displayed on the recipe page.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that recipes are saved in the user's profile.\*\***

Test Case ID: RC003

Test Case Title: Verify that recipes are saved in the user's profile.

Test Case Description: Test that recipes are saved in the user's profile after creation.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- The user is logged in to the platform.

- The user has access to the recipe creation feature.
- No test data needed.

Test Steps:

1. Log in to the platform as a registered user.
2. Navigate to the recipe creation page.
3. Enter a valid recipe name, ingredients, and cooking instructions.
4. Click the "Save Recipe" button.
5. Verify that the recipe is saved in the user's profile.

Postconditions:

- The recipe is displayed in the user's profile recipe list.
- The recipe is accessible from the user's profile.

Expected Result: The recipe is saved in the user's profile.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that recipes are accessible to other users on the platform.\*\***

Test Case ID: RC004

Test Case Title: Verify that recipes are accessible to other users on the platform.

Test Case Description: Test that recipes created by one user are accessible to other users on the platform.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- Two registered users are logged in to the platform.
- User 1 has created a recipe.



- No test data needed.

Test Steps:

1. Log in to the platform as User 1 and create a new recipe.
2. Log out as User 1 and log in as User 2.
3. Navigate to the recipe browse page.
4. Search for the recipe created by User 1.
5. Verify that the recipe is displayed in the search results.

Postconditions:

- The recipe is displayed in the search results.
- The recipe is accessible from the search results.

Expected Result: The recipe is accessible to other users on the platform.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users receive a confirmation upon successful recipe submission.\*\***

Test Case ID: RC005

Test Case Title: Verify that users receive a confirmation upon successful recipe submission.

Test Case Description: Test that users receive a confirmation message after successfully submitting a recipe.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- The user is logged in to the platform.
- The user has access to the recipe creation feature.
- No test data needed.

#### Test Steps:

1. Log in to the platform as a registered user.
2. Navigate to the recipe creation page.
3. Enter a valid recipe name, ingredients, and cooking instructions.
4. Click the "Save Recipe" button.
5. Verify that a confirmation message is displayed.

#### Postconditions:

- The confirmation message is displayed on the screen.
- The recipe is saved successfully.

Expected Result: The user receives a confirmation message upon successful recipe submission.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

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:**

Here are the test cases for the Recipe Discovery feature:

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

Test Case ID: RD-1

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

Test Case Description: This test case ensures that users can search for recipes using keywords and get relevant results.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

\* The user is logged in to the platform

\* The search bar is visible on the homepage

\* The user has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Enter a keyword (e.g., "chicken") in the search bar
2. Click the search button
3. Verify that the search results page is displayed
4. Verify that the search results include recipes with the keyword "chicken"

Postconditions:

- \* The search results page is displayed
- \* The search results include recipes with the keyword "chicken"

Expected Result: The user can search for recipes using keywords and get relevant results.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

**\*\*Test Case 2: Verify that search results are relevant and accurately match the search criteria\*\***

Test Case ID: RD-2

Test Case Title: Verify that search results are relevant and accurately match the search criteria

Test Case Description: This test case ensures that the search results are relevant and accurately match the search criteria.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- \* The user is logged in to the platform
- \* The search bar is visible on the homepage
- \* The user has a stable internet connection

Test Data:

- \* Keyword: "vegan"
- \* Ingredient: "quinoa"

Test Steps:

1. Enter the keyword "vegan" and ingredient "quinoa" in the search bar
2. Click the search button
3. Verify that the search results page is displayed
4. Verify that the search results include recipes that are vegan and contain quinoa

Postconditions:

- \* The search results page is displayed
- \* The search results include recipes that are vegan and contain quinoa

Expected Result: The search results are relevant and accurately match the search criteria.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

**\*\*Test Case 3: Verify that users can filter search results by cuisine type, dietary preferences, and other criteria\*\***

Test Case ID: RD-3

Test Case Title: Verify that users can filter search results by cuisine type, dietary preferences, and other criteria

Test Case Description: This test case ensures that users can filter search results by cuisine type, dietary preferences, and other criteria.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

- \* The user is logged in to the platform
- \* The search bar is visible on the homepage
- \* The user has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Enter a keyword (e.g., "Italian") in the search bar
2. Click the search button
3. Verify that the search results page is displayed
4. Click on the "Filter" button

5. Select a cuisine type (e.g., "Italian")

6. Verify that the search results are filtered to show only Italian recipes

Postconditions:

- \* The search results page is displayed

- \* The search results are filtered to show only Italian recipes

Expected Result: The user can filter search results by cuisine type, dietary preferences, and other criteria.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

**\*\*Test Case 4: Verify that recipes in the search results are accessible and detailed\*\***

Test Case ID: RD-4

Test Case Title: Verify that recipes in the search results are accessible and detailed

Test Case Description: This test case ensures that recipes in the search results are accessible and detailed.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

- \* The user is logged in to the platform

- \* The search bar is visible on the homepage

- \* The user has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Enter a keyword (e.g., "chicken") in the search bar
2. Click the search button
3. Verify that the search results page is displayed
4. Click on a recipe in the search results
5. Verify that the recipe details page is displayed
6. Verify that the recipe details include ingredients, instructions, and cooking time

Postconditions:

- \* The recipe details page is displayed
- \* The recipe details include ingredients, instructions, and cooking time

Expected Result: Recipes in the search results are accessible and detailed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

**\*\*Test Case 5: Verify that users can save search results for future reference\*\***

Test Case ID: RD-5

Test Case Title: Verify that users can save search results for future reference

Test Case Description: This test case ensures that users can save search results for future reference.

Test Suite: Recipe Discovery



Test Priority: Low

Preconditions:

- \* The user is logged in to the platform
- \* The search bar is visible on the homepage
- \* The user has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Enter a keyword (e.g., "breakfast") in the search bar
2. Click the search button
3. Verify that the search results page is displayed
4. Click on the "Save" button
5. Verify that the search results are saved to the user's profile

Postconditions:

- \* The search results are saved to the user's profile

Expected Result: The user can save search results for future reference.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

-----