

Generated Test Cases via gpt-4 LLM Model and Zero-shot Prompting Technique

Data - 1

Project name -

Social Media Platform

Project description -

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

Feature name -

User Profiles

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1

Test Case ID: TC_USR_001

Test Case Title: Verify that users can add and edit their name, bio, profile picture, location, and interests.

Test Case Description: This test case verifies that the user is able to add and edit all the necessary personal information to their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions: The user must be registered and logged in.

Test Data: Test data includes various user information such as name, bio, profile picture, location, and interests.

Test Steps:

1. Login as a standard user.
2. Navigate to the profile information section.
3. Add or edit personal information.
4. Save the changes.

Postconditions: The profile information should reflect the changes made.

Expected Result: The personal information should be updated successfully and viewable in the user's profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_USR_002

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

Test Case Description: This test case verifies that the profile picture field supports standard image formats

such as JPEG, PNG, etc.

Test Suite: User Profiles

Test Priority: Medium

Preconditions: The user must be logged in and at the 'Edit profile' section.

Test Data: Image files in multiple formats (JPEG, PNG, GIF, etc.)

Test Steps:

1. Login as a standard user.
2. Navigate to the 'Edit profile' section.
3. Upload profile pictures in different file formats.
4. Save changes.

Postconditions: The uploaded profile picture should be displayed properly.

Expected Result: The profile picture, regardless of the format, should appear correctly after being uploaded.

Severity: Major

Priority: Medium

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_USR_003

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

Test Case Description: This test case checks the ability to configure different privacy settings for each field in the profile.

Test Suite: User Profiles

Test Priority: High

Preconditions: A user account must be created and logged in.

Test Data: No test data needed.

Test Steps:

1. Login as a registered user.
2. Navigate to the profile settings.
3. Set different privacy settings for each field.
4. Save changes.

Postconditions: Privacy settings should apply successfully.

Expected Result: The privacy settings for each field should be updated as per user selection.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC_USR_004

Test Case Title: Verify that the profile updates are saved successfully.

Test Case Description: This test case verifies if the system successfully saves any updates made to the user profile.

Test Suite: User Profiles

Test Priority: High

Preconditions: The user must be registered and logged in.

Test Data: User information to update profile.

Test Steps:

1. Login as a standard user.
2. Navigate to the 'Edit profile' section.
3. Change any profile information.
4. Save changes.

Postconditions: User's profile must be updated successfully.

Expected Result: The profile must show the updated changes after they are saved.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC_USR_005

Test Case Title: Verify that users receive a confirmation upon successful profile updates.

Test Case Description: This case tests whether the user receives a confirmation message after successful profile updates.

Test Suite: User Profiles

Test Priority: Medium

Preconditions: The user must be registered and logged in.

Test Data: User information to update profile.

Test Steps:

1. Login as a standard user.
2. Navigate to the 'Edit profile' section.
3. Change any profile information.
4. Save changes.

Postconditions: A confirmation message must appear.

Expected Result: A confirmation message indicating a successful profile update should be shown.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Data - 2

Project name -

Social Media Platform

Project description -

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

Feature name -

News Feed

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: NF01

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

Test Case Description: This test case is designed to validate that the user's news feed is populated with posts from accounts that the user follows.

Test Suite: News Feed

Test Priority: High

Preconditions: User is logged in, User follows at least one account

Test Data: Username, password

Test Steps: 1. Log in to the system. 2. Navigate to the news feed. 3. Identify posts belonging to followed accounts.

Postconditions: Log out of the system

Expected Result: All posts in the news feed belong to accounts followed by the user

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: NF02

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

Test Case Description: This test case ensures that the posts in the news feed are ordered either chronologically or based on the implemented algorithm.

Test Suite: News Feed

Test Priority: High

Preconditions: User is logged in, User follows at least two accounts that have multiple posts

Test Data: Username, password

Test Steps: 1. Log in to the system. 2. Navigate to the news feed. 3. Identify the order of the posts.

Postconditions: Log out of the system

Expected Result: Content in the news feed is displayed in chronological or algorithmic order

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: NF03

Test Case Title: Verify that users can interact with posts through likes, comments, and shares

Test Case Description: This test case verifies the functionality of buttons for "Like", "Comment", and "Share" on individual posts on the news feed.

Test Suite: News Feed

Test Priority: High

Preconditions: User is logged in, There are posts on the news feed

Test Data: Username, password

Test Steps: 1. Log in to the system. 2. Navigate to the news feed. 3. Interact with a post through "Like", "Comment", and "Share".

Postconditions: Log out of the system

Expected Result: User successfully likes, comments, and shares a post

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: NF04

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

Test Case Description: This test case is designed to confirm that the news feed updates in real-time when there are new posts from followed accounts.

Test Suite: News Feed

Test Priority: High

Preconditions: User is logged in, User follows at least one account, The followed account creates a new post during the test

Test Data: Username, password

Test Steps: 1. Log in to the system. 2. Navigate to the news feed. 3. Wait for the followed account to post a new update.

Postconditions: Log out of the system

Expected Result: The new post appears in the news feed in real-time

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: NF05

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

Test Case Description: This test case will verify the functionality of the filter options for content types and

topics in the news feed.

Test Suite: News Feed

Test Priority: High

Preconditions: User is logged in, There are various types and topics of posts on the news feed

Test Data: Username, password

Test Steps: 1. Log in to the system. 2. Navigate to the news feed. 3. Filter posts based on a specific type or topic.

Postconditions: Log out of the system

Expected Result: The news feed displays only posts matching the filter criteria

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 3

Project name -

Social Media Platform

Project description -

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

Feature name -

Messaging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TC_MSG_001

Test Case Title: Verify that users can send text messages to their friends successfully.

Test Case Description: This test case verifies that the Messaging feature allows users to send text messages to other users.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged in the system.
- User has at least one friend in the contact list.
- Friend's status is online.

Test Data: Text message content

Test Steps:

1. Log in to the application.
2. Select a friend from the contact list.

3. Open a chat window.
4. Type a message in the text field.
5. Click on the 'Send' button.

Postconditions:

- The message appears in the chat window.
- Notification is sent to the recipient.

Expected Result: The entered text message is successfully sent to the selected friend.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_MSG_002

Test Case Title: Verify that images and videos can be sent in messages.

Test Case Description: This test case checks if users can send multimedia (images, videos) through messages.

Test Suite: Messaging

Test Priority: Medium

Preconditions:

- User is logged in the system.
- User has valid multimedia files on their device.
- User is in the chat window with a friend.

Test Data: Image file, Video file

Test Steps:

1. Log in to the application.

2. Open a chat window with a friend.
3. Click on the 'Attach' button.
4. Select 'Image' or 'Video' option.
5. Upload the appropriate multimedia file.
6. Click on the 'Send' button.

Postconditions:

- Multimedia file is sent in the message.
- File appears in the chat window.
- Notification is sent to the recipient.

Expected Result: The multimedia file (image or video) is successfully sent to the friend.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_MSG_003

Test Case Title: Verify that users can start a group chat.

Test Case Description: This test case validates the functionality of creating a group to have a group chat.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged into the system.
- User has more than two friends in the contact list.

Test Data: Friends' names

Test Steps:

1. Log in to the application.
2. Click on 'New Group' in the chat section.
3. Select friends to add in the group.
4. Click on 'Create Group'.

Postconditions:

- A new group chat window opens.
- All added friends are present in the group.

Expected Result: The group is successfully created with selected friends, and a group chat window is opened.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_MSG_004

Test Case Title: Verify the functionality of read receipts in chat.

Test Case Description: This test case verifies whether the Messaging feature shows read receipts when a friend reads the user's message.

Test Suite: Messaging

Test Priority: Medium

Preconditions:

- User is logged into the system.
- User has sent a text message to a friend.
- Friend's status is online.

Test Data: No test data needed

Test Steps:

1. Log in to the application.
2. Open a chat window with a friend.
3. Send a text message.
4. Wait for the friend to read the message.

Postconditions:

- The status of the message changes to 'Read'.

Expected Result: The sent text message shows read receipts when the friend reads the user's message.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_MSG_005

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

Test Case Description: This test case verifies whether users receive notifications when new messages are received.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged into the system.
- User received a new message.
- Notification setting for new messages is enabled.

Test Data: No test data needed

Test Steps:

1. Log in to the application.

2. Wait for a friend to send a new message.

3. Check the notification icon or area.

Postconditions:

- A notification pops up or appears in the notification area.

Expected Result: The user receives a notification for a new message.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 4

Project name -

Social Media Platform

Project description -

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

Feature name -

Content Sharing

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: CSP-01

Test Case Title: Verify that users can create and share text posts, photos, and videos

Test Case Description: This test case verifies the basic functionality of creating and sharing various types of content on the social media platform.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is registered and logged in
- User has access to the content creation module

Test Data: Sample text post, photo, and video

Test Steps:

1. Navigate to the content creation module.
2. Create a text post, attach a photo or video.
3. Confirm the content and click on the "Share" button.
4. Check if the post shows up on your profile/news feed.

Postconditions:

- Content is posted to the user's profile/news feed

Expected Result: The respective text post, photo, or video is successfully posted to the user's profile/news

feed.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: CSP-02

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

Test Case Description: This test case verifies the functionality of setting the sharing privacy for posts to followers or public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is registered and logged in
- User creates a post

Test Data: No test data needed

Test Steps:

1. Create a text post.
2. Choose "Share with followers" or "Share publicly" before posting.
3. Confirm and post the content.
4. Check your profile/news feed from another account

Postconditions:

- The post appears based on the selected sharing privacy

Expected Result: The post should appear accordingly, either only visible to followers or publicly visible.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Please note, this structure would follow for the remaining 3 test cases, but changing Test Case ID, Test Case Title, Test Case Description, Preconditions, Test Steps, Postconditions, Expected Result, and Test Case Approach according to the acceptance criteria which have to be tested. The Test Suite, Test Priority, Severity, Priority, and Type of Testing would remain the same, indicating that all these test cases related to this user story are of high priority, major severity, functional in nature, and follow mostly a positive test case approach.

Data - 5

Project name -

Social Media Platform

Project description -

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

Feature name -

Privacy Settings

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: 001

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

Test Case Description: This test case was created to confirm that users can change their privacy settings to either public, private or restricted.

Test Suite: Privacy Settings

Test Priority: High

Preconditions: User has an account on the social media platform and is logged in.

Test Data: No test data needed.

Test Steps: 1. Go to 'Settings'. 2. Click on 'Privacy Settings'. 3. Choose either 'Public', 'Private', or 'Restricted'. 4. Save changes.

Postconditions: The system saves the new privacy settings.

Expected Result: The user's privacy setting is changed and saved successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: 002

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

Test Case Description: This test case aims to determine if the privacy settings can be altered by the user at any moment.

Test Suite: Privacy Settings

Test Priority: High

Preconditions: User has an account on the social media platform and is logged in.

Test Data: No test data needed

Test Steps: 1. Go to 'Settings'. 2. Click on 'Privacy Settings'. 3. Alter the privacy setting. 4. Save changes. 5.

Repeat steps 2-4 after an interval.

Postconditions: The system updates the user's privacy settings as many times as needed.

Expected Result: The privacy setting change is saved every time regardless of frequency.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: 003

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

Test Case Description: This test scenario checks if users can manipulate the visibility of individual posts through their privacy settings.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions: User has an account on the social media platform, is logged in and has made at least one post.

Test Data: User's posts

Test Steps: 1. Go to user's post. 2. Click on 'Privacy Settings' for the post. 3. Select the desired visibility option. 4. Save changes.

Postconditions: The post's visibility setting is changed and saved.

Expected Result: The visibility of the individual post changes according to user's selection.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: 004

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

Test Case Description: This case ensures the system instantly applies changes made to privacy settings.

Test Suite: Privacy Settings

Test Priority: High

Preconditions: User has an account on the social media platform and is logged in.

Test Data: No test data needed

Test Steps: 1. Go to 'Settings'. 2. Click on 'Privacy Settings'. 3. Alter the privacy setting. 4. Save changes. 5. Check if the modified settings are immediately in effect.

Postconditions: The altered Privacy Settings become effective instantly.

Expected Result: The system immediately starts respecting the updated privacy settings.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: 005

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

Test Case Description: This case ensures that users receive notifications whenever they make changes to their privacy settings.

Test Suite: Privacy Settings

Test Priority: High

Preconditions: User has an account on the social media platform and is logged in.

Test Data: No test data needed

Test Steps: 1. Go to 'Settings'. 2. Click on 'Privacy Settings'. 3. Alter the privacy setting. 4. Save changes. 5. Check if a notification appears about the change.

Postconditions: A notification is sent out due to the alterations in Privacy Settings.

Expected Result: The user receives a notification confirming the change in privacy settings.

Severity: Critical

Priority: High

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:

Test Case 1:

Test Case ID: TC_001

Test Case Title: Verify that users receive notifications for likes, comments, and shares.

Test Case Description: This test case checks if the system is generating notifications when users receive likes, comments, and shares.

Test Suite: In-App Notifications

Test Priority: High

Preconditions:

- User is logged in to the platform

- User has some posts
- Other users have access to these posts

Test Data: Posts, Likes, Comments, and Shares from other users.

Test Steps:

1. Log in to the user profile
2. Navigate to a post
3. From another user profile, Like, Comment, and Share the post
4. Check if a notification appears on the original user's notification panel

Postconditions:

- Notification received and acknowledged
- User redirected to relevant post upon clicking notification

Expected Result: Notification for like, comment, and share appears in the real-time on the user's notification panel.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_002

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

Test Case Description: This test case checks the real-time functionality of notifications.

Test Suite: In-App Notifications

Test Priority: High

Preconditions:

- User is logged in to the platform

- Other users interact with the user's posts or profile in some way

Test Data: Interactions from other users.

Test Steps:

1. Log in to the user profile
2. Have a different user perform an action (like, comment, share, friend request) related to the logged-in user's content or profile
3. Check whether the notification for that action appears immediately in the notification panel

Postconditions:

- Notification received and acknowledged
- User redirected to relevant activity upon clicking notification

Expected Result: Notification appears immediately after the concerned action is performed.

Severity: Critical

Priority: High

Type of Testing: Performance Testing

Test Case Approach: Positive

(Test cases will continue following these same steps using this template until all acceptance criteria have been fully covered by at least one test case)

Data - 7

Project name -

Social Media Platform

Project description -

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

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

Feature name -

Search Functionality

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

- Test Case ID: TC_SRCH_01
- Test Case Title: Verify that users can search for content using keywords or hashtags.
- Test Case Description: This test case verifies if the search functionality allows users to find specific content or profiles by entering keywords or hashtags in the search bar.
- Test Suite: Search Functionality Suite.
- Test Priority: High.
- Preconditions: The user must have an account and be logged in. The platform should have enough content for search results.
- Test Data: Keyword, e.g. #Technology, Coding.

- Test Steps:

1. Login to the application.
2. Enter the keyword or hashtag in the search bar.
3. Click on the search button.

- Postconditions: Display the search results on a new page.

- Expected Result: The system should display content matching the keyword or hashtag in the results.

- Severity: Major.

- Priority: High.

- Type of Testing: Functional Testing.

- Test Case Approach: Positive.

Test Case 2:

- Test Case ID: TC_SRCH_02

- Test Case Title: Verify the order of displaying search results based on relevance.

- Test Case Description: This test case checks if the search results are displayed in order of relevance.

- Test Suite: Search Functionality Suite.

- Test Priority: Medium.

- Preconditions: User has entered a valid keyword or hashtag and executed a search.

- Test Data: Keyword or hashtag.

- Test Steps:

1. Login to the platform.
2. Enter a valid keyword (or hashtag) and perform a search.
3. View the results of the search.

- Postconditions: None.

- Expected Result: Search results should be displayed in order of relevance to the keyword or hashtag entered.

- Severity: Minor.
- Priority: Medium.
- Type of Testing: Functional Testing.
- Test Case Approach: Positive.

Test Case 3:

- Test Case ID: TC_SRCH_03
- Test Case Title: Verify that the search results include profiles, posts, and topics.
- Test Case Description: The test case checks the variety of results that the search operation should display - profiles, posts, and topics linked to the keyword or hashtag entered.
- Test Suite: Search Functionality Suite.
- Test Priority: High.
- Preconditions: User has entered a valid keyword and executed a search.
- Test Data: Valid Keyword.
- Test Steps:
 1. Login to the platform.
 2. Enter a popular keyword and execute a search.
 3. Examine the search results.
- Postconditions: None.
- Expected Result: The search results should include profiles, posts, and topics related to the keyword entered.
- Severity: Major.
- Priority: High.
- Type of Testing: Functional Testing.
- Test Case Approach: Positive.

Test Case 4:

- Test Case ID: TC_SRCH_04
- Test Case Title: Verify that users can filter search results by content type.
- Test Case Description: This test case ensures that users can refine their search results by choosing to view only profiles, posts, or topics.
- Test Suite: Search Functionality Suite.
- Test Priority: Medium.
- Preconditions: User has performed a search and obtained results.
- Test Data: Filter selection (Posts, profiles, topics).
- Test Steps:
 1. Perform a search with a popular keyword.
 2. Filter the results by one of the content types.
 3. Check the filtered results.
- Postconditions: None.
- Expected Result: The search results should display only the chosen content type.
- Severity: Major.
- Priority: Medium.
- Type of Testing: Functional Testing.
- Test Case Approach: Positive.

Test Case 5:

- Test Case ID: TC_SRCH_05
- Test Case Title: Verify the auto-suggestion feature while typing in the search bar.
- Test Case Description: This test case is designed to verify if the auto-suggestions appear while typing in the search bar.
- Test Suite: Search Functionality Suite.

- Test Priority: Low.
 - Preconditions: User is logged into the application.
 - Test Data: No test data needed.
 - Test Steps:
 1. Login to the platform.
 2. Start typing in the search bar.
 - Postconditions: None.
 - Expected Result: As the user types, automatic keyword suggestions should appear.
 - Severity: Minor.
 - Priority: Low.
 - 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:

Test Case 1:

- Test Case ID: 001
- Test Case Title: Verify that users can add hashtags to their posts.
- Test Case Description: To test that the system allows users to add hashtags to their posts and that these hashtags are created and displayed properly.
- Test Suite: User Content Features
- Test Priority: High
- Preconditions: User is logged into their account; user is creating or editing a post.
- Test Data: Post content, hashtag (#example)
- Test Steps:
 1. Log into the user account.
 2. Navigate to create a new post.
 3. Add text to the post and include a hashtag (#example).
 4. Click on "Post".
 5. View the posted content.
- Postconditions: The post with a hashtag is published on the platform.

- Expected Result: The hashtag (#example) is highlighted and clickable in the published post.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing, User Interface Testing
- Test Case Approach: Positive

Test Case 2:

- Test Case ID: 002
- Test Case Title: Verify that users can tag other profiles, brands, or locations.
- Test Case Description: To test if the users are allowed to tag others and that these tags are created correctly and lead to the tagged profiles, brands, or locations when clicked.
- Test Suite: User Content Features
- Test Priority: High
- Preconditions: Users should be logged into their account; users are composing a post.
- Test Data: Post content, profile/brand/location tag (@example)
- Test Steps:
 1. Log into the user account.
 2. Navigate to create a new post.
 3. Add text to the post and include a tag (@example).
 4. Click on "Post".
 5. View the posted content.
- Postconditions: The post containing a tag is available on the platform for other users to view.
- Expected Result: The tagged profile/brand/location (@example) is highlighted and clickable in the published post.
- Severity: Major
- Priority: High

- Type of Testing: Functional Testing, User Interface Testing
- Test Case Approach: Positive

Test Case 3:

- Test Case ID: 003
- Test Case Title: Verify that hashtags and tags are clickable and lead to related content.
- Test Case Description: Ensure that users can engage with social features of the platform by clicking on hashtags or tags to locate related content.
- Test Suite: User Interface and Interaction
- Test Priority: High
- Preconditions: User is logged into their account; there are posts with tags and hashtags.
- Test Data: No test data needed.
- Test Steps:
 1. Log into the user's account.
 2. Find a post with a hashtag or tag.
 3. Click on the hashtag or tag.
- Postconditions: User is redirected to a page showing all related content.
- Expected Result: A page appears showing all posts containing the clicked hashtag or tag.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing, User Experience Testing
- Test Case Approach: Positive

Test Case 4:

- Test Case ID: 004

- Test Case Title: Verify that users can search for content using hashtags.
- Test Case Description: To confirm that the search functionality allows users to find content using hashtags correctly.
- Test Suite: Search Functionality
- Test Priority: High
- Preconditions: User is logged into their account; there are posts containing hashtags.
- Test Data: Hashtag for search (#example)
- Test Steps:
 1. Log into the user's account.
 2. Navigate to the search bar.
 3. Enter a hashtag (#example) into the search bar.
 4. Click on the search icon or press Enter.
- Postconditions: User is presented with the search results.
- Expected Result: A list of posts containing the searched hashtag (#example) appears.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing, User Interface Testing
- Test Case Approach: Positive

Test Case 5:

- Test Case ID: 005
- Test Case Title: Verify that the system suggests popular hashtags as users type.
- Test Case Description: Validate that the system's hashtag recommendation feature functions as expected, suggesting popular hashtags to users as they compose posts.
- Test Suite: User Content Features
- Test Priority: Medium

- Preconditions: User is logged into their account; user begins to compose a post.
 - Test Data: Partial hashtag input (#exa)
 - Test Steps:
 1. Log into the user account.
 2. Navigate to create a new post.
 3. Start typing a hashtag (#exa).
 - Postconditions: Hashtag suggestions appear.
 - Expected Result: The system suggests popular hashtags that match the input (#exa).
 - Severity: Minor
 - Priority: Medium
 - Type of Testing: Functional Testing, User Interface Testing, User Experience Testing
 - Test Case Approach: Positive
-

Data - 9

Project name -

Social Media Platform

Project description -

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

Feature name -

Emojis and Reactions

Feature description -

Emojis and reactions provide users with expressive tools to react to posts and messages. Users can use emojis

to convey emotions such as joy, sadness, love, or excitement, enhancing communication and engagement on the platform.

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TC001_EA

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

Test Case Description: Test to confirm that the platform allows users to react to posts using different emojis.

Test Suite: Emoji & Reaction Functionality Suite

Test Priority: High

Preconditions: User is logged in and viewing a post.

Test Data: No test data needed.

Test Steps:

1. Navigate to a post.
2. Click on the "react" button under the post.
3. Select an emoji from the provided list.

Postconditions: User has reacted to the post, the post reaction count increments by 1.

Expected Result: Post should display user's selected emoji as a reaction and the reaction count increases.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002_EA

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

Test Case Description: This will test whether emojis can be successfully added within text in comments and messages.

Test Suite: Emoji & Reaction Functionality Suite

Test Priority: High

Preconditions: User is logged in, and is in a text-entry field (comment box or message).

Test Data: No test data needed.

Test Steps:

1. Navigate to a post or message box.
2. Click in the text-entry field.
3. Open the emoji picker.
4. Choose and insert an emoji.
5. Submit the comment or message.

Postconditions: The sent message or comment with the emoji is visibly posted.

Expected Result: The comment or message should be posted successfully with the chosen emoji.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003_EA

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

Test Case Description: This test is to verify if the reaction count on posts is visible and accurately updated.

Test Suite: Emoji & Reaction Functionality Suite

Test Priority: Medium

Preconditions: User is logged in and viewing a post, post has reactions by different users.

Test Data: No test data needed.

Test Steps: View a post with reactions.

Postconditions: None.

Expected Result: User should be able to see a count of reactions on the post and it should match the actual number of reactions.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004_EA

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

Test Case Description: This test is to verify if a broad spectrum of emojis is accessible for user interactions.

Test Suite: Emoji & Reaction Functionality Suite

Test Priority: Medium

Preconditions: User is logged in and is at a place where emoji interaction is possible.

Test Data: No test data needed.

Test Steps:

1. Navigate to an emoji interaction area (e.g., a post, message box).
2. Open the emoji picker.

Postconditions: None.

Expected Result: A wide variety of emojis should be visible and selectable by the user.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005_EA

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

Test Case Description: Test to ensure users have the ability to change or remove their reactions on posts.

Test Suite: Emoji & Reaction Functionality Suite

Test Priority: High

Preconditions: User is logged in and has reacted to a post.

Test Data: No test data needed.

Test Steps:

1. Navigate to a post the user has reacted to.
2. Click on the user's reaction.
3. Select a different emoji or 'No Reaction'.

Postconditions: User's reaction is changed or removed on that post.

Expected Result: Post reaction should change according to user's new selection, or be removed if 'No Reaction' is selected. The reaction count adjusts accordingly.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive.

Data - 10

Project name -

Social Media Platform

Project description -

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

Feature name -

Live Streaming

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TC_001

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

Test Case Description: In this test case, the process of starting and stopping a user's live stream is evaluated.

Test Suite: Live Streaming Suite

Test Priority: High

Preconditions:

- User has a valid login
- User has a stable internet connection
- User has access to broadcasting device

Test Data: User login data

Test Steps:

1. Login to the social media account.
2. Navigate to the live streaming feature.
3. Start the live streaming.
4. Broadcast content.
5. Stop the live streaming.

Postconditions:

- Confirmation message of shutting down the stream
- User redirected to previous screen

Expected Result: The user should be able to start and stop a live stream successfully without any glitches.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_002

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

Test Case Description: This test case evaluates the privacy options available for live streaming.

Test Suite: Live Streaming Suite

Test Priority: Medium

Preconditions:

- User has a valid login
- User is about to start a live stream

Test Data: User login data

Test Steps:

1. Login to the social media account.
2. Navigate to the live streaming feature.
3. Select the audience for the stream: public or followers only.
4. Start the live streaming.
5. Verify the audience visibility.

Postconditions:

- The chosen audience is able to view the live stream

Expected Result: The live stream should be visible only to the selected audience.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_003

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

Test Case Description: This test case tests the degree of interactivity in the live streaming feature.

Test Suite: Live Streaming Suite

Test Priority: High

Preconditions:

- User has a valid login
- A live stream is ongoing

Test Data: User login data, live stream link

Test Steps:

1. Login to the social media account.
2. Navigate to the ongoing live stream.
3. Post a comment or react to the live stream.
4. Verify that the comment or reaction is posted successfully.

Postconditions:

- User interaction recorded and displayed on the live stream session

Expected Result: The viewer should be able to post comments and reactions successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_004

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

Test Case Description: This test case assesses the visibility of viewer count and comments in the live

streaming feature.

Test Suite: Live Streaming Suite

Test Priority: Medium

Preconditions:

- User has a valid login
- A live stream is ongoing

Test Data: User login data, live stream link

Test Steps:

1. Login to the social media account.
2. Navigate to the ongoing live stream.
3. Observe the viewer count and comments displayed.

Postconditions:

- Updated viewer count and comments are visible on the live stream session.

Expected Result: Viewer count and live comments should be accurately displayed during the live stream.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_005

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

Test Case Description: This test case evaluates the functionality of saving a completed live stream for later viewing.

Test Suite: Live Streaming Suite

Test Priority: Low

Preconditions:

- User has a valid login
- A live stream has been completed

Test Data: User login data, finished live stream link

Test Steps:

1. Login to the social media account.
2. Navigate to the completed live stream.
3. Select the option to save the live stream.
4. Verify that the live stream is saved successfully.

Postconditions:

- The saved live stream is visible in the user's video archive.

Expected Result: The user should be able to save completed live streams for later viewing.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 11

Project name -

Social Media Platform

Project description -

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

Feature name -

Stories

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

- Test Case ID: TCS_01
- Test Case Title: Verify that users can create stories with photos, videos, and text overlays.
- Test Case Description: This test case checks if users can correctly create stories by adding photos, videos, and text overlays.
- Test Suite: Stories Feature
- Test Priority: High
- Preconditions:
 - User must be logged in.
 - User has access to photo, video, or text overlay creation functions.
- Test Data: Various types of photos, videos, and text.
- Test Steps:

1. Log in to the application.
 2. Navigate to the 'Stories' feature.
 3. Initiate 'Create Story' action.
 4. Select and upload a photo, video, or create a text overlay.
 5. Finish and post the story.
- Postconditions:
 - User's story is live for their followers to see.
 - Expected Result: The created story should be successfully posted with the selected photo, video, or text overlay.
 - Severity: Major
 - Priority: High
 - Type of Testing: Functional Testing
 - Test Case Approach: Positive

Test Case 2:

- Test Case ID: TCS_02
- Test Case Title: Verify that stories disappear after 24 hours.
- Test Case Description: This test case aims to confirm that any posted story stays live for 24 hours only.
- Test Suite: Stories Feature
- Test Priority: High
- Preconditions:
 - User must be logged in.
 - User has posted a story.
- Test Data: No test data needed.
- Test Steps:
 1. Log in to the application.

2. Create and post a story.
3. Wait for 24 hours.
4. Check the story after 24 hours.

- Postconditions:
 - The viewed story should not be available.
- Expected Result: After 24 hours, the posted story should disappear from the user's profile.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Please note: The rest of the test cases would follow a similar pattern, testing different acceptance criteria such as users being able to see who viewed their stories, the ability to share stories publicly or with selected followers, and the functionality of adding stickers and filters to their stories.

Data - 12

Project name -

Social Media Platform

Project description -

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

Feature name -

Analytics and Insights

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case ID: TC-01

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

Test Case Description: This test case is designed to validate if users can view information regarding follower demographics such as age, gender, and location.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged into the platform
- User has followers

Test Data: Data regarding followers' demographics

Test Steps:

1. Log into user account
2. Navigate to "Analytics and Insights" section

3. Select "Follower Demographics"

Postconditions:

- System should display follower demographics data

Expected Result: User can view follower demographics data in a user-friendly format

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-02

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

Test Case Description: This test case verifies if the system can display post reach and user engagement metrics such as likes, comments, and shares.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User has made posts on the platform

Test Data: Data regarding post reach and engagement metrics

Test Steps:

1. Log into user account
2. Navigate to "Analytics and Insights" section
3. Select "Post Reach and Engagement Metrics"

Postconditions:

- System displays post reach and engagement data

Expected Result: User can view data on post reach and user engagement metrics

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-03

Test Case Title: Verify users can track audience growth trends

Test Case Description: This test case is to check if the user is able to track growth trends of their audience over a certain period.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is registered and has audience/followers

Test Data: Data showing audience growth over time

Test Steps:

1. Log into user account
2. Navigate to "Analytics and Insights" section
3. Select "Audience Growth Trends"

Postconditions:

- System displays audience growth trends

Expected Result: User can view data representing audience growth trends

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-04

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

Test Case Description: This test case will verify if analytics data is displayed in a user-friendly format that is

easy to understand.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in and navigates to the "Analytics and Insights" section

Test Data: No test data needed

Test Steps:

1. Log into user account
2. Navigate to "Analytics and Insights" section
3. Browse through various analytic metrics

Postconditions:

- User is able to easily comprehend analytics

Expected Result: Analytics is displayed in a user-friendly format

Severity: Major

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: TC-05

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

Test Case Description: This test case will verify if users have the options to filter data by time range and content type.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User has published various types of content over time

Test Data: Data based on different content types and time ranges

Test Steps:

1. Log into user account
2. Navigate to "Analytics and Insights" section
3. Choose filters to sort by time range and content type

Postconditions:

- System should display data corresponding to the selected filters

Expected Result: User can filter and view data based on time range and content type

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 13

Project name -

Social Media Platform

Project description -

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

Feature name -

Events and Groups

Feature description -

Events and groups enable users to organize and participate in events, communities, or interest-based groups on the platform. Users can create events, RSVP to invitations, join groups, and participate in discussions,

fostering community engagement and connection around shared interests.

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case ID: TC-001

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

Test Case Description: This test case verifies that the user has the ability to create and join events through the platform

Test Suite: Events and Groups Features

Test Priority: High

Preconditions:

- User has an account and is logged in
- User has access to the create event page

Test Data: User credentials, event details

Test Steps:

1. Log in with user credentials
2. Navigate to the create event page
3. Enter event details and save
4. Navigate to the event page after creation
5. Click on the "Join" button

Postconditions:

- Event appears in user's list of joined events

- User appears in event's list of attendees

Expected Result: Event is created and user is able to join it

Severity: High

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-002

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

Test Case Description: This test case verifies that the user can search and join interest-based groups in the platform.

Test Suite: Events and Groups Features

Test Priority: High

Preconditions:

- User has an account and is logged in
- User has access to the search and join group page

Test Data: User credentials, group search keywords

Test Steps:

1. Log in with user credentials
2. Navigate to the search page
3. Enter group search keywords and initiate search
4. Click on the group from the search result
5. Click on the "Join" button on group page.

Postconditions:

- The group appears in the list of groups the user has joined
- User appears in the group's list of members

Expected Result: User is able to join the group

Severity: High

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-003

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

Test Case Description: This test case verifies that the user can create and join discussions organized by topics within a group.

Test Suite: Events and Groups Features

Test Priority: Medium

Preconditions:

- User has an account, is logged in, and is a member of at least one group
- User has access to the discussion board of the group

Test Data: User credentials, discussion topic, discussion message

Test Steps:

1. Log in with user credentials
2. Navigate to the group page
3. Enter discussion topic and message then save
4. Navigate to the discussion board and find the newly created discussion
5. Join the discussion by posting a reply

Postconditions:

- The discussion appears on the group's discussion board

Expected Result: User can start and join a discussion around a topic within the group

Severity: Medium

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-004

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

Test Case Description: This test case verifies that event details such as date, time, location and description can be added when creating an event.

Test Suite: Events and Groups Features

Test Priority: High

Preconditions:

- User has an account, is logged in and on the event page
- User has access to create events

Test Data: User credentials, event details such as (title, date, time, location, description)

Test Steps:

1. Log in with user credentials
2. Navigate to the event creation page
3. Fill in the event details (title, date, time, location, description)
4. Save the event
5. Navigate to the list of created events and open the event
6. Verify the event details

Postconditions:

- The event is created with the required details in the user's event list

Expected Result: Event details are all present and correctly displayed

Severity: High

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC-005

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

Test Case Description: This test checks if users can RSVP to events and also see the list of attendees for the event on the platform.

Test Suite: Events and Groups Features

Test Priority: High

Preconditions:

- User has an account, is logged in and has at least one created event
- User has access to the event where he/she desires to RSVP

Test Data: User credentials, event details

Test Steps:

1. Log in with user credentials
2. Navigate to the desired event
3. Click on the "RSVP" button
4. Navigate to the list of attendees

Postconditions:

- User is listed as an attendee of the event
- The attendee list is updated with the new RSVP

Expected Result: Users can successfully RSVP to events and the list of attendees is updated

Severity: High

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Project name -

Social Media Platform

Project description -

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

Feature name -

Monetization Options

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TC_001

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

Test Case Description: This test case checks whether the functionality allowing users to apply for various

monetization options is working as intended.

Test Suite: Monetization Suite

Test Priority: High

Preconditions:

- User is logged in
- User's profile meets eligibility requirements for monetization

Test Data: User login credentials

Test Steps:

1. Log into the platform.
2. Access the monetization options from the settings.
3. Proceed with the application for monetization.

Postconditions:

- User has applied for monetization options

Expected Result: The application for the monetization proceeds without any issues.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_002

Test Case Title: Verify that monetization options include advertising revenue and sponsored content.

Test Case Description: This test case ensures that the monetization options offered by the platform include earning from advertising revenue and sponsored content.

Test Suite: Monetization Suite

Test Priority: High

Preconditions:

- User is logged in
- User's profile has been approved for monetization

Test Data: User login credentials

Test Steps:

1. Log into the platform.
2. Access the monetization settings from the user profile.
3. Check for the availability of advertising revenue and sponsored content options.

Postconditions:

- User has verified monetization options

Expected Result: The options for earning via advertising revenue and sponsored content are displayed and accessible.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_003

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

Test Case Description: This test case confirms if users can receive their earnings generated from monetization directly through the platform.

Test Suite: Monetization Suite

Test Priority: High

Preconditions:

- User is logged in

- The user's profile is monetized and has generated revenue

Test Data: User login credentials

Test Steps:

1. Log into the platform.
2. Go to the earnings summary in the user profile.
3. Request for a payout.

Postconditions:

- User has successfully requested for a payout

Expected Result: The user is able to request a payment of their earnings through the platform without any errors.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_004

Test Case Title: Verify the availability of detailed earnings reports for users.

Test Case Description: This test case checks if users have access to detailed reports showcasing their earnings from the platform's monetization.

Test Suite: Monetization Suite

Test Priority: High

Preconditions:

- User is logged in
- The user's profile is monetized

Test Data: User login credentials

Test Steps:

1. Log into the platform.
2. Open the earnings summary from the user profile section.
3. Review the earnings report.

Postconditions:

- User has viewed the detailed earnings report

Expected Result: A detailed earnings report, showing the breakdown of user's revenue is displayed.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_005

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

Test Case Description: This test case aims to confirm that the platform clearly states the requirements for users to be eligible for monetization.

Test Suite: Monetization Suite

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: User login credentials

Test Steps:

1. Log into the platform.
2. Navigate to the settings.
3. Open Monetization options.

4. Review the eligibility requirements.

Postconditions:

- User is aware of the monetization eligibility requirements

Expected Result: The user can view a clearly stated list of requirements needed for monetization.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Data - 15

Project name -

Social Media Platform

Project description -

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

Feature name -

Explore or Discover Tab

Feature description -

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

Userstory title -

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

topics.

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TC001

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

Test Case Description: This test case is designed to verify that trending and popular content such as posts, photos, and videos from various accounts are correctly displayed on the Explore tab.

Test Suite: Functional Testing

Test Priority: High

Preconditions:

- The user must be logged in.
- The System/server is functional.
- The Explore tab exists.

Test Data: User login credentials.

Test Steps:

1. Open the social media platform.
2. Log in with user credentials.
3. Navigate to the Explore tab.
4. Verify that trending and popular content is displayed.

Postconditions:

- Log out the user.

Expected Result: Trending and popular content should be displayed on the Explore tab.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that the Explore tab shows personalized recommendations based on users' interests.

Test Case Description: This test case is designed to verify that the Explore tab provides personalized recommendations based on each particular user's interests, engagement history, and social connections.

Test Suite: Functional Testing

Test Priority: Medium

Preconditions:

- The user must be logged in.
- The System/server is functional.
- The Explore tab exists.

Test Data: User login credentials.

Test Steps:

1. Open the social media platform.
2. Log in with user credentials.
3. Navigate to the Explore tab.
4. Verify that personalized recommendations are shown based on the user's interest, engagement history, and social connections.

Postconditions:

- Log out the user.

Expected Result: Personalized recommendations should be displayed based on user's interests.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003

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

Test Case Description: This test case is designed to ensure that users can use filters to sort the content by category or topic on the Explore tab.

Test Suite: Acceptance Testing

Test Priority: High

Preconditions:

- The user must be logged in.
- The System/server is functional.
- The Explore tab exists.

Test Data: User login credentials.

Test Steps:

1. Open the social media platform.
2. Log in with user credentials.
3. Navigate to the Explore tab.
4. Apply filters to sort content by a specific category or topic.
5. Verify that the content is sorted according to the applied filters.

Postconditions:

- Log out the user.

Expected Result: The content should be properly sorted based on the selected category or topic.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004

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

Test Case Description: This test case is designed to ensure that the Explore tab updates regularly with fresh content.

Test Suite: System Testing

Test Priority: Medium

Preconditions:

- The user must be logged in.
- The Server/System is functional.
- The Explore tab exists.

Test Data: User login credentials.

Test Steps:

1. Open the social media platform.
2. Log in with user credentials.
3. Navigate to the Explore tab.
4. Note the displayed content.
5. Refresh the page after some time.
6. Verify the content is updated.

Postconditions:

- Log out the user.

Expected Result: The Explore tab should update regularly with fresh content.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005

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

Test Case Description: This test case is designed to ensure that users have the ability to save or follow any content they find interesting on the Explore tab.

Test Suite: Acceptance Testing

Test Priority: High

Preconditions:

- The user must be logged in.
- The Server/System is functional.
- The Explore tab exists.

Test Data: User login credentials.

Test Steps:

1. Open the social media platform.
2. Log in with user credentials.
3. Navigate to the Explore tab.
4. Select a piece of content.
5. Click the save or follow option.
6. Verify the content has been saved or followed.

Postconditions:

- Log out the user.

Expected Result: Users should be able to save or follow the selected content from 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:

Test Case 1:

Test Case ID: TC1_AC1

Test Case Title: Verify that the platform provides alt text for images.

Test Case Description: This test case is designed to ensure that each image uploaded to the platform has an associated alt text for enhancing the accessibility for visually impaired users.

Test Suite: Accessibility Features

Test Priority: High

Preconditions: User is logged in, user has uploaded a photo.

Test Data: User profile with uploaded photo.

Test Steps:

1. Log into user profile.
2. Navigate to photos section.
3. Select a photo and check for alt text.

Postconditions: Record the presence or absence of alt text.

Expected Result: Each image, when selected, should display alt text.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC2_AC2

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

Test Case Description: This test case ensures the platform's compatibility with screen readers, making the content accessible to visually impaired users.

Test Suite: Accessibility Features

Test Priority: High

Preconditions: User is logged in, a screen reader tool is installed.

Test Data: No test data needed.

Test Steps:

1. Log into the Social Media platform.
2. Navigate through various sections of the site using the screen reader.

Postconditions: Record the screen reader's output.

Expected Result: The screen reader should correctly narrate the content of all sections.

Severity: Critical

Priority: High

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC3_AC3

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

Test Case Description: This test case is designed to ensure that the platform includes keyboard shortcuts to enhance navigation for users with motor disabilities.

Test Suite: Accessibility Features

Test Priority: High

Preconditions: User is logged in.

Test Data: No test data needed.

Test Steps:

1. Log into the Social Media platform.
2. Try to navigate through the site using keyboard shortcuts.

Postconditions: Record the functionality of keyboard shortcuts.

Expected Result: The user should be able to navigate the entire site using only keyboard shortcuts.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC4_AC4

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

Test Case Description: This test case checks whether the platform offers adjustable font sizes and color contrast to enhance readability for users with visual impairments.

Test Suite: Accessibility Features

Test Priority: High

Preconditions: User is logged in.

Test Data: No test data needed.

Test Steps:

1. Log into the Social Media platform.
2. Navigate to the settings/options section.
3. Look for color contrast and font size adjustment features.

Postconditions: Record the presence or absence of color contrast and font size adjustment features.

Expected Result: User settings should include color contrast and font size options.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC5_AC5

Test Case Title: Verify that the accessibility features are clearly documented.

Test Case Description: This test ensures the platform provides clear documentation for its accessibility features, offering guidance for users with disabilities.

Test Suite: Accessibility Features

Test Priority: High

Preconditions: User is logged in.

Test Data: No test data needed.

Test Steps:

1. Log into the Social Media platform.
2. Navigate to the help or FAQ section.
3. Look for clear documentation on accessibility features.

Postconditions: Record the clarity and understandability of accessibility feature documentation.

Expected Result: The help or FAQ section should contain understandable information about each accessibility feature.

Severity: Critical

Priority: High

Type of 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:

Test Case 1:

- Test Case ID: TC01_SMP
- Test Case Title: Verify that content moderation algorithms filter inappropriate content
- Test Case Description: This test case checks the functioning of content moderation algorithms by trying to post inappropriate content.
- Test Suite: Safety and Moderation
- Test Priority: High
- Preconditions: User is logged in, user makes an attempt to post inappropriate content
- Test Data: Inappropriate text or image
- Test Steps: 1. Log into the platform. 2. Attempt to post inappropriate content.
- Postconditions: System flags the inappropriate content and blocks it from being posted.
- Expected Result: Inappropriate content is not posted, the user receives a warning message.
- Severity: Blocker
- Priority: High
- Type of Testing: System Testing, Functional Testing
- Test Case Approach: Negative

Test Case 2:

- Test Case ID: TC02_SMP
- Test Case Title: Verify that users can report harmful or inappropriate content
- Test Case Description: This test case ensures users can report content on the platform that they find harmful or inappropriate.
- Test Suite: Safety and Moderation
- Test Priority: Medium
- Preconditions: User is logged in, user finds a post that violates guidelines
- Test Data: Report options (spam, nudity, violence, etc.)
- Test Steps: 1. Log into the platform. 2. Find a post violating guidelines. 3. Click the report button, select a

reason and submit.

- Postconditions: System registers the report for review by moderators
- Expected Result: User receives confirmation that the report has been registered.
- Severity: Major
- Priority: Medium
- Type of Testing: System Testing, Functional Testing
- Test Case Approach: Positive

Test Case 3:

- Test Case ID: TC03_SMP
- Test Case Title: Verify that blocking capabilities are available to users
- Test Case Description: This test case will check the functionality of blocking other users.
- Test Suite: Safety and Moderation
- Test Priority: Medium
- Preconditions: User is logged in, User identifies another user to block
- Test Data: Username or ID of the user to block
- Test Steps: 1. Log into the platform. 2. Go to another user's profile. 3. Click the block button.
- Postconditions: Blocked user's activities don't appear in feed or search, blocked user cannot message the user who blocked them
- Expected Result: User is notified that the selected user has been blocked.
- Severity: Major
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 4:

- Test Case ID: TC04_SMP
- Test Case Title: Verify that comment filters prevent offensive language
- Test Case Description: This test case checks the efficiency of comment filters in preventing offensive language from being published.
- Test Suite: Safety and Moderation
- Test Priority: Medium
- Preconditions: User is logged in, User attempts to post a comment with offensive language
- Test Data: Text with offensive language
- Test Steps: 1. Log into the platform. 2. Open a post. 3. Attempt to comment using offensive language.
- Postconditions: System blocks the comment, user receives a warning about offensive language.
- Expected Result: Comment is not posted, user receives a message informing them the comment was blocked due to offensive language.
- Severity: Major
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Negative

Test Case 5:

- Test Case ID: TC05_SMP
- Test Case Title: Verify that community guidelines are enforced consistently
- Test Case Description: This test case verifies that actions violating the platform's community guidelines are flagged and suitable action is taken.
- Test Suite: Safety and Moderation
- Test Priority: High
- Preconditions: User is logged in, User behaves in a way violating community guidelines

- Test Data: Behaviour, posts, or comments violating community guidelines
 - Test Steps: 1. Log into the platform. 2. Act in a manner violating community guidelines.
 - Postconditions: System detects the violation and takes appropriate action, such as issuing a warning, blocking the content or suspending the account.
 - Expected Result: User receives a warning message or notice of action taken due to the violation of community guidelines.
 - Severity: Blocker
 - Priority: High
 - Type of Testing: System Testing, Functional Testing
 - Test Case Approach: Negative
-

Data - 18

Project name -

Social Media Platform

Project description -

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

Feature name -

Multi-Language Support

Feature description -

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

experience across global audiences.

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1

Test Case ID: TC_MLS_001

Test Case Title: Verify that users can select their preferred interface language

Test Case Description: This test case ensures that the system allows users to select their preferred language for the interface.

Test Suite: Language Selection Module

Test Priority: High

Preconditions: User is signed in, User is on the language selection page

Test Data: Various available languages

Test Steps:

1. Navigate to setting page
2. Select language selection tab
3. Choose a preferred language
4. Save changes

Postconditions: Ensure language preference is saved

Expected Result: System updates the interface to the selected language

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_MLS_002

Test Case Title: Verify the content translation feature

Test Case Description: This test case checks if posts and messages can be translated into different languages based on user preference.

Test Suite: Content Translation Module

Test Priority: High

Preconditions: User is signed in, User viewed a post / message, Selected language preference

Test Data: Posts or messages in different languages

Test Steps:

1. View a post / message in another language
2. Click on translate button
3. Select the preferred language for translation
4. View translated post / message

Postconditions: The post / message is translated and displayed in selected language

Expected Result: Posts or messages are translated into user's preferred language successfully

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_MLS_003

Test Case Title: Verify the saved language preference

Test Case Description: This test case checks if saved language preferences are applied consistently across the platform.

Test Suite: Language Preference Module

Test Priority: Medium

Preconditions: User is signed in, Language preference is already selected

Test Data: No test data needed

Test Steps:

1. Log out from the app
2. Log in back to the app
3. Check if the interface is displayed in the previously selected language

Postconditions: Verify if preference is persistent across sessions

Expected Result: After logging back into the app, the system interface should still be displayed in the user's previously selected language.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC_MLS_004

Test Case Title: Verify the range of languages offered by the platform

Test Case Description: This test case checks the available languages offered by the system for interface and translation.

Test Suite: Multi-Language Support Suite

Test Priority: Medium

Preconditions: User is signed in, User is on the language selection page

Test Data: No test data needed

Test Steps:

1. Navigate to the language selection page
2. Check the available languages for selection
3. Repeat the above steps for the translation feature

Postconditions: Confirm the system supports a wide range of languages

Expected Result: A wide range of languages is available for selection and translation.

Severity: Blocker

Priority: Medium

Type of Testing: Functional testing, Compatibility testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC_MLS_005

Test Case Title: Verify the functionality when unsupported language is selected

Test Case Description: This test case is for when a user tries to select or translate into an unsupported language, it should handle gracefully.

Test Suite: Negative Test Cases Suite

Test Priority: Low

Preconditions: User is signed in, User is on the language selection / translation page

Test Data: Language not supported by the platform

Test Steps:

1. Try to select an unsupported language for interface

2. Try to translate a post/message into unsupported language

Postconditions: An appropriate error message should be displayed

Expected Result: The system does not crash and provides an appropriate error message.

Severity: Critical

Priority: Low

Type of Testing: Functional Testing, Negative 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:

Test Case 1:

Test Case ID: TC_CPI_01

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

Test Case Description: This test case will validate the functionality of linking accounts with other social media platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is registered on the platform
- User has an account on other social media platforms

Test Data: User login credentials for other platforms

Test Steps:

1. Login to the application
2. Navigate to "Account Settings"
3. Click on "Link other accounts"
4. Enter other social media platform details
5. Click "Link Account"

Postconditions:

- Verify that account linking message is displayed
- Verify that the linked account is displayed in the user's profile

Expected Result: The user's account is successfully linked with other platforms.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_CPI_02

Test Case Title: Verify that users can cross-post content to other platforms.

Test Case Description: This test case verifies that users can share their posts from our platform to their linked social media platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User has linked their account with other platforms
- User's post is ready to be posted on our platform

Test Data: Test post to be shared

Test Steps:

1. Login to the application
2. Create a post
3. Click on "Share to other platforms"
4. Select the platforms to share the post
5. Click "Share"

Postconditions:

- Verify successful sharing message is displayed
- Verify that the post is shared on the selected platforms

Expected Result: User's post is successfully cross-posted to other platforms.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_CPI_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 login to the application using the SSO feature.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is already registered on our platform
- User's social account is linked with our platform

Test Data: User's social account credentials

Test Steps:

1. Go to the application login page
2. Click on "Sign in with SSO"
3. Choose the social account to use
4. Enter the social account credentials
5. Confirm login

Postconditions:

- The user is logged into the application.

Expected Result: The user is successfully signed in using SSO.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_CPI_04

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

Test Case Description: This test case checks if users can integrate their account with external media sources (eg. Google photos, Dropbox etc.).

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is registered on our platform.
- User has an account with external media sources.

Test Data: Login credentials of external media sources.

Test Steps:

1. Login to the application.
2. Go to "Account Settings".
3. Click "Link external media sources".
4. Enter the details of media source account.
5. Click "Link account".

Postconditions:

- Account linking message is displayed
- Linked external media source appears in user's profile.

Expected Result: The user can successfully link their account with external media sources.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_CPI_05

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

Test Case Description: This test case verifies that sharing content across different platforms from our social media platform is seamless with all features functioning as expected and content appearing qualitatively unabridged on other platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User has linked their account with other social platforms
- User has content ready to share across platforms

Test Data: Test content to be shared across platforms

Test Steps:

1. Login to the application
2. Select the content to share
3. Click "Share to other platforms"
4. Choose the platforms to share the content
5. Click "Share"

Postconditions:

- Confirmation message for successful content sharing is displayed
- Verify the shared content in the linked platforms

Expected Result: Content shared from our platform appears in the selected platforms as expected, maintaining

quality and format.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Data - 20

Project name -

Social Media Platform

Project description -

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

Feature name -

Gamification and Rewards

Feature description -

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

Userstory title -

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

Acceptance criteria -

['Users can earn points and badges for their activities.', 'Leaderboards display top users based on points or

achievements.', 'Challenges are available for users to complete and earn rewards.', 'Rewards can include virtual goods, discounts, or exclusive content.', 'Users can view their progress and achievements in a dedicated section.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_GAM_001

Test Case Title: Verify that users can earn points and badges for their activities on the platform.

Test Case Description: This test case validates the functionality that allows users to earn points and badges based on their activities on the social media platform.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions: The user is registered and logged in

Test Data: User account credentials, preferred activities on the platform

Test Steps:

1. Login to the user account.
2. Perform several activities such as posting, commenting, liking, etc.
3. Monitor the accumulated points and badges for each activity.

Postconditions: Record the points and badge details in user's profile

Expected Result: The points and badges on the user profile increase based on activities.

Severity: Major

Priority: High

Type of Testing: Functional Testing; Regression Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_GAM_002

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

Test Case Description: This test case validates whether the leaderboard correctly displays top users based on their accumulated points or achievements.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions: There are various active users on the platform

Test Data: Points or achievement details of several users

Test Steps:

1. Login to the user account.
2. Perform several activities to earn points or achievements.
3. Navigate to the leaderboard section.
4. Check the leaderboard for user's ranking.

Postconditions: Validate the user's ranking regularly on the leaderboard.

Expected Result: The leaderboard is updated and displays the top users in the correct order based on points or achievements.

Severity: Major

Priority: High

Type of Testing: Functional Testing; Regression Testing

Test Case Approach: Positive

The remaining test cases can follow similar templates for validating the Challenges, Rewards, and Progress and Achievements sections, under the same Test Suite, while alternating between positive and negative test case approaches where necessary. The priority and severity for each test case will largely depend on the specific acceptance criteria and its impact on user experience should it fail.

Data - 21

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Creation

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TC-001

Test Case Title: Verify that users can enter a task title, description, due date, and priority level

Test Case Description: This test case ensures that users can successfully create tasks.

Test Suite: Task Creation

Test Priority: High

Preconditions: The user must be logged into the task management tool.

Test Data: Task title - "Test Task", Description - "This is a test task.", Due date - "12/12/2022", Priority level - "High"

Test Steps: 1. Login to the task management tool, 2. Click on "Create task" button, 3. Enter task details, 4. Click "Save"

Postconditions: The task is displayed in the task list.

Expected Result: The task is successfully created and displayed in the task list.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC-002

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

Test Case Description: This test case ensures that users can assign tasks to team members.

Test Suite: Task Creation

Test Priority: High

Preconditions: The user must be logged into the task management tool, Team members must already be added to the project

Test Data: Task assigned to - "Team Member A"

Test Steps: 1. Login to the task management tool, 2. Click on "Create task" button, 3. Enter task details and assign to specific member, 4. Click "Save"

Postconditions: The assigned team member receives a notification.

Expected Result: The task is successfully assigned and the team member receives a notification.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC-003

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

Test Case Description: This test case ensures that created tasks are saved and correctly displayed.

Test Suite: Task Creation

Test Priority: High

Preconditions: The user must be logged into the task management tool, The task must have been created

Test Data: No test data needed

Test Steps: 1. Login to the task management tool, 2. Navigate to task list

Postconditions: None

Expected Result: The task is displayed in the list of tasks.

Severity: Blocker

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC-004

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

Test Case Description: This test case checks if users can modify existing tasks.

Test Suite: Task Creation

Test Priority: High

Preconditions: The user must be logged into the task management tool, The task must have been created

Test Data: Updated Task title - "Updated test task", Updated Description - "This is an updated test task."

Test Steps: 1. Login to the task management tool, 2. Navigate to the specific task in the task list, 3. Click on "edit" button, 4. Update task details, 5. Click "Save"

Postconditions: The edited task is updated in the task list.

Expected Result: The task details are successfully updated and reflected in the task list.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC-005

Test Case Title: Verify that the system validates required fields such as task title

Test Case Description: This test case ensures that form validation works correctly while creating a task, preventing the creation of a task without a title.

Test Suite: Task Creation

Test Priority: High

Preconditions: The user must be logged into the task management tool

Test Data: No task title provided

Test Steps: 1. Login to the task management tool, 2. Click on "Create task" button, 3. Leave the task title field empty and fill up other fields, 4. Click "Save"

Postconditions: The user gets an error message

Expected Result: An error message appears saying "Task title cannot be empty."

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative

Data - 22

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Assignment

Feature description -

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

Userstory title -

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

Acceptance criteria -

['Users can select team members to assign tasks.', 'Assignees receive notifications about their tasks.', 'The

system should display assigned tasks in the assignee's task list.", 'Users can reassign tasks to different team members.', 'Task assignment history should be tracked.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT001

Test Case Title: Verify that users can select team members to assign tasks

Test Case Description: Testing the functionality that allows users to assign tasks to specific team members

Test Suite: Task Assignment

Test Priority: High

Preconditions: The user account is logged in, Task is created, At least one team member added into the user's team

Test Data: Task details and team member name

Test Steps: 1. Login into account, 2. Navigate to Task assignment area, 3. Select a task, 4. Select a team member from the drop-down, 5. Click on the Assign button.

Postconditions: Task information is updated

Expected Result: The selected team member is shown as assignee in the task assignment list

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT002

Test Case Title: Verify that assignees receive notifications about their tasks

Test Case Description: Testing if notifications are sent to assignees when a task is assigned to them

Test Suite: Task Assignment

Test Priority: High

Preconditions: Task is assigned to a team member, The notification feature is enabled

Test Data: No test data needed

Test Steps: 1. Assign a task, 2. Navigate to assignee's dashboard, 3. Check for the notification

Postconditions: Notification information is displayed on assignee's dashboard

Expected Result: Assignee received notification about their assigned task

Severity: Major

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT003

Test Case Title: Verify that assigned tasks are displayed on the assignee's task list

Test Case Description: Checking if assigned tasks are visible in the assignee's task dashboard

Test Suite: Task Assignment

Test Priority: High

Preconditions: User is logged in as assignee, Task(s) are assigned

Test Data: No test data needed

Test Steps: 1. Login into account as an assignee, 2. Navigate to task dashboard, 3. Check assigned tasks

Postconditions: Task list is updated

Expected Result: The assigned tasks appear in the assignee's task list

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT004

Test Case Title: Verify that users can reassign tasks to different team members

Test Case Description: Testing the ability to switch task responsibility among team members

Test Suite: Task Assignment

Test Priority: High

Preconditions: Task is already assigned to one team member

Test Data: Task and recipient team member's details

Test Steps: 1. Login into account, 2. Navigate to task assignment area, 3. Select the task, 4. Reassign the task to a different team member

Postconditions: Task assignment information is updated

Expected Result: The task assignee status is updated with the newly assigned team member

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT005

Test Case Title: Verify that task assignment history is tracked

Test Case Description: Testing if a task's assignment history is maintained properly

Test Suite: Task Assignment

Test Priority: High

Preconditions: There are previous assignment history for a task

Test Data: No test data needed

Test Steps: 1. Login into account, 2. Navigate to task assignment history, 3. Check the history of a previously assigned and reassigned task

Postconditions: None

Expected Result: The task's assignment history reflects all the previous and current assignees correctly

Severity: Major

Priority: High

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:

Test Case 1:

Test Case ID: TC001

Test Case Title: Verify that users can set reminders for tasks based on deadlines.

Test Case Description: This test verifies that users can set reminders for upcoming task deadlines.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions: User must have tasks scheduled with set deadlines.

Test Data: Tasks with set deadlines

Test Steps:

1. Log into the task management tool.
2. Create a new task with a deadline.
3. Set a reminder for the task deadline.
4. Save the task and logout.

Postconditions: Verify the task reminder is saved successfully.

Expected Result: Task reminder must be successfully set and saved to match the task deadline.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that reminders can be customized by urgency level.

Test Case Description: This test verifies that users can customize reminders based on the urgency of tasks.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions: User must have tasks scheduled with set deadlines and different levels of urgency.

Test Data: Tasks with set deadlines and varied urgency level

Test Steps:

1. Log into the task management tool.
2. Create a new task with a deadline and urgency level.
3. Set a customized reminder for the task based on its urgency level.
4. Save the task and logout.

Postconditions: Verify the customized task reminder is saved successfully.

Expected Result: Reminder is successfully set and saved based on the input urgency level.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003

Test Case Title: Verify that the system sends notifications for upcoming deadlines.

Test Case Description: This test verifies that the system successfully sends notification reminders for

upcoming task deadlines.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions: User must have task scheduled with a set reminder.

Test Data: Set reminder for a specific task

Test Steps:

1. Log into the task management tool.
2. Wait until the set reminder time.
3. Check if notification for upcoming deadline has been received.

Postconditions: Verify the notification received matches the set reminder.

Expected Result: Notification for upcoming task deadline must successfully be received.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004

Test Case Title: Verify that users can choose the delivery method for their reminders.

Test Case Description: This test verifies that users can select their desired method of receiving reminders (e.g., email, in-app).

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions: User must have tasks scheduled with set deadlines and reminders.

Test Data: Users' chosen notification delivery method (email or in-app)

Test Steps:

1. Log into the task management tool.
2. Select preferred delivery method for reminders.
3. Set reminder for a task.
4. Verify if reminder is received through the selected delivery method.

Postconditions: Verify the delivery method for reminders is saved successfully.

Expected Result: Reminder must be successfully delivered through the selected delivery method.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005

Test Case Title: Verify that reminder settings can be adjusted or disabled.

Test Case Description: This test verifies that users can adjust the settings of their task reminders or disable them completely.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions: User must have tasks scheduled with set reminders.

Test Data: Set reminders with settings to adjust or disable

Test Steps:

1. Log into the task management tool.
2. Locate an existing reminder.
3. Adjust settings of the reminder or disable the reminder.
4. Save changes and logout.

Postconditions: Verify the changes to the settings of reminders are saved successfully.

Expected Result: Reminder must be successfully adjusted or disabled based on the changes made.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 24

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

Users can prioritize tasks based on their urgency and importance, allowing them to focus on high-priority tasks and manage their workload effectively. Tasks can be categorized into different priority levels, such as high, medium, or low, based on their importance to the project or team.

Userstory title -

As a user, I want to prioritize tasks so that I can manage my workload effectively.

Acceptance criteria -

['Users can set priority levels for tasks (High, Medium, Low).', 'Tasks can be sorted based on priority levels.', 'Users can change the priority level of a task.', 'The system visually distinguishes tasks based on priority.', 'Tasks without a specified priority default to Medium.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC001

Test Case Title: Verify that users can set priority levels for tasks

Test Case Description: This test case verifies if the user can assign High, Medium or Low priorities to their tasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged into the system
- At least one task is created

Test Data: UserCreateTask1, PriorityLevelHigh

Test Steps:

1. Log into the system
2. Navigate to the tasks page
3. Open UserCreateTask1
4. Set task priority to High
5. Save changes

Postconditions:

- Task is updated with the new priority level

Expected Result: UserCreateTask1 shows as 'High' Priority

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that tasks are sorted based on priority levels

Test Case Description: This test case verifies if the tasks are ordered based on their assigned priority levels.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged into the system
- Multiple tasks with different priorities exist

Test Data: UserCreateTask1, UserCreateTask2, UserCreateTask3

Test Steps:

1. Log into the system
2. Navigate to the tasks page
3. Sort tasks by priority

Postconditions:

- Tasks are sorted by priority

Expected Result: Tasks show up in the order of High, Medium and Low priority

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003

Test Case Title: Verify that users can change the priority level of a task

Test Case Description: This test case confirms if the user can change the previously set task priority.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged into the system
- At least one task with set priority exists

Test Data: UserCreateTask1, PriorityLevelLow

Test Steps:

1. Log into the system
2. Navigate to the tasks page
3. Open UserCreateTask1
4. Change task priority to Low
5. Save changes

Postconditions:

- Task is updated with the new priority level

Expected Result: UserCreateTask1 shows as 'Low' priority

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004

Test Case Title: Verify that the system visually distinguishes tasks based on priority

Test Case Description: This test validates if tasks with different priority levels are visually distinguishable.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged into the system
- Tasks with different priorities exist

Test Data: No test data needed

Test Steps:

1. Log into the system
2. Navigate to the tasks page
3. Observe the visual distinction of tasks based on priority

Postconditions:

- Tasks remain visually distinct based on priorities

Expected Result: Tasks are visually distinguishable based on their priority levels

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005

Test Case Title: Verify that tasks without a specified priority default to Medium

Test Case Description: This test ensures that tasks without a user-specified priority are automatically given a medium priority.

Test Suite: Task Management

Test Priority: Low

Preconditions:

- User is logged into the system
- Task without a set priority exists

Test Data: UserCreateTask4

Test Steps:

1. Log into the system
2. Navigate to the tasks page
3. Open UserCreateTask4

Postconditions:

- Task remains as Medium priority

Expected Result: UserCreateTask4 shows as 'Medium' priority

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 25

Project name -

Task Management Tool

Project description -

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

Feature name -

Subtasks

Feature description -

Tasks can be broken down into smaller subtasks, enabling users to divide complex tasks into manageable components. Subtasks can have their own due dates, assignees, and dependencies, providing granularity and clarity in task execution.

Userstory title -

As a user, I want to create subtasks so that I can break down complex tasks into manageable parts.

Acceptance criteria -

['Users can create subtasks under main tasks.', 'Subtasks can have their own due dates and assignees.', 'The system tracks the progress of each subtask.', 'Subtasks can be marked as complete independently of the main task.', 'Dependencies between subtasks can be set.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC001

Test Case Title: Verify that subtasks can be created under main tasks.

Test Case Description: This test case tests the creation of subtasks under a main task.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged into the Task Management tool.
- At least one main task is created.

Test Data: Main Task details, Subtask description

Test Steps:

1. From the tasks list, select a main task.
2. Click on the 'Create Subtask' button.
3. Fill in the necessary details and click 'Save'.

Postconditions:

- The subtask is saved under the main task.
- The subtask appears under the main task in the tasks list.

Expected Result: Subtask is successfully created and associated with the main task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that subtasks can have their due dates and assignees.

Test Case Description: This test case tests the ability to add due dates and assignees to a subtask.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged into the Task Management tool.
- At least one main task is created.

Test Data: Subtask description, Due date, Assignee

Test Steps:

1. From the tasks list, select a main task.
2. Click on the 'Add Subtask' button.
3. Fill in the necessary details, including due date and assignee, and click 'Save'.

Postconditions: Subtask is updated with due date and assignee.

Expected Result: The subtask is successfully updated with due date and assignee.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003

Test Case Title: Verify that the system tracks the progress of each subtask.

Test Case Description: This test case tests the ability of the system to track the progress of each subtask.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged into the Task Management tool.
- At least one main task with subtask is created.

Test Data: No test data needed.

Test Steps:

1. Select the main task with the subtasks.
2. Update progress for one or more subtasks.
3. Save the changes.

Postconditions: The updated progress is displayed on the subtask.

Expected Result: The system successfully tracks the progress of each subtask.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004

Test Case Title: Verify that subtasks can be marked as complete independently of the main task.

Test Case Description: This test case tests the ability to mark subtasks as complete independently of the main task.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged into the Task Management tool.
- At least one main task with subtask is created.

Test Data: No test data needed.

Test Steps:

1. Select the main task with the subtasks.
2. Choose a subtask and mark it as completed.

Postconditions: The subtask is marked as completed, regardless of the main task's status.

Expected Result: The subtask is successfully marked as completed.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005

Test Case Title: Verify that dependencies between subtasks can be set.

Test Case Description: This test case tests the ability to set dependencies between subtasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged into the Task Management tool.
- The main task contains at least two subtasks.

Test Data: No test data needed.

Test Steps:

1. Select the main task with the subtasks.
2. Choose one subtask and set another subtask as its dependency.
3. Save the changes.

Postconditions: The dependency is saved and reflected in the system.

Expected Result: Dependencies between subtasks can be successfully set.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 26

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Dependencies

Feature description -

Users can define dependencies between tasks, specifying the order in which tasks should be completed to ensure smooth workflow and project progression. Task dependencies help in identifying critical paths and managing interdependencies between tasks.

Userstory title -

As a user, I want to set task dependencies so that tasks are completed in the correct sequence.

Acceptance criteria -

['Users can set dependencies between tasks.', 'The system displays dependencies in the task details.', 'Tasks cannot be marked as complete if dependent tasks are not completed.', 'Users can update or remove dependencies as needed.', 'The system provides alerts or warnings if a dependent task is not completed.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT-1

Test Case Title: Verify that users can set dependencies between tasks

Test Case Description: This test case validates whether users are able to set dependencies between different tasks.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- user is logged in
- at least two tasks are created

Test Data: Task IDs

Test Steps:

1. Navigate to the task assignment page
2. Select one task and click on "add dependency"
3. Select another task as the dependent task

Postconditions:

- Task dependency is saved in the system
- User is returned to task assignment page

Expected Result: The dependency between the selected tasks is successfully saved.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-2

Test Case Title: Verify that the system displays dependencies in the task details

Test Case Description: This test case checks whether the dependencies between tasks are displayed in the task details.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- At least two interdependent tasks exist

Test Data: No test data needed

Test Steps:

1. Navigate to a task with dependencies
2. View task details

Postconditions:

- User remains on task details page

Expected Result: The dependent tasks are listed in the task details.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-3

Test Case Title: Verify that tasks cannot be marked as complete if dependent tasks are not completed.

Test Case Description: This test case asserts that tasks with incomplete dependent tasks cannot be marked as complete.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- There exists a task with an incomplete dependent task

Test Data: Task ID

Test Steps:

1. Navigate to the task with incomplete dependent task
2. Try to mark the task as complete

Postconditions:

- User stays on the task page

Expected Result: The system disallows marking the task as complete and warns the user about the incomplete dependent tasks.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case 4:

Test Case ID: TMT-4

Test Case Title: Verify that users can update or remove dependencies as needed

Test Case Description: This test case validates whether users can modify or delete previously added dependencies.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- A task with dependencies exists

Test Data: Task ID

Test Steps:

1. Navigate to a task with dependencies
2. Click on 'Edit dependencies'
3. Remove or change the dependent tasks

Postconditions:

- Changes are saved in the system
- User is redirected back to task details page

Expected Result: The alterations applied to the task dependencies are successfully stored.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-5

Test Case Title: Verify that the system provides alerts or warnings if a dependent task is not completed.

Test Case Description: This test case checks whether a warning or alert is generated if an incomplete dependent task exists.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- exists a task with an incomplete dependent task

Test Data: No test data needed

Test Steps:

1. Navigate to the task page
2. View the task details

Postconditions:

- User stays on task details page

Expected Result: An alert or warning about incomplete dependent tasks is displayed.

Severity: Major

Priority: High

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:

Test Case 1:

Test Case ID: TMT_01

Test Case Title: Verify that users can add comments to tasks.

Test Case Description: This test case is designed to confirm that the system enables users to add comments to existing tasks.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- User is a part of a project.
- A task has been created.

Test Data: No test data needed.

Test Steps:

1. Navigate to a specific task.
2. Click on the comment section.
3. Type a comment.
4. Press the "Submit" button.

Postconditions:

- Comment is saved.
- Comment appears under the task.

Expected Result: The comment is visible under the task.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT_02

Test Case Title: Verify that comments can include text, attachments, and links.

Test Case Description: This test case ensures that the system allows users to attach files and enter URLs in their comments.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- User is a part of a project.
- A task has been created.

Test Data: Sample text, attachment, URL.

Test Steps:

1. Navigate to a specific task.
2. Click on the comment section.
3. Type a comment, attach a file, and enter a URL.
4. Press the "Submit" button.

Postconditions:

- Comment with attached file and URL is saved.
- Comment appears under the task.

Expected Result: The comment with its file and URL is successfully displayed under the task.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT_03

Test Case Title: Verify that users can reply to and edit their comments.

Test Case Description: This test case is to assert that the system allows users to interact with their comments after they've been posted.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- User is a part of a project.
- A task has been created.
- User had added a comment.

Test Data: Sample reply text, edited text of comment.

Test Steps:

1. Navigate to a specific task.
2. Find an existing comment.
3. Reply to the comment.
4. Edit the original comment.

Postconditions:

- Reply is displayed under original comment.
- Edits are updated in the comment.

Expected Result: The comment is updated and the reply is visible under it.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT_04

Test Case Title: Verify that the system notifies users of new comments.

Test Case Description: This test case ensures the software notifies users whenever there's a new comment on a task they are a part of.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- User is a part of a project.
- A task has been created.
- Another user adds a comment to the task.

Test Data: New comment text.

Test Steps:

1. User1 adds a comment to a task.
2. User2 receives a notification of the new comment.

Postconditions: User2 checks the notification and verifies the new comment on the task.

Expected Result: User2 gets notified about the new comment.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT_05

Test Case Title: Verify that comments are displayed in chronological order.

Test Case Description: This test case tests whether comments are correctly displayed by the order they were posted, with the most recent comment at the top.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is logged in.

- User is a part of a project.
- A task has been created.
- Several comments have been added with different timestamps.

Test Data: Multiple comments with timestamp.

Test Steps:

1. Navigate to a task with multiple comments.
2. Check the order of the comments based on timestamps.

Postconditions: None.

Expected Result: Comments are ordered from newest to oldest based on the timestamps.

Severity: Minor

Priority: Medium

Type of Testing: Non-Functional Testing

Test Case Approach: Positive

Data - 28

Project name -

Task Management Tool

Project description -

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

Feature name -

File Attachments

Feature description -

Users can attach relevant files, documents, and resources to tasks, ensuring that all necessary information is

accessible in one place. File attachments can include documents, spreadsheets, images, links, and other resources needed for task execution.

Userstory title -

As a user, I want to attach files to tasks so that all relevant information is easily accessible.

Acceptance criteria -

['Users can attach multiple files to a task.', 'Attachments can include various file types (documents, images, etc.).', 'Users can view and download attachments.', 'The system limits the size of attachments to prevent data overload.', 'Users can remove attachments if needed.']

Generated Test Cases:

Test Case 1

Test Case ID: TMT_F_001

Test Case Title: Verify that users can attach multiple files to a task.

Test Case Description: This test case focuses on validating the feature that allows users to attach more than one file to each task.

Test Suite: File Attachments Suite

Test Priority: High

Preconditions:

- A task is created.
- Necessary files for attachment are available.

Test Data: Various types of files

Test Steps:

1. Log in to the task management tool.
2. Open a task.
3. Click on the 'Attach Files' option.
4. Select multiple files to attach.

5. Confirm attachment of files.

Postconditions:

- The files have been attached to the task.
- The user returns to the task view.

Expected Result: Multiple files are successfully attached to the task shown in the attachments section.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TMT_F_002

Test Case Title: Verify that the system prevents attachment of files beyond the size limit.

Test Case Description: This case verifies if the system rejects files that exceed the allowed size limit to prevent data overload.

Test Suite: File Attachments Suite

Test Priority: High

Preconditions:

- A task is created.
- A file size exceeding the allowed size limit is available for attachment.

Test Data: An over-sized file

Test Steps:

1. Log in to the task management tool.
2. Open a task.
3. Click on the 'Attach Files' option.
4. Select an over-sized file to attach.

Postconditions:

- The user receives an error message.
- The user returns to the 'Attach Files' view.

Expected Result: The over-sized file is not attached to the task and an error message is displayed.

Severity: Critical

Priority: High

Type of Testing: Non-Functional Testing

Test Case Approach: Negative

Test Case 3

Test Case ID: TMT_F_003

Test Case Title: Verify that users can view and download attachments.

Test Case Description: This case ensures users can view the list of attached files and download each one.

Test Suite: File Attachments Suite

Test Priority: Medium

Preconditions:

- A file is attached to a task.

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool.
2. Open a task with attachments.
3. View the list of attached files.
4. Download an attached file.

Postconditions:

- The selected file has been downloaded onto the user's local system.

Expected Result: The attached file is viewable and downloadable successfully.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TMT_F_004

Test Case Title: Verify that users can remove attachments.

Test Case Description: This case validates if users can delete attached files from their tasks.

Test Suite: File Attachments Suite

Test Priority: Medium

Preconditions:

- A file has been attached to a task.

Test Data: No test data needed

Test Steps:

1. Log in to the task management tool.
2. Open a task with attachments.
3. Click on the 'Remove' option next to an attached file.

Postconditions:

- The selected associated file is removed from the task.
- The user sees a confirmation message.

Expected Result: The attached file is successfully removed from the task.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TMT_F_005

Test Case Title: Verify that users can attach different types of files to tasks.

Test Case Description: This case tests whether users can attach various file types to tasks (e.g., documents, images, etc.).

Test Suite: File Attachments Suite

Test Priority: High

Preconditions:

- A task is created.
- Various types of files to be attached are available.

Test Data: Various types of files (documents, images, etc.)

Test Steps:

1. Log in to the task management tool.
2. Open a task.
3. Click on the 'Attach Files' option.
4. Select different types of files to attach.
5. Confirm attachment of files.

Postconditions:

- The selected files are attached to the task.
- The user returns to the task view.

Expected Result: Different types of files are successfully attached to the task and these attachments are visible on the task page.

Severity: Major

Priority: High

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:

Test Case 1:

- Test Case ID: TM_O1
- Test Case Title: Verify the functionality of setting up recurring tasks at specified intervals.
- Test Case Description: This test case evaluates the ability to create recurring tasks on a daily, weekly, and monthly basis.
- Test Suite: Recurring Tasks
- Test Priority: High
- Preconditions: The user is registered and logged into the system.
- Test Data: A dummy task with sample description and attachments.
- Test Steps:
 1. Navigate to the task creation window in the application.
 2. Create a new task with suitable title, description, and attachments
 3. Choose to make this task recurring and set the frequency (daily, weekly, monthly).
 4. Save the task.
- Postconditions:
 1. The recurring task should be visible in the task list.
 2. The next instances of the task should be scheduled.
- Expected Result: The task is set to recur at the chosen interval without any errors.
- Severity: Critical
- Priority: High
- Type of Testing: Functional Testing, Integration Testing
- Test Case Approach: Positive

Test Case 2:

- Test Case ID: TM_O2
- Test Case Title: Verify that the system generates instances of recurring tasks automatically.
- Test Case Description: This test case checks if the system automatically generates instances of recurring tasks.

- Test Suite: Recurring Tasks
- Test Priority: High
- Preconditions: The user has created a recurring task.
- Test Data: No test data needed.
- Test Steps:
 1. Create a recurring task.
 2. Move ahead in the application's calendar.
 3. Check if the instances of the task have been created automatically.
- Postconditions: The future instances of recurring tasks are created and properly displayed on the calendar.
- Expected Result: All future instances of the task are generated automatically by the system.
- Severity: Critical
- Priority: High
- Type of Testing: Functional Testing, Integration Testing
- Test Case Approach: Positive

Test Case 3:

- Test Case ID: TM_O3
- Test Case Title: Verify that users can modify the recurrence pattern or cancel the recurrence of a task.
- Test Case Description: This test case evaluates the functionality that allows users to change the recurrence pattern or cancel a recurring task.
- Test Suite: Recurring Tasks
- Test Priority: Medium
- Preconditions: A recurring task has been created.
- Test Data: No test data needed.
- Test Steps:
 1. Go to the task that is already set as recurring.
 2. Modify its recurrence pattern or cancel the recurring setting.

3. Save the task.

- Postconditions:

1. If the recurrence pattern was modified, the modified recurrence should be reflected in future instances.

2. If recurrence was cancelled, no future instances should be generated.

- Expected Result: The system successfully updates the recurrence settings based on user input.

- Severity: Major

- Priority: High

- Type of Testing: Functional Testing, Regression Testing

- Test Case Approach: Positive

Test Case 4:

- Test Case ID: TM_O4

- Test Case Title: Verify that the system sends notifications for upcoming instances of recurring tasks.

- Test Case Description: This test case checks whether the system sends notifications for upcoming instances of recurring tasks.

- Test Suite: Recurring Tasks

- Test Priority: Medium

- Preconditions: A recurring task is set up and the notification feature is enabled.

- Test Data: No test data needed.

- Test Steps:

1. Create a recurring task with notification.

2. Wait until the time of notification.

- Postconditions: A notification should be received at the specified time.

- Expected Result: The system sends notifications successfully for upcoming instances of recurring tasks.

- Severity: Major

- Priority: Medium

- Type of Testing: Functional Testing, Integration Testing

- Test Case Approach: Positive

Test Case 5:

- Test Case ID: TM_O5

- Test Case Title: Verify that users can view a list of all instances of a recurring task.

- Test Case Description: This test case verifies the functionality that allows users to see all instances of a recurring task.

- Test Suite: Recurring Tasks

- Test Priority: low

- Preconditions: A recurring task is created.

- Test Data: No test data needed.

- Test Steps:

1. Create a recurring task.

2. Navigate to the recurring task list.

3. Select the created task.

- Postconditions: The list of all instances of the recurring task should be displayed.

- Expected Result: The user can view a list of all instances of the task without any issue.

- Severity: Minor

- Priority: Low

- Type of Testing: Functional Testing, Usability Testing

- Test Case Approach: Positive

Data - 30

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tags and Labels

Feature description -

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

Userstory title -

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

Acceptance criteria -

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

Generated Test Cases:

Test Case 1

Test Case ID: TMT01

Test Case Title: Verify that users can add tags and labels to tasks.

Test Case Description: This test case verifies if a user can add tags and labels, allowing customized categorization for each task.

Test Suite: Task Management

Test Priority: High

Preconditions:

- The user is logged in
- The user has created a task

Test Data: Example tag "High Priority", Example label "Marketing"

Test Steps:

1. User logs into account.
2. User selects a specific task.
3. User adds a tag and a label.
4. User saves changes.

Postconditions:

- The system should save changes

Expected Result: The added tags and labels should be displayed on the task.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TMT02

Test Case Title: Verify that tags can be customized and edited.

Test Case Description: This case verifies if tags and labels once added can be altered.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in
- A task with added tags exists

Test Data: Edited Tag: "Critical", Edited label: "Sales"

Test Steps:

1. User logs into account.
2. User selects a task with a tag.
3. User edits the tag and label.
4. User saves changes.

Postconditions:

-The system should save changes

Expected Result: The edited tags and labels appear on the task.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TMT03

Test Case Title: Verify that tasks can be filtered and sorted based on tags and labels.

Test Case Description: This test verifies if the tool allows tasks to be filtered and sorted via their respective tags and labels.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in
- There are multiple tasks created and tagged

Test Data: No test data needed

Test Steps:

1. User logs into account.

2. User applies filter or sort by tag or label.
3. System displays the sorted or filtered tasks.

Postconditions:

- The system should display tasks based on the filter or sort operation performed.

Expected Result: The tasks are shown properly filtered or sorted based on the chosen tag or label.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TMT04

Test Case Title: Verify that the system displays tags and labels prominently in the task list.

Test Case Description: This test case checks whether the tags and labels are clearly visible and identifiable in the task list.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in
- There are tasks with added tags and labels

Test Data: No test data needed

Test Steps:

1. User logs into their account.
2. User navigates to the task list.
3. User checks for visibility of labels and tags.

Postconditions:

-No postconditions required

Expected Result: The tasks with tags and labels are easily identifiable in the task list.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TMT05

Test Case Title: Verify that users can remove tags and labels if needed.

Test Case Description: This case verifies the functionality of removing tags and labels from tasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- The user is logged in
- The user has a task with tags and labels

Test Data: No test data needed

Test Steps:

1. User logs into account.
2. User selects a specific task with tags and labels.
3. User removes a tag and a label.
4. User saves changes.

Postconditions:

- The system should save changes

Expected Result: The removed tags and labels should not appear on the task.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 31

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Status Tracking

Feature description -

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

Userstory title -

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

Acceptance criteria -

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

Generated Test Cases:

Test Case 1

Test Case ID: TMT-01

Test Case Title: Verify that users can update the status of a task.

Test Case Description: The test case is designed to validate that users can successfully update the status of tasks with options like To Do, In Progress, Completed.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions: User must have at least one task created

Test Data: User credentials, Task details

Test Steps:

1. Log in to the task management tool using user credentials.
2. Open an existing task.
3. Change the status of the task to a different status.
4. Confirm the status change.

Postconditions: Update task list

Expected Result: The status of the selected task is updated consistently with the new status.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TMT-02

Test Case Title: Verify that the system displays the task status prominently in the task list.

Test Case Description: The test case is designed to verify if the task status is displayed prominently in the task

list in accordance with user interaction.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions: The user has at least one task created with a set status

Test Data: User credentials, Task list

Test Steps:

1. Log in to the task management tool using user credentials.
2. Navigate to the task list
3. Verify the status display for each task.

Postconditions: None needed

Expected Result: The task status for each task should be displayed prominently.

Severity: Major

Priority: High

Type of Testing: Usability Testing, Acceptance Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TMT-03

Test Case Title: Verify that users can view the progress percentage of tasks.

Test Case Description: The test case will validate if users can view progress percentage for tasks and if it is updated correctly when task status changes.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions: User must have at least one task created and in progress

Test Data: User credentials, Task list

Test Steps:

1. Log in to the task management tool using user credentials.
2. Navigate to the task list.
3. Open an existing task in progress.
4. Check the progress percentage displayed.

Postconditions: None needed

Expected Result: The task should show a correct progress percentage.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TMT-04

Test Case Title: Verify that the system provides notifications for status changes.

Test Case Description: The test case ensures that the system sends notifications whenever a task status changes.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions: User must have set notification settings and have at least one task with a status to be changed

Test Data: User credentials, Task list

Test Steps:

1. Log in to the task management tool using user credentials.
2. Open an existing task.
3. Change the task status.
4. Check for a notification regarding the task status change.

Postconditions: None

Expected Result: A notification regarding the change in task status should be received.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TMT-05

Test Case Title: Verify that task status can be filtered and sorted.

Test Case Description: The test case is designed to confirm that users can filter and sort tasks based on their status.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions: User must have multiple tasks with different status.

Test Data: User credentials, Task list

Test Steps:

1. Log in to the task management tool using user credentials.
2. Navigate to the task list.
3. Apply a filter by a specific status.
4. Verify the filtered task list.
5. Sort the tasks by status.
6. Confirm the correct sorting order.

Postconditions: Reset the filter and sorting order to original

Expected Result: Tasks should be correctly filtered and sorted based on their status.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Data - 32

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Filtering and Sorting

Feature description -

Users can filter and sort tasks based on various criteria such as due date, priority, assignee, or project. Filtering and sorting options help users focus on specific tasks or view tasks from different perspectives, enhancing task management and organization.

Userstory title -

As a user, I want to filter and sort tasks so that I can focus on specific tasks or view tasks from different perspectives.

Acceptance criteria -

['Users can filter tasks based on criteria such as due date, priority, assignee, or project.', 'Tasks can be sorted by various attributes.', 'The system provides multiple filter and sort options in the task list.', 'Users can save and apply custom filter and sort settings.', 'Filters and sorts do not affect the actual data, only the view.']

Generated Test Cases:

Test Case 1: Test Filtering by Due Date Functionality

- Test Case ID: TCT-001
- Test Case Title: Verify that users can filter tasks based on due date
- Test Case Description: Test the functionality of filtering tasks by their due date to ensure users can effectively organize and prioritize their tasks.
- Test Suite: Task Management - Task Filtering
- Test Priority: High
- Preconditions:
 - User is logged into the application
 - Multiple tasks with different due dates are created
- Test Data: Multiple tasks with various due dates
- Test Steps:
 1. Login to the application
 2. Go to the task list
 3. Activate the filter by due date function
 4. View the tasks that appear
- Postconditions: N/A
- Expected Result: Tasks should be filtered based on their due date
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 2: Test Saving and Applying Custom Filter and Sort Settings

- Test Case ID: TCT-002

- Test Case Title: Verify that users can save and apply custom filter and sort settings
- Test Case Description: Test the feature that allows users to save and apply their personalized filter and sort settings.
- Test Suite: Task Management - Task Filtering
- Test Priority: High
- Preconditions: User is logged in and has previously created a custom filter and sort setting.
- Test Data: Custom filter and sort settings
- Test Steps:
 1. Login to the application
 2. Navigate to task list
 3. Apply a previously created custom filter and sort setting
 4. Verify the tasks are displayed as per the selected filter and sort setting.
- Postconditions: N/A
- Expected Result: The tasks should be sorted and filtered according to the selected settings.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 3: Test Filter and Sort functionality Effect on Actual Data

- Test Case ID: TCT-003
- Test Case Title: Verify that the filters and sorts do not affect the actual data, only the view.
- Test Case Description: Test to ensure that the application of filters and sort settings only affects the view of the tasks and not the underlying data.
- Test Suite: Task Management - Task Filtering
- Test Priority: High

- Preconditions: User is logged into the application and has applied filters and sort settings.
- Test Data: No test data needed
- Test Steps:
 1. Login to the application
 2. Apply various filters and sort settings
 3. Verify that tasks view change according to the applied settings
 4. Remove all filters and sort settings
 5. Verify that the tasks view return to its original state without any permanent changes.
- Postconditions: N/A
- Expected Result: The application of filters and sort settings should not affect the underlying data.
- Severity: Blocker
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 4: Test Multiple Filter and Sort Options

- Test Case ID: TCT-004
- Test Case Title: Verify that the system provides multiple filter and sort options in the task list.
- Test Case Description: To validate that the system provides diverse filtering and sorting options to meet different users' needs.
- Test Suite: Task Management - Task Filtering
- Test Priority: High
- Preconditions: User has logged into the task management tool
- Test Data: No test data needed
- Test Steps:
 1. Login to the application

2. Navigate to the task list page

3. Check available filter and sort options

- Postconditions: N/A
- Expected Result: The application should display several filter and sort options to the user.
- Severity: Minor
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 5: Test Sorting by Priority Functionality

- Test Case ID: TCT-005
- Test Case Title: Verify that users can sort tasks based on priority.
- Test Case Description: Test the feature that allows users to sort their tasks based on priority to enhance task management and organization.
- Test Suite: Task Management - Task Filtering
- Test Priority: High
- Preconditions:
 - User is logged into the application
 - Multiple tasks with different priorities are created
- Test Data: Multiple tasks with various priorities
- Test Steps:
 1. Login to the application
 2. Go to the task list
 3. Activate the sort by priority function
 4. View the tasks that appear
- Postconditions: N/A

- Expected Result: Tasks should appear sorted based on their priority
 - Severity: Major
 - Priority: High
 - Type of Testing: Functional Testing
 - Test Case Approach: Positive
-

Data - 33

Project name -

Task Management Tool

Project description -

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

Feature name -

Time Tracking

Feature description -

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

Userstory title -

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Acceptance criteria -

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

Generated Test Cases:

Test Case 1:

- Test Case ID: 001
- Test Case Title: Verify that users can start and stop a timer for each task.
- Test Case Description: This test case verifies the proper functionality of the starting and stopping timer feature for each task in the task management tool.
- Test Suite: Time Tracking Module
- Test Priority: High
- Preconditions: 1) The user is logged in to the task management tool 2) A task is created
- Test Data: Task creation details
- Test Steps:
 - 1) Go to the task
 - 2) Start the timer
 - 3) Perform some operations or wait for a few minutes
 - 4) Stop the timer
- Postconditions: 1) Time tracking data for the task should be updated 2) Timer should be stopped
- Expected Result: The timer for the task starts and stops successfully and the time is recorded correctly.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 2:

- Test Case ID: 002
- 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 tasks.

- Test Suite: Time Tracking Module
- Test Priority: High
- Preconditions: 1) User is logged in to the task management tool 2) A task with recorded timer data exists
- Test Data: No test data needed
- Test Steps:
 - 1) Open the task
 - 2) Check the recorded time
- Postconditions: The time spent on the task should be displayed
- Expected Result: The total time spent on a task is accurately recorded and displayed.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 3:

- Test Case ID: 003
- Test Case Title: Verify that users can manually enter or adjust time entries.
- Test Case Description: This test case checks the functionality of manual entry or adjustment of times in time tracking.
- Test Suite: Time Tracking Module
- Test Priority: High
- Preconditions: 1) User is logged in 2) A task with recorded timer data exists
- Test Data: Time entries data
- Test Steps:
 - 1) Open the task
 - 2) Click on the time record

3) Enter or adjust the time manually

- Postconditions: The time record for the task should be updated
- Expected Result: Users can successfully manually enter or adjust time entries.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 4:

- Test Case ID: 004
- Test Case Title: Verify that time tracking data is stored and can be exported.
- Test Case Description: This case verifies if the system allows the data regarding time tracking to be stored and exported correctly
- Test Suite: Time Tracking Module
- Test Priority: Medium
- Preconditions: 1) User is logged in 2) Time tracking data exists for a task
- Test Data: No test data needed
- Test Steps:
 - 1) Open the task
 - 2) Click on the export data option
 - 3) Check the exported file for the correct time tracking data
- Postconditions: Time tracking data is copied into an exportable file properly
- Expected Result: The time tracking data is stored correctly and is exportable.
- Severity: Major
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 5:

- Test Case ID: 005
- Test Case Title: Verify that the system provides reports on time spent per task or project.
- Test Case Description: Test to verify if the system generates accurate reports specifying the time spent on individual tasks and projects.
- Test Suite: Time Tracking Module
- Test Priority: Medium
- Preconditions: 1) User is logged in 2) Time tracking data exists for tasks or projects
- Test Data: No test data needed
- Test Steps:
 - 1) Go to the reports section
 - 2) Generate a report for a specific task or project
 - 3) Check the reported time in the generated report
- Postconditions: A report with accurate task or project time data is generated
- Expected Result: The system accurately provides reports on time spent on individual tasks or projects.
- Severity: Major
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Data - 34

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Templates

Feature description -

Users can create and use templates for common task types or project workflows, streamlining task creation and standardizing processes across projects. Task templates can include predefined task structures, descriptions, assignees, and due dates, saving time and ensuring consistency in task execution.

Userstory title -

As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

Acceptance criteria -

['Users can create and save task templates.', 'Templates can include predefined task structures, descriptions, assignees, and due dates.', 'Users can apply templates to new tasks or projects.', 'Templates can be edited or deleted as needed.', 'The system displays a list of available templates for selection.']

Generated Test Cases:

Test Case ID: TMT001

Test Case Title: Verify that users can create and save task templates

Test Case Description: The test case validates the ability of users to create and save task templates as per the business requirement.

Test Suite: Task Template Management

Test Priority: High

Preconditions:

- The user is logged in.
- The user has the necessary rights to create task templates.

Test Data: No test data needed

Test Steps:

- Log in to the Task Management Tool.
- Navigate to the Task Template section.
- Click on the Create new template button.
- Provide the necessary details for the template.
- Click on the Save button.

Postconditions:

- The new template is listed under the Task Template section.
- The system triggers a confirmation message.

Expected Result: The task template is created and saved successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMT002

Test Case Title: Verify that templates can include predefined task structures, descriptions, assignees, and due dates

Test Case Description: This test case aims to validate the feature which allows templates to include predefined task structures, descriptions, assignees, and due dates.

Test Suite: Task Template Features

Test Priority: High

Preconditions:

- The user is logged in.
- The template creation/editing module is accessible.

Test Data: Predefined task structure, Description, User details, Due dates

Test Steps:

- Login to the application.
- Navigate to the Task Template section.
- Click on the Create new template/Edit existing template button.
- Provide predefined task structures, descriptions, assignees, and due dates.
- Save the template.

Postconditions:

- The predefined tasks structures, descriptions, assignees, and due dates are saved in the template.
- A confirmation message is triggered.

Expected Result: The template includes predefined task structures, descriptions, assignees, and due dates.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMT003

Test Case Title: Verify that users can apply templates to new tasks or projects

Test Case Description: This test case is designed to verify if users can apply chosen task templates to new task or project creations.

Test Suite: Task Template Application

Test Priority: High

Preconditions:

- The user is logged in.
- A task template exists.

Test Data: Task or project details

Test Steps:

- Login to the application.
- Navigate to the New Task/New Project section.

- Select the created task template from the drop-down menu.
- Create the new task or project.

Postconditions:

- The task or project is created with the template details.
- A confirmation message is displayed.

Expected Result: The user can successfully apply the template to new tasks or projects.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMT004

Test Case Title: Verify that templates can be edited or deleted as needed

Test Case Description: This test case checks that users can edit or delete task templates whenever necessary.

Test Suite: Template Management

Test Priority: High

Preconditions:

- The user is logged in.
- At least one task template exists.

Test Data: No test data needed

Test Steps:

- Login to the application.
- Navigate to the Task Templates section.
- Choose an existing template.
- Edit or delete the template.

Postconditions:

- Changes to the template are saved or the template is removed from the list.

- A confirmation message is displayed.

Expected Result: The selected task template is edited or deleted successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMT005

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 correctly displays a list of available task templates for user selection.

Test Suite: Template Display

Test Priority: High

Preconditions:

- The user is logged in.
- There are available templates.

Test Data: No test data needed

Test Steps:

- Login to the application.
- Navigate to the Create new task or project section.
- Click on the Template drop-down.

Postconditions:

- The system shows available templates.

Expected Result: The list of task templates is displayed for selection.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Data - 35

Project name -

Task Management Tool

Project description -

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

Feature name -

Calendar Integration

Feature description -

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

Userstory title -

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

Acceptance criteria -

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

Generated Test Cases:

Test Case 1

- Test Case ID: TC-TMT-01

- Test Case Title: Verify that tasks are able to sync with external calendars such as Google Calendar and Outlook.
- Test Case Description: This test case is designed to ensure that tasks created on the Task Management Tool sync seamlessly with external calendars.
- Test Suite: Calendar Integration
- Test Priority: High
- Preconditions: An account is created on the Task Management Tool and the external calendar is accessible.
- Test Data: Test Task 1, Google Calendar Account, Outlook Account
- Test Steps:
 1. Create a task on the Task Management Tool.
 2. Sync this task with both Google Calendar and Outlook.
 3. Check if the task appears on both calendars.
- Postconditions:
 1. The task remains on the Task Management Tool.
 2. Any changes to the task on the Tool should reflect on the synced calendars.
- Expected Result: The task created is visible on both Google Calendar and Outlook.
- Severity: Critical
- Priority: High
- Type of Testing: Integration Testing
- Test Case Approach: Positive

Test Case 2

- Test Case ID: TC-TMT-02
- Test Case Title: Verify that the system displays task deadlines in the calendar view.
- Test Case Description: This test case ensures that task deadlines set on the Task Management Tool reflect on the synced calendars.
- Test Suite: Calendar Integration

- Test Priority: High
- Preconditions: A task with a set deadline is created and synced with a calendar.
- Test Data: Test Task 2, Task Deadline
- Test Steps:
 1. Create a task with a set deadline.
 2. Sync this task with a calendar.
 3. Check if the task's deadline is displayed on the calendar view.
- Postconditions:
 1. Task with deadline remains on the Task Management Tool.
 2. Any changes to the deadline should reflect on the synced calendar.
- Expected Result: The task's deadline is displayed correctly on the calendar view.
- Severity: Critical
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 3

- Test Case ID: TC-TMT-03
- Test Case Title: Verify that the user can set reminders for tasks via the calendar.
- Test Case Description: This test case verifies that the user can set additional alerts or reminders for a task via the synced calendar.
- Test Suite: Calendar Integration
- Test Priority: Medium
- Preconditions: A task is created and synced with a calendar.
- Test Data: Test Task 3, Reminder Time
- Test Steps:
 1. Sync a task with a calendar.

2. On the calendar, set a reminder for the task.

3. Verify that the reminder is set and works as expected.

- Postconditions:

1. An alert for the reminder is received at the set time.

- Expected Result: The reminder for the task functions correctly.

- Severity: Major

- Priority: Medium

- Type of Testing: Functional Testing

- Test Case Approach: Positive

Test Case 4

- Test Case ID: TC-TMT-04

- Test Case Title: Verify that calendar events update automatically when tasks are changed.

- Test Case Description: This test case ensures that any modifications made to the task on the Task Management Tool are reflected in the calendar events.

- Test Suite: Calendar Integration

- Test Priority: High

- Preconditions: A task is created, with a deadline, and synced with a calendar.

- Test Data: Test Task 4, New Deadline

- Test Steps:

1. Sync a task with a calendar.

2. Change the deadline for the task on the Task Management Tool.

3. Verify if the new deadline is updated on the calendar event.

- Postconditions: Task with new deadline remains on the Task Management Tool.

- Expected Result: The new deadline is updated correctly on the calendar event.

- Severity: Critical

- Priority: High

- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 5

- Test Case ID: TC-TMT-05
- Test Case Title: Verify that users can manage calendar sync settings within the app.
- Test Case Description: This test case evaluates the functionality of managing calendar sync settings within the Task Management Tool.
- Test Suite: Calendar Integration
- Test Priority: Medium
- Preconditions: The user has an account and is logged into the Task Management Tool.
- Test Data: No test data needed
- Test Steps:
 1. Under settings, select the option to manage calendar sync settings.
 2. Make changes to the sync settings.
 3. Verify if the changes are saved and reflect accordingly.
- Postconditions: Updated sync settings saved in the Task Management Tool.
- Expected Result: The changes made to the sync settings are saved and function correctly.
- Severity: Major
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Data - 36

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Duplication

Feature description -

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

Userstory title -

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

Acceptance criteria -

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

Generated Test Cases:

Test Case 1:

Test Case ID: TD-001

Test Case Title: Verify that users can duplicate tasks or task templates.

Test Case Description: This test case verifies that users can effectively duplicate existing tasks or task templates.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in
- There is at least one task or task template available to be duplicated

Test Data: Name of the task or task template

Test Steps:

1. Select a task or template
2. Click on the duplicate option
3. Confirm duplication

Postconditions:

- Verify the duplicated task or template exists
- Confirm the task or template is identical to the original one

Expected Result: Successful duplication of task or template

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TD-002

Test Case Title: Verify that the duplicated task retains all details of the original task.

Test Case Description: This test case checks if all details including description, assignees, and deadlines are copied to the duplicated task.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in

- Task or template is duplicated

Test Data: Original Task or template details

Test Steps:

1. Compare the original and duplicated task or template
2. Check for similarity in description, assignees, and deadlines

Postconditions: None

Expected Result: All details of the original task or template are present in the duplicated one

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TD-003

Test Case Title: Verify that users can modify the duplicated task.

Test Case Description: This test checks if the user can edit and save changes to the duplicated task.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in
- Task or template has been duplicated

Test Data: Modifications to the task or template details

Test Steps:

1. Select the duplicated task or template
2. Make changes to the task or template details
3. Save the modifications

Postconditions: None

Expected Result: Modifications to the duplicated task or template are saved

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TD-004

Test Case Title: Verify that the system indicates that a task has been duplicated.

Test Case Description: This test checks if there's a system notification or indication that a task or template has been duplicated.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in
- Task or template has been duplicated

Test Data: No data needed

Test Steps:

1. Duplicate a task or template
2. Check for system message or indicator of duplication

Postconditions: None

Expected Result: System indicates that the task or template has been duplicated

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TD-005

Test Case Title: Verify that duplicated tasks are saved and displayed in the task list.

Test Case Description: This test makes sure that after duplicating a task, the created task is saved and visible in the task list.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in
- Task or template has been duplicated

Test Data: No data needed

Test Steps:

1. Duplicate a task or template
2. Save the duplicated task or template
3. Check the task list for the presence of the duplicated task

Postconditions: None

Expected Result: Duplicated tasks are saved and visible in the task list

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Project name -

Task Management Tool

Project description -

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

Feature name -

Task History

Feature description -

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

Userstory title -

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

Acceptance criteria -

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT_001

Test Case Title: Verify that users can view a detailed history of changes to tasks.

Test Case Description: This test case is designed to verify that users can view the complete history of changes made to a specific task.

Test Suite: Task History Test Suite

Test Priority: High

Preconditions:

- User is logged in the task management tool.
- User has created and modified a task.

Test Data: User credentials, Task ID

Test Steps:

1. Log in to the task management tool using user credentials.
2. Navigate to the specific task using the task ID.
3. Click on the 'History' tab.

Postconditions:

- No follow-up actions required.

Expected Result: The complete history of changes made to the task is displayed to the user.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT_002

Test Case Title: Verify that the history of changes to tasks includes modifications, comments, and other relevant activity.

Test Case Description: This test case intends to validate that the task history includes all the relevant activities like modifications, comments etc.

Test Suite: Task History Test Suite

Test Priority: High

Preconditions:

- User is logged in the task management tool.
- User has carried out modifications, added comments to a task.

Test Data: User credentials, Task ID

Test Steps:

1. Log in to the task management tool using user credentials.
2. Navigate to a specific task using the task ID.
3. Click on the 'History' tab.

Postconditions:

- No follow-up actions required.

Expected Result: All changes, including modifications, comments, and other related activities, are displayed in the history of the task.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT_003

Test Case Title: Verify that users can filter the task history by type of change or date.

Test Case Description: This test case is designed to verify that users can filter the task history by change type or date.

Test Suite: Task History Test Suite

Test Priority: High

Preconditions:

- User is logged in the task management tool.
- User is on the 'History' tab of a task

Test Data: User credentials, Task ID

Test Steps:

1. Log in to the task management tool using user credentials.
2. Navigate to a specific task using the task ID.
3. On 'History' tab, choose the filter option by type of change or date.

Postconditions:

- No follow-up actions required.

Expected Result: The history is updated to only display changes that match the selected filter.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT_004

Test Case Title: Verify that a timestamp is provided for each change in the task history.

Test Case Description: This test case is designed to ensure a timestamp is provided for each modification in the task history.

Test Suite: Task History Test Suite

Test Priority: High

Preconditions:

- User is logged in the task management tool.
- User is on the task details page, 'History' tab.

Test Data: User credentials, Task ID

Test Steps:

1. Log in to the task management tool using user credentials.

2. Navigate to a specific task using the task ID.

3. Navigate to the 'History' tab of the task.

Postconditions:

- No follow-up actions required.

Expected Result: For each modification or activity in the history, a timestamp is displayed.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT_005

Test Case Title: Verify that Task history is accessible from the task details page.

Test Case Description: This test case is designed to ensure that the Task history is accessible from the task details page.

Test Suite: Task History Test Suite

Test Priority: High

Preconditions:

- User is logged in the task management tool.
- User is on the task details page.

Test Data: User credentials, Task ID

Test Steps:

1. Log in to the task management tool using user credentials.
2. Navigate to a specific task using the task ID.
3. Look for the 'History' tab or link on the task details page.

Postconditions:

- No follow-up actions required.

Expected Result: The 'History' link or tab is present on the task details page, and by clicking on it, the task history is displayed.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 38

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

Userstory title -

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

Acceptance criteria -

['Users receive notifications for task updates, comments, mentions, and other relevant activities.',

'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.',

'The system logs notification history for future reference.']

Generated Test Cases:

Test Case 1

Test Case ID: TMT01

Test Case Title: Verify that the system sends notifications for task updates.

Test Case Description: This test case is intended to check if the notifications feature is functioning properly when tasks are updated.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

1. A user is logged into the system,
2. The user has ongoing tasks,
3. Tasks are updated by other users or creator.

Test Data: Task details

Test Steps:

1. Log into the system,
2. Open an existing task,
3. Update the task details,
4. Save changes.

Postconditions:

1. Check the notification panel,
2. Check email and mobile notifications.

Expected Result: Notifications regarding the task updates should be received through all enabled notification channels.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TMT02

Test Case Title: Verify that notification settings can be customized based on user preferences.

Test Case Description: This test case checks if the user can customize the notification settings as per his preferences.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

1. A user is logged into the system,
2. The user accesses the notification settings in the user profile.

Test Data: User preferences

Test Steps:

1. Log into the system,
2. Navigate to profile settings,
3. Update notification settings,
4. Save changes.

Postconditions: Check the delivery method of notifications based on updated preferences.

Expected Result: Notifications should be delivered based on the updated preferences.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TMT03

Test Case Title: Verify that the system logs notification history for future reference.

Test Case Description: This test case checks if the system logs all past notifications for future reference.

Test Suite: Task Notifications

Test Priority: Low

Preconditions: A user is logged into the system

Test Data: Past notifications

Test Steps:

1. Log into the system,
2. Navigate to notification settings,
3. Access the notification history tab.

Postconditions: Cross-verify any past notification with the notification history.

Expected Result: All past notifications should be logged in the notification history.

Severity: Minor

Priority: Low

Type of Testing: System Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TMT04

Test Case Title: Verify that the system sends detailed task notifications.

Test Case Description: This test case checks if the notifications provide detailed information about the task activity.

Test Suite: Task Notifications

Test Priority: High

Preconditions: A user is logged into the system, and an existing task is updated.

Test Data: Task details

Test Steps:

1. Log into the system,
2. Open any notification.

Postconditions: Cross-verify the information in the task with the notification details.

Expected Result: The notification should contain detailed information about the task activity.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TMT05

Test Case Title: Verify that the system does not send notifications if the user opts out of notifications.

Test Case Description: This test case checks if the system respects the user's preferences concerning notification settings.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions: A user is logged into the system, and the user opts out of notifications in the settings.

Test Data: User preferences

Test Steps:

1. Log into the system,
2. Opt-out of notifications in the settings,
3. Perform an action that would typically generate a notification.

Postconditions: Check all notification channels.

Expected Result: No notifications should be received if the user opts out of notifications.

Severity: Minor

Priority: Medium

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:

Test Case 1:

Test Case ID: TC1-5

Test Case Title: Verify that completed tasks can be archived

Test Case Description: This test case aims to verify if the tool allows archiving of completed tasks

Test Suite: Archiving Functionality

Test Priority: High

Preconditions: User has completed a task

Test Data: Completed tasks

Test Steps: 1. Log into the tool. 2. Go to the completed tasks list. 3. Select a task. 4. Click on 'Archive Task'

Postconditions: Redirection to the 'Archived tasks' section

Expected Result: Selected task is moved to the 'Archived tasks' section

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC2-5

Test Case Title: Verify that the archived tasks are moved to a separate section for easy access

Test Case Description: Testing the functionality if the archived tasks are categorized into a separate section for easy access

Test Suite: Archiving Functionality

Test Priority: High

Preconditions: Tasks have been archived

Test Data: No test data needed

Test Steps: 1. Login to the tool. 2. Click on the 'Archived Tasks' section. 3. Check if the previously archived tasks are present.

Postconditions: Update of the 'View Archived Tasks' list

Expected Result: Archived tasks are visible in the 'Archived Tasks' section

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC3-5

Test Case Title: Verify that users can search archived tasks

Test Case Description: This test case verifies if the user can search for a specific task in the 'Archived Tasks' section

Test Suite: Archiving Functionality

Test Priority: High

Preconditions: Tasks have been archived

Test Data: A specific archived task name

Test Steps: 1. Login to the tool. 2. Navigate to 'Archived Tasks' section. 3. Use the search bar to search for a

particular task.

Postconditions: Display of the search results

Expected Result: Selected task is visible in the search results

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC4-5

Test Case Title: Verify the option to restore archived tasks

Test Case Description: This test verifies whether the tool allows the restoration of archived tasks back to the active list

Test Suite: Archiving Functionality

Test Priority: High

Preconditions: Tasks have been archived

Test Data: Archived task

Test Steps: 1. Login to the tool. 2. Navigate to 'Archived Tasks' section. 3. Select a task. 4. Click on 'Restore Task'

Postconditions: Task is moved back to the 'Active Tasks' list

Expected Result: Task is visible in the 'Active Tasks' list

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC5-5

Test Case Title: Verify that archived tasks retain all original details and history

Test Case Description: This test checks whether all task details and history are retained during archiving

Test Suite: Archiving Functionality

Test Priority: High

Preconditions: Tasks have been archived

Test Data: Archived task

Test Steps: 1. Login to the tool. 2. Navigate to 'Archived Tasks' section. 3. Open a task. 4. View task details and history

Postconditions: Display of detailed task view

Expected Result: All task details and history remain intact after archiving

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 40

Project name -

Task Management Tool

Project description -

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

Feature name -

Guest Access

Feature description -

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

Userstory title -

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

Acceptance criteria -

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_01

Test Case Title: Verify that users can invite external collaborators as guests with limited access.

Test Case Description: This test case is designed to ensure users are able to invite external collaborators as guests and assign them with limited access.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in.
- User has a valid external collaborator to invite.

Test Data: Collaborator's email address.

Test Steps:

1. Login to the system.
2. Navigate to guest access feature in the dashboard.
3. Enter the collaborator's email address and send an invite.
4. Set delegations and save changes.

Postconditions:

- An invitation is sent to the collaborator.

Expected Result: Collaborator receives an email invitation to join as a guest with specified limited access.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_02

Test Case Title: Verify that guests can view and collaborate on specific tasks or projects.

Test Case Description: This test case will check if guests can view and collaborate in the tasks assigned to them.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- Guest has accepted the invitation and logged into the system.

Test Data: Existing tasks or projects assigned to the guest.

Test Steps:

1. Guest logs in to the system.

2. Navigate to assigned tasks or project in the dashboard.
3. View the task details and add inputs or collaborations if required.

Postconditions:

- Guest participation is reflected in the task progress.

Expected Result: Guests can view tasks assigned to them and provide necessary input or collaboration.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case 3:

Test Case ID: TC_03

Test Case Title: Verify that the system provides clear distinctions between guest and full access roles.

Test Case Description: This test verifies that the system clearly distinguishes between roles with Full access and Guest access.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User or guest is logged in.

Test Data: No test data needed.

Test Steps:

1. Login to the system.
2. Navigate to account settings or profile.
3. Check the assigned role.

Postconditions:

- No After-action needed.

Expected Result: The system states whether the logged in account is a 'Guest' or 'Full User'.

Severity: Minor

Priority: Medium

Type of Testing: User Interface Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_04

Test Case Title: Verify that users can manage and revoke guest access at any time.

Test Case Description: The test case ensures that users can manage or revoke guest access whenever required.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in.
- There is an existing guest on the system.

Test Data: Guest's email or user name.

Test Steps:

1. Login to the system.
2. Navigate to the guest access management.
3. Select the guest user and revoke access.

Postconditions:

- Guest access is successfully revoked.

Expected Result: The selected guest should no longer have access to the system.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_05

Test Case Title: Verify that guests cannot access sensitive or restricted areas of the platform.

Test Case Description: This case will check if the guests are blocked from accessing sensitive or restricted parts of the platform.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- Guest is logged in.

Test Data: No test data needed.

Test Steps:

1. Login as a guest.
2. Try to access restricted or sensitive areas of the platform.

Postconditions:

- Revert any changes attempted by the guest.

Expected Result: The guest should receive a message or experience a block preventing them from accessing any sensitive or restricted areas.

Severity: Critical

Priority: High

Type of Testing: Security 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 Import/Export

Feature description -

Users can import tasks from external sources or export tasks to other task management tools or file formats. Import/export capabilities facilitate data migration and integration with other systems, ensuring seamless collaboration and workflow management across platforms.

Userstory title -

As a user, I want to import and export tasks so that I can migrate data between different task management tools.

Acceptance criteria -

['Users can import tasks from external sources in various formats (e.g., CSV, Excel).', 'Users can export tasks to other task management tools or file formats.', 'The system provides clear mapping of fields during import/export.', 'Users can preview imported data before finalizing the process.', 'The system logs import/export activities for future reference.']

Generated Test Cases:

Test Case ID: TMT001

Test Case Title: Verify that the Task Management Tool can import tasks from CSV files.

Test Case Description: The test is designed to validate the system's ability to import tasks from external sources in CSV format.

Test Suite: Task Import/Export

Test Priority: High

Preconditions: User has an operational Task Management Tool account with permissions to import tasks, and a valid CSV file with task data is available.

Test Data: CSV data file with properly formatted task data (title, description, priority, deadline, attachments).

Test Steps:

1. Log in to the Task Management Tool.
2. Navigate to the import tasks feature.
3. Select CSV format.
4. Upload the test CSV file.
5. Preview and finalize the import process.

Postconditions: CSV file is cleared after import, system logs the import activity.

Expected Result: All tasks from CSV file are imported successfully to the user's task list in the Task Management Tool.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TMT002

Test Case Title: Verify that the Task Management Tool can export tasks to CSV files.

Test Case Description: The test verifies the system's ability to export tasks to CSV format for data migration.

Test Suite: Task Import/Export

Test Priority: High

Preconditions: User has an operational Task Management Tool account, and there are tasks available to export.

Test Data: No test data needed.

Test Steps:

1. Log in to the Task Management Tool.
2. Select the tasks to export.
3. Navigate to the export tasks feature.
4. Select the CSV format.
5. Confirm the export process.

Postconditions: System logs the export activity.

Expected Result: All the selected tasks are exported successfully into a CSV file.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TMT003

Test Case Title: Verify the clear field mapping during import/export tasks process.

Test Case Description: The test is to ensure clear field mapping during task import/export on Task Management Tool.

Test Suite: Task Import/Export

Test Priority: High

Preconditions: Import/Export operation is initiated.

Test Data: CSV data file for importing tasks.

Test Steps:

1. Start the import/export process.
2. Observe the field mapping.
3. Verify the clarity and accuracy of the field mapping.

Postconditions: System logs the import/export activity.

Expected Result: The field mapping process is clear and accurate during the import/export process.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TMT004

Test Case Title: Verify that users can preview the imported data before finalizing the process.

Test Case Description: This test checks if the application allows users to preview imported tasks before concluding the process.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions: User has uploaded a file to be imported.

Test Data: CSV data file for importing tasks.

Test Steps:

1. Complete the process up to the point of finalizing the import.
2. Preview the tasks to be imported.
3. Check if the preview matches the actual data to be imported.

Postconditions: Import process is complete, system logs the import activity.

Expected Result: The user can correctly view and validate the preview of imported data before finalizing.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TMT005

Test Case Title: Verify that import/export activities are logged for future reference.

Test Case Description: The aim of this test is to confirm if the system accurately logs import/export operations.

Test Suite: Task Import/Export

Test Priority: Low

Preconditions: Import/export operation is performed.

Test Data: No test data needed.

Test Steps:

1. Carry out an import/export operation.
2. Go to the section where the system logs activities.
3. Look for the log of the performed import/export operation.

Postconditions: None.

Expected Result: The system logs import/export activities accurately for future reference.

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:

Test Case 1:

Test Case ID: TC01

Test Case Title: Verify that users can set and adjust task priorities.

Test Case Description: This test case will verify that users have the ability to set and change task priorities as needed.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User must be logged in
- At least one task should exist

Test Data: User credentials, Task details

Test Steps:

1. Login to the application with valid user credentials.
2. Navigate to the 'Tasks' page.
3. Select a task and set a priority level.

4. Verify that the priority level is displayed next to the task.
5. Change the priority level of the task.
6. Verify the updated priority is correctly displayed.

Postconditions:

- User logs out of the application
- Priority changes are saved

Expected Result: The user can set and adjust the task priorities.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC02

Test Case Title: Verify that tasks can be sorted and filtered based on priority levels.

Test Case Description: This test case verifies if the application provides functionality to sort and filter tasks based on their priority.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User must be logged in
- At least one task should exist

Test Data: User credentials, Task details, Priority details

Test Steps:

1. Login to the application with valid user credentials.
2. Navigate to the 'Tasks' page.

3. Sort the tasks by priority level.
4. Now, filter the tasks by a specific priority level.
5. Verify the tasks are correctly sorted and filtered.

Postconditions:

- User logs out of the application
- Task sorting and filter changes saved

Expected Result: The tasks can be sorted and filtered based on priority levels.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC03

Test Case Title: Verify that visual indicators of task priority are provided.

Test Case Description: This test case checks whether the system provides visual indicators of task priority.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User must be logged in
- At least one task should exist

Test Data: User credentials, Task details, Priority details

Test Steps:

1. Login to the application with valid user credentials.
2. Navigate to the 'Tasks' page.
3. Check for visual indicators next to each task reflecting their priority.

4. Change the priority of a task.
5. Verify that the visual indicator updates accordingly.

Postconditions:

- User logs out of the application
- Visual indicators of task priority saved

Expected Result: The application provides clear and correct visual indicators of task priority.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC04

Test Case Title: Verify that users receive notifications for high-priority tasks.

Test Case Description: This test case checks whether the system sends notifications for high-priority tasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User must be logged in
- Notifications should be enabled
- At least one high priority task exists

Test Data: User credentials, Task details, Priority details

Test Steps:

1. Login to the application with valid user credentials.
2. Navigate to the 'Tasks' page.
3. Set a task to high priority and set a deadline.

4. Wait for the notification.

5. Verify a notification is received for the high-priority task.

Postconditions:

- User logs out of the application
- Notifications received by user saved

Expected Result: The application sends notifications for high-priority tasks.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC05

Test Case Title: Verify that priority changes are logged and tracked.

Test Case Description: This test case verifies that changes in task priority are logged and accessible to users.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User must be logged in
- Task with changes in priority exists

Test Data: User credentials, Task details, Priority details

Test Steps:

1. Login to the application with valid user credentials.
2. Navigate to the 'Tasks' page.
3. Select a task that has had its priority changed.
4. Check the task log for priority changes.

5. Verify that the log reflects all changes made to the task's priority.

Postconditions:

- User logs out of the application
- Priority change logs saved

Expected Result: Priority changes are correctly logged and tracked in the system.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 43

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

Userstory title -

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay

updated on task progress.

Acceptance criteria -

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

Generated Test Cases:

Test Case ID: TC001

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

Test Case Description: This test case will validate whether team members can be assigned tasks in the application.

Test Suite: Task Management and Collaboration

Test Priority: High

Preconditions: User is logged in, the team member to be assigned is registered in the project, the task has been created.

Test Data: Task details, team member user ID/name

Test Steps: 1. Navigate to the specific task. 2. Click on 'Assign'. 3. Search and select the required team member 4. Click on 'Confirm'.

Postconditions: Confirmation message is displayed, the task assignment is visible in task details, team member receives a notification about the assignment.

Expected Result: The task is assigned to the selected team member and is visible in their task list.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC002

Test Case Title: Verify that task updates and comments are visible to assigned team members.

Test Case Description: This test case will validate whether changes made to a task or comments added are visible to all assigned team members.

Test Suite: Task Management and Collaboration

Test Priority: High

Preconditions: User is logged into the application, a task has been assigned to at least one team member.

Test Data: Task details, team member user IDs names, comments details.

Test Steps: 1. Log in as a user. 2. Navigate to the task. 3. Make updates or add comments to the task. 4. Log out and login as an assigned team member. 5. Navigate to the same task.

Postconditions: The updates / comments made are visible from the other assigned team members' account.

Expected Result: All task updates and comments should be visible to assigned team members.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC003

Test Case Title: Verify that users receive notifications for task updates and comments.

Test Case Description: This test case will check if users are getting notifications whenever a task update or comment is made.

Test Suite: Task Management and Collaboration

Test Priority: Medium

Preconditions: User is logged into the application, a task has been assigned to the user, email (for notifications) is set up correctly in the user's profile.

Test Data: User login details, task details, comments data.

Test Steps: 1. Log in as another user. 2. Navigate to the specific task and make an update or add a comment. 3. Log out and log in as the assigned team member.

Postconditions: The user receives a notification (e.g. on their homepage/dashboard or by email)

Expected Result: The user should receive notifications for task updates and comments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TC004

Test Case Title: Verify that the system provides a clear view of task collaboration history.

Test Case Description: This test case will validate if the system shows a clear history of the task collaboration.

Test Suite: Task Management and Collaboration

Test Priority: Medium

Preconditions: User is logged into the application, tasks have been created and updated.

Test Data: User login details, task details, collaboration history data.

Test Steps: 1. Log into the system. 2. Navigate to a task. 3. Click on 'Task Collaboration History'.

Postconditions: The task collaboration history is displayed.

Expected Result: The task collaboration history should clearly list all previous collaboration actions, such as task updates, comments, task assignments.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC005

Test Case Title: Verify that users are unable to assign tasks to non-team members.

Test Case Description: This test case will check if users can assign tasks to individuals who are not part of their team within the application.

Test Suite: Task Management and Collaboration

Test Priority: Medium

Preconditions: User is logged into the application, the task has been created, the individual to be assigned is not registered in the project.

Test Data: Task details, non-team member user ID/name

Test Steps: 1. Navigate to the specific task. 2. Click on 'Assign'. 3. Search and try to select the non-team member.

Postconditions: Alert message is displayed.

Expected Result: The system does not allow assigning tasks to non-team members and displays an appropriate error message.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Negative

Data - 44

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tracking

Feature description -

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

Userstory title -

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

Acceptance criteria -

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.', 'System provides visual indicators for task progress.', '']

Generated Test Cases:

Test Case 1:

Test Case ID: TT-01

Test Case Title: Verify that users can properly view task statuses

Test Case Description: This test case verifies the ability of users to view task statuses on the Task Management Tool.

Test Suite: Task Tracking Module

Test Priority: High

Preconditions: User is logged in and has at least one task created.

Test Data: Task data including task status.

Test Steps:

1. Log into the application
2. Navigate to the "My Tasks" menu
3. Click on an existing task

Postconditions:

1. User returns to dashboard

Expected Result: The user can view the task status accurately.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TT-02

Test Case Title: Verify that users can track task progress and deadlines

Test Case Description: This test case is for testing whether users can track their task progress and deadlines.

Test Suite: Task Tracking Module

Test Priority: High

Preconditions: User is logged in and has tasks assigned with a set deadline.

Test Data: Task details including deadlines

Test Steps:

1. Log into the application
2. Navigate to the "My Tasks" menu
3. Select a task
4. Check due date and task completion status

Postconditions:

1. User returns to dashboard

Expected Result: User can accurately view the task deadline and progress.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TT-03

Test Case Title: Verify that tasks are updated in real-time

Test Case Description: This test case ensures that any changes to tasks are displayed in real-time in the Task Management Tool.

Test Suite: Task Tracking Module

Test Priority: High

Preconditions: User is logged in and has a task to edit.

Test Data: Updated task information

Test Steps:

1. Log into the application
2. Navigate to the "My Tasks" menu
3. Select a task and edit details
4. Save changes and verify if changes reflected immediately

Postconditions:

1. User returns to dashboard

Expected Result: All task edits are immediately reflected on the application.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TT-04

Test Case Title: Verify that the system provides visual indicators for task progress

Test Case Description: This test case is to verify if the system provides visual indicators showing the progress

of the tasks.

Test Suite: Task Tracking Module

Test Priority: High

Preconditions: User is logged in and has tasks to track.

Test Data: No test data needed

Test Steps:

1. Log into the application
2. Navigate to the "My Tasks" menu
3. Check for visual indicators associated with task progress

Postconditions:

1. User returns to dashboard

Expected Result: System should display visual indicators to represent the progress of tasks.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TT-05

Test Case Title: Verify that viewing task status fails with improper credentials

Test Case Description: This test case is to confirm if the application restricts viewing of task status for users logged in with incorrect credentials.

Test Suite: Task Tracking Module

Test Priority: High

Preconditions: User is attempting to log in

Test Data: Incorrect username and/or password

Test Steps:

1. Deliberately use incorrect credentials to log in
2. Attempt to navigate to the "My Tasks" menu

Postconditions:

1. User remains on login screen

Expected Result: System should deny access to task status without proper login credentials.

Severity: Critical

Priority: High

Type of Testing: Security Testing

Test Case Approach: Negative

Data - 45

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

Userstory title -

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my

tasks effectively.

Acceptance criteria -

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.', 'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', '']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT-1

Test Case Title: Verify that users receive notifications for new tasks

Test Case Description: This test case checks whether users receive notifications when a new task is assigned to them.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool
- A new task is assigned to the user

Test Data: No test data needed

Test Steps:

1. Log into the Task Management Tool
2. Assign a new task to a user
3. Check if a notification is received by the user

Postconditions:

- User has received notification
- User opens and checks new task

Expected Result: The user should receive a notification for the new task

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-2

Test Case Title: Verify that users receive notifications for task updates and comments

Test Case Description: This test case checks whether users receive notifications when there is any update or comment on their tasks.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool
- Task exists for user
- Comments or updates are made on the task

Test Data: No test data needed

Test Steps:

1. Log into the Task Management Tool
2. Make an update or comment on a user's task
3. Check if a notification is received by the user

Postconditions:

- User has received notification
- User checks updates or comments

Expected Result: The user should receive a notification for the updates or comments on task

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-3

Test Case Title: Verify that users receive reminders for upcoming deadlines

Test Case Description: This test case checks whether users receive reminder notifications as deadlines for their tasks approach.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool
- User has a task with a set deadline

Test Data: No test data needed

Test Steps:

1. Log into the Task Management Tool
2. Check if a reminder notification is received by the user as the task deadline approaches

Postconditions:

- User has received reminder notification
- User checks task details

Expected Result: The user should receive a reminder notification for upcoming task deadlines

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-4

Test Case Title: Verify that notification settings can be customized by users

Test Case Description: This test case checks whether users are able to customize their notification settings as per their preferences.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged into the Task Management Tool

Test Data: No test data needed

Test Steps:

1. Log into the Task Management Tool
2. Navigate to the settings
3. Customize the notification settings
4. Check if the changes are saved and applied

Postconditions:

- User has customized notification settings

Expected Result: The user should be able to customize their notification settings

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-5

Test Case Title: Verify that users do not receive notifications for tasks they are not involved in

Test Case Description: This test case checks that users do not receive irrelevant notifications for tasks they are not assigned to or involved in.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- More than one user is logged into the Task Management Tool
- A task is assigned to one user, not involving the other

Test Data: No test data needed

Test Steps:

1. Log into the Task Management Tool with two different users
2. Assign a task to one user
3. On the second user's profile, check if a notification is received

Postconditions:

- Second user does not receive notification about the other's task

Expected Result: The user should not receive a notification for tasks they are not involved in

Severity: Minor

Priority: Medium

Type of Testing: Negative Testing

Test Case Approach: Negative

Data - 46

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Categorization

Feature description -

Users can categorize tasks into different groups or projects for better organization and management. Task categorization aims to help users organize tasks by type, priority, or project.

Userstory title -

As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

Acceptance criteria -

['Users can create and assign tasks to different categories or projects.', 'Users can view tasks grouped by categories or projects.', 'Tasks can be moved between categories or projects.', 'Category or project information is accurately reflected in task views.', "']

Generated Test Cases:

Test Case 1:

- Test Case ID: 001
- Test Case Title: Verify that the user can create and assign tasks to different categories or projects.
- Test Case Description: This test case is designed to validate the functionality which lets users assign tasks to various categories.
- Test Suite: Task Categorization
- Test Priority: High
- Preconditions: User should have an authorized account, User interface(UI) should be accessible, User is logged into their account

- Test Data: User account details, Task details, Category details

- Test Steps:

1. Login to the Task Management Tool.

2. Navigate to 'Create Task' option.

3. Provide necessary task information.

4. Assign the task to a specific category.

5. Save the task.

- Postconditions: Task is created and assigned to a category, User is notified of successful task assignment,

Task appears in the appropriate category/task view

- Expected Result: Task is successfully assigned to the chosen category.

- Severity: Major

- Priority: High

- Type of Testing: Functional Testing

- Test Case Approach: Positive

Test Case 2:

- Test Case ID: 002

- Test Case Title: Verify that users can view tasks grouped by categories or projects.

- Test Case Description: This test case is designed to verify the viewability of tasks grouped by categories or projects.

- Test Suite: Task Categorization

- Test Priority: High

- Preconditions: User should have an authorized account, UI should be accessible, User is logged into their account, Tasks are created and assigned to various categories

- Test Data: User account details

- Test Steps:

1. Login to the Task Management Tool.
 2. Navigate to 'Task View' option.
 3. Select the 'View by Category' option.
 4. Observe how tasks are grouped.
- Postconditions: Tasks are presented in a categorized manner.
 - Expected Result: Tasks are accurately grouped by categories or projects.
 - Severity: Major
 - Priority: High
 - Type of Testing: Functional Testing
 - Test Case Approach: Positive

Please note: Remaining Test cases would follow the same template ensuring all acceptance criteria are covered.

Data - 47

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Search

Feature description -

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due

date. The task search feature aims to help users quickly find specific tasks within their project or task list.

Userstory title -

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

Generated Test Cases:

Test Case 1

Test Case ID: TC001

Test Case Title: Verify that users can search tasks by name.

Test Case Description: In this test, we will check the search functionality with task names as the criteria.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in.
- There are tasks created previously without same names.

Test Data: Valid task name.

Test Steps:

1. User navigates to the search bar.
2. User enters a valid task name in the search field.
3. User submits the search.

Postconditions:

- User is presented with search results.

Expected Result: Search results should display the tasks with the same name.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC002

Test Case Title: Verify that users can search tasks by description.

Test Case Description: This test case is to verify that the user will get the correct task when searching with a task description in the search bar.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- User is logged in.
- The task having the desired description exists.

Test Data: Valid task description.

Test Steps:

1. User navigates to the search bar.
2. User enters a valid task description in the search field.
3. User submits the search.

Postconditions:

- User is presented with search results.

Expected Result: The search results should display the task with the entered description.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC003

Test Case Title: Verify that users can filter search results by category, priority, or due date.

Test Case Description: This test case will validate the filtering functionality of the task search results based on category, priority, or due date.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in.
- Results from a search query are available.

Test Data: Task category, Task priority, or Task due date.

Test Steps:

1. User performs a task search.
2. User applies the filter based on category, priority, or due date in the search results.
3. User observes the search results.

Postconditions:

- User can see the filtered search results.

Expected Result: The filtered search results should only include tasks matching the applied filter criteria.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC004

Test Case Title: Verify that the search results are accurate and relevant.

Test Case Description: This test case will confirm that the search results are accurate and match the search criteria from the user.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in.
- Tasks related to the search criteria exist.

Test Data: Search query

Test Steps:

1. User performs a task search.
2. User examines the search results.

Postconditions:

- User can see the search results.

Expected Result: All tasks in the search results should be accurate and relevant to the search query.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC005

Test Case Title: Verify that no results are returned when non-existing task name is used in the search.

Test Case Description: This test case will validate that the search functionality returns no results when a

non-existing task name is searched.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- User is logged in.
- The searched task name does not exist.

Test Data: Task name not present in the system.

Test Steps:

1. User navigates to the search bar.
2. User enters a non-existing task name in the search field.
3. User submits the search.

Postconditions:

- User can see the search results.

Expected Result: The search results should display a message indicating no matching tasks were found.

Severity: Minor

Priority: Medium

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:

Test Case ID: TC001

Test Case Title: Verify that users can add comments to tasks

Test Case Description: This test case verifies that a user can successfully add comments to an existing task in the 'Task Collaboration' feature.

Test Suite: Task Collaboration Tests

Test Priority: High

Preconditions:

- User is logged into the task system
- A task exists for the user to collaborate on

Test Data: Comment text "Excellent progress!"

Test Steps:

1. User logs into the system.
2. User navigates to the "Task Collaboration" feature and selects an existing task.

3. User clicks on the ?Add Comment? button and types in a comment.

4. User clicks on the ?Submit? button to add the comment to the task.

Postconditions:

- Comment is displayed in the task's comment section

Expected Result: The added comment appears correctly under the task's comment section.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC002

Test Case Title: Verify that users can attach files to tasks

Test Case Description: This test case checks the functionality of attaching files to existing tasks by users.

Test Suite: Task Collaboration Tests

Test Priority: High

Preconditions:

- User is signed into the task management system
- An existing task is available for manipulation

Test Data: PDF file for upload

Test Steps:

1. User logs into the system.
2. User navigates to the "Task Collaboration" feature and selects a task.
3. User clicks on the ?Attach File? button.
4. User selects a file from their system and clicks on ?Upload.?

Postconditions:

- File is attached to the selected task

Expected Result: The user-selected file is successfully attached to the task.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

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.

Test Suite: Task Collaboration Tests

Test Priority: High

Preconditions:

- User is signed into the task management tool
- Another user exists to share the task with
- An existing task is in the system

Test Data: Username of the user with whom to share the task

Test Steps:

1. User logs in and chooses a task in the "Task Collaboration" feature.
2. User clicks on the ?Share? button.
3. User inputs the username of another user and clicks on ?Share.?

Postconditions:

- Task is shared with the designated user

Expected Result: The task is successfully shared with the specified user.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC004

Test Case Title: Verify that task collaboration features work seamlessly

Test Case Description: This test case checks that features within 'Task Collaboration' interact seamlessly and function as intended.

Test Suite: Task Collaboration Tests

Test Priority: High

Preconditions:

- At least 2 tasks exist in the system
- At least 2 system users exist

Test Data: No test data needed

Test Steps:

1. User A logs in and adds a comment to task 1.
2. User A shares task 1 with User B.
3. User B logs in, views task 1, and adds another comment.
4. User A responds in the comment thread.
5. User B attaches a file to task 1.

Postconditions:

- Task is shared and comments and files are visible to both users.

Expected Result: All the collaborative actions performed (adding comments, sharing tasks, and attaching files) are visible and working for both users.

Severity: Major

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: TC005

Test Case Title: Verify that task collaboration enhances communication

Test Case Description: This test case confirms whether the 'Task Collaboration' feature enhances communication between team members.

Test Suite: Task Collaboration Tests

Test Priority: High

Preconditions:

- At least 2 tasks exist in the system
- At least 2 system users exist

Test Data: No test data needed

Test Steps:

1. User A shares a task with User B.
2. User B adds a comment on a shared task.
3. User A views the comment and responds on the shared task.
4. Monitor time taken between steps 1, 2, and 3.

Postconditions:

- Interaction has occurred between both users using the 'Task Collaboration' system

Expected Result: The communication process takes place faster and more efficiently within the task management system than via external communication methods (e.g. email).

Severity: Major

Priority: High

Type of Testing: Usability Testing, Performance Testing

Test Case Approach: Positive

Data - 49

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Course Creation Tools

Feature description -

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to structure content effectively, ensuring an engaging and informative learning experience for students.

Userstory title -

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

Acceptance criteria -

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC001

Test Case Title: Verify that instructors can create courses using videos, presentations, quizzes, and assignments.

Test Case Description: This test case is designed to check if instructors are able to use the various tools provided for course creation.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- Instructor account is created
- Instructor is logged in

Test Data: Sample video, presentations, quizzes, and assignments files

Test Steps:

1. Log in with the instructor account
2. Navigate to the course creation section
3. Use the upload function to add videos, presentations
4. Create quizzes and assignments using the respective tools

Postconditions:

- Course gets saved in draft

Expected Result: New course data is successfully saved as a draft.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that the course creation tools are user-friendly and require minimal technical skills.

Test Case Description: This test case checks if the course creation tools are easy to use without needing high technical knowledge.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor is in the course creation section

Test Data: No test data needed

Test Steps:

1. Navigate to each tool used for course creation
2. Check each tool's simplicity of use and straightforwardness

Postconditions:

- Instructor should be able to simply navigate

Expected Result: The tools are intuitive and easy to use.

Severity: Critical

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003

Test Case Title: Verify that instructors can preview courses before publishing.

Test Case Description: This test case checks the feasibility of course preview before it is published.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- Instructor is logged in
- A course has been created

Test Data: No test data needed.

Test Steps:

1. Click on the course created
2. Click the 'Preview' button
3. Observe the course as it would appear post-publication

Postconditions: -

Expected Result: The course is displayed in preview mode, mimicking its appearance after being published.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004

Test Case Title: Verify that instructors can save courses as drafts.

Test Case Description: This test case tests the capability of saving the progress of course creation.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor is at the course creation interface

Test Data: Various course material for a new course.

Test Steps:

1. Start creating a course
2. Click 'Save as draft'

Postconditions:

- Course should be saved in the drafts folder

Expected Result: The course in creation is saved as draft successfully, without being published.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005

Test Case Title: Verify that instructors can edit courses post-publishing.

Test Case Description: To test the functionality that allows instructors to modify the course even after it is published.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

- Instructor is logged in
- A course has been published

Test Data: Updated course content

Test Steps:

1. Navigate to the published courses
2. Select the respective course
3. Make desired changes and save

Postconditions:

- Course is updated with the modifications

Expected Result: The published course is successfully updated and the changes are visible to the students immediately.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 50

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Interactive Lessons

Feature description -

Lessons are thoughtfully designed to foster learner engagement through interactive components such as quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these interactive elements enhance comprehension and retention of course material.

Userstory title -

As a learner, I want interactive lessons so that I can engage with the content more effectively.

Acceptance criteria -

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided after each interactive component.', 'Interactive lessons can be accessed on various devices.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC1_Interactive Elements

Test Case Title: Verify that lessons include quizzes, polls, simulations, and discussions

Test Case Description: This test case verifies the existence and functionality of interactive elements in the online lessons (quizzes, polls, simulations, discussions)

Test Suite: Interactive Lesson

Test Priority: High

Preconditions:

- Learner is logged into the platform
- Lesson content has been loaded

Test Data: Lesson with interactive elements

Test Steps:

1. Start any lesson with interactive elements
2. Participate in a quiz
3. Partake in a poll
4. Engage with a simulation
5. Start a discussion

Postconditions:

- User feedback is recorded
- Participation history is updated

Expected Result: All four interactive elements should function properly

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC2_Device Compatibility

Test Case Title: Verify that interactive lessons can be accessed on various devices

Test Case Description: This test evaluates the compatibility of interactive lessons across different devices

Test Suite: Interactive Lesson

Test Priority: Medium

Preconditions:

- User account is functioning properly
- Devices are connected to a stable network

Test Data: No test data needed

Test Steps:

1. Access the platform on various devices (mobile phone, tablet, and desktop PC)
2. Start an interactive lesson on each device

Postconditions:

- Each lesson should fit the device's screen resolution
- User progress is synchronized across devices

Expected Result: Lessons should load and function properly on all selected devices

Severity: Major

Priority: High

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC3_Tracking Progress

Test Case Title: Verify that learners can track their progress through activities

Test Case Description: This test ensures that the platform accurately tracks user progress in various interactive elements

Test Suite: Interactive Lesson

Test Priority: High

Preconditions:

- User is logged in to the platform
- At least one lesson has been completed

Test Data: User activity history

Test Steps:

1. Log in to your account
2. Navigate to the progress tracking feature
3. Verify if progress indicators for quizzes, polls, simulations, and discussions are accurate

Postconditions:

- User progress status is updated in synchronization with completed tasks

Expected Result: Progress tracking should reflect accurate and current status

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC4_Interactive Feedback

Test Case Title: Verify that feedback is provided after each interactive component

Test Case Description: This test checks whether the system provides immediate and accurate feedback after engaging with interactive components.

Test Suite: Interactive Lesson

Test Priority: High

Preconditions:

- User is on an active lesson
- Internet connection is stable

Test Data: User response to a quiz, poll, simulation, or discussion

Test Steps:

1. Enter any lesson module
2. Engage with any interactive component
3. Submit response

Postconditions:

- Feedback for the interactive component should display

Expected Result: User receives immediate feedback after engagement

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC5_Integration Elements

Test Case Title: Verify that interactive elements are integrated seamlessly into the lesson content

Test Case Description: This test ensures that interactive elements are smoothly incorporated within the lesson context without causing any disruption

Test Suite: Interactive Lesson

Test Priority: High

Preconditions:

- User has selected a lesson

- Lesson has loaded successfully on the platform

Test Data: No test data needed

Test Steps:

1. Start the lesson
2. Engage with an interactive component
3. Observe the transition from and to the lesson content

Postconditions:

- User proceeds with the lesson content after engagement

Expected Result: Interactive elements are smoothly embedded within lesson content

Severity: Major

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Data - 51

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Personalized Learning Paths

Feature description -

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities

aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

Userstory title -

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

Acceptance criteria -

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC001

Test Case Title: Verify that learners can choose courses and modules based on their interests.

Test Case Description: This test case verifies whether learners can select courses and modules based on their unique interests.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

1. The learner has registered and logged into the platform.
2. The platform has enough course and module options related to various interests.

Test Data: Learner's credentials, interest topics

Test Steps:

1. Log into the platform.
2. Navigate to the courses page.
3. Search and select courses and modules based on interests.

Postconditions:

1. The selected courses and modules are added to the learner's personalized learning path.
2. The platform records the learner's interests.

Expected Result: The learner successfully selects courses and modules based on personal interests.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress.

Test Case Description: This test case verifies whether the personalized learning path updates as the learner progresses through a course or module.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

1. The learner has courses and modules in their personalized learning path.
2. The learner has started learning.

Test Data: Learner's progress details

Test Steps:

1. Log into the platform.
2. Start learning from the personalized learning path.
3. Complete a module or course.

Postconditions:

1. Personal learning path reflects progress by marking completed courses or modules.
2. Recommendations are updated as per the completion.

Expected Result: The personalized learning path must update according to the learner's progression.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003

Test Case Title: Verify that learners receive recommendations for new courses and modules based on their learning path.

Test Case Description: This test case verifies whether the system recommends new courses and modules aligned with the learner's interests and progress.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

1. The learner has courses and modules in their personalized learning path.
2. The platform has variety of courses and modules related to those currently in the learner's path.

Test Data: Current learning path, learner preferences and progress

Test Steps:

1. Log into the platform.
2. Access the personalized learning path.
3. Check for new course and module recommendations.

Postconditions:

1. Learner has the ability to add recommended courses and modules to their path.
2. Recommendations are updated regularly as per the learner's progress and changes in their interests.

Expected Result: The platform recommends new courses and modules that align with the learner's interests and progress.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004

Test Case Title: Verify that progress tracking is personalized to each learner's path.

Test Case Description: This test case checks if the progress tracker accurately reflects individual's progress on their personalized learning path.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

1. The learner has active courses and modules in their personalized learning path.
2. The learner has started learning.

Test Data: Learner's progress data

Test Steps:

1. Log in to the platform.
2. Access the personalized learning path.
3. Check the progress tracker.

Postconditions: The progress tracker has been updated based on the learner's progress in their personalized course or module.

Expected Result: The progress tracker accurately shows the learner's progress on their personalized path.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005

Test Case Title: Verify that learners can adjust their learning paths at any time.

Test Case Description: This test case verifies whether learners can make changes to their learning paths at any time.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

1. The learner is logged in and has an active learning path.

Test Data: No test data needed

Test Steps:

1. Log into the platform.
2. Navigate to the personalized learning path.
3. Make adjustments to the learning path by adding or removing courses or modules.

Postconditions: The system reflects the changes made in the learner's personalized learning path.

Expected Result: The learner successfully adjusts their learning path.

Severity: Major

Priority: High

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:

Test Case 1:

Test Case ID: PT_1

Test Case Title: Verify that the progress tracking shows completed activities and grades.

Test Case Description: This test case checks if the progress tracking feature properly displays completed tasks and grades for a learner.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- User is registered and logged in.
- User has completed a few activities and received grades.

Test Data: User data, Completed activities, Grades.

Test Steps:

1. Login to the online platform as a learner.
2. Navigate to the Progress Tracking section.
3. Check for the list of completed activities and grades.

Postconditions:

- Data is displayed either in a list or graphical form.
- User can navigate back to the courses.

Expected Result: The progress tracking feature correctly shows the completed activities and the associated grades.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: PT_2

Test Case Title: Verify that learners can view detailed reports on their progress.

Test Case Description: This test ensures that learners are able to access and view comprehensive reports detailing their progress.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- User is registered and logged in.
- User has some progress in a course.

Test Data: User data, Progress report data.

Test Steps:

1. Login to the platform as a learner.
2. Navigate to the Progress Tracking section.
3. Click on the 'View Detailed Report' option.

Postconditions:

- Detailed progress report is displayed.
- User can download or print the report.

Expected Result: The learner can successfully view a detailed progress report.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: PT_3

Test Case Title: Verify that achievements are highlighted in the progress tracker.

Test Case Description: This test case verifies that any achievements or badges earned by the learner are showcased in the Progress Tracking section.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- User is registered and logged in.
- User has earned some achievements/badges.

Test Data: User data, Achievements data.

Test Steps:

1. Login to the platform as a learner.
2. Navigate to the Progress Tracking section.
3. Check for showcased achievements/badges.

Postconditions:

- Achievements are displayed with relevant icons or badges.

Expected Result: The progress tracker correctly displays the achievements/badges earned by the learner.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: PT_4

Test Case Title: Verify that learners receive notifications about their progress.

Test Case Description: This test case checks if the system sends out notifications about progress updates to the learner.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

- User is registered and logged in.
- User has some progress in a course.
- Notifications are enabled for the user.

Test Data: User data, Notification text.

Test Steps:

1. Login to the platform as a learner.
2. Complete a new learning activity.
3. Check for the reception of a progress notification.

Postconditions:

- Notification received for activity completion.

Expected Result: Learner successfully receives notifications regarding progress updates.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: PT_5

Test Case Title: Verify that progress tracking can be exported as reports.

Test Case Description: This test ensures that the learner can export their progress tracking data as a detailed report.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- User is registered and logged in.
- User has some progress in a course.

Test Data: User data, Progress report data.

Test Steps:

1. Login to the platform as a learner.
2. Navigate to the Progress Tracking section.
3. Select the 'Export Report' option.

Postconditions:

- Download of report starts.
- User can open and view the downloaded report.

Expected Result: The learner successfully exports their progress tracking data as a detailed report.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 53

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Assessment and Grading

Feature description -

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

Userstory title -

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide

feedback.

Acceptance criteria -

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

Generated Test Cases:

Test Case 1

Test Case ID: TC_001

Test Case Title: Verify that instructors are able to create quizzes, tests, and assignments.

Test Case Description: This test case validates the functionality that allows instructors to create various types of assessments like quizzes, tests, and assignments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions: Logged in as an instructor

Test Data: Quiz, Test, and Assignment details

Test Steps:

1. Login to the platform as an instructor
2. Navigate to 'Create Assessment' option
3. Select the type of assessment (quiz, test, assignment)
4. Fill in the necessary assessment details
5. Click on 'Save' or 'Create' button

Postconditions: Assessment created successfully and displayed in the list of instructor's assessments

Expected Result: The created assessment should be saved and should show up in the list of instructor's assessments.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_002

Test Case Title: Verify that the assessments can include different question types.

Test Case Description: This test case examines the platform's ability to accept and display varied types of question formats like MCQs, short answers, and essays in the assessments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions: Logged in as an instructor

Test Data: Details of different types of questions

Test Steps:

1. Login to the platform as an instructor
2. Navigate to 'Create Assessment' option
3. Input various types of questions including MCQs, short answers, and essays
4. Save the assessment

Postconditions: Assessment with various question types created successfully

Expected Result: The assessment should be saved and should consist of multiple formats of questions.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_003

Test Case Title: Verify that automatic grading is available for objective questions.

Test Case Description: This test case validates the feature that allows automatic grading for objective type questions in an assessment.

Test Suite: Assessment and Grading

Test Priority: Medium

Preconditions: Logged in as a learner

Test Data: Answers to objective type questions

Test Steps:

1. Login to the platform as a learner
2. Start an assessment with objective type questions
3. Submit the assessment after answering
4. Check the grade

Postconditions: Automatic grading applied to the submitted answers

Expected Result: The system should automatically provide grades for the objective type questions soon after the assessment is submitted.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC_004

Test Case Title: Verify that manual grading is available for subjective questions.

Test Case Description: This test case ensures that instructors can manually grade subjective type questions in

an assessment.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions: Logged in as an instructor

Test Data: Student's answer scripts

Test Steps:

1. Login to the platform as an instructor
2. Navigate to 'Review Submissions' for an assessment with subjective questions
3. Manually grade the submitted answers
4. Save the grades

Postconditions: Manual grading applied to the submitted answers

Expected Result: The instructor should be able to input and save grades manually for the subjective type questions.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC_005

Test Case Title: Verify that feedback is provided to learners after grading.

Test Case Description: This test case validates that feedback is given to the learners after their assessments have been graded.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions: Learner has submitted an assessment

Test Data: Instructor's feedback

Test Steps:

1. Login to the platform as a learner
2. Navigate to the 'Submitted Assessments' section
3. Click on the assessment to review the grades and feedback

Postconditions: Feedback displayed to the learner

Expected Result: The learners should be able to view the feedback given by the instructor for their submitted assessment.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 54

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Forums

Feature description -

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge

sharing, enriching the learning experience through diverse perspectives and active participation.

Userstory title -

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

Acceptance criteria -

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all devices.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC1_DF_01

Test Case Title: Verify that learners can post questions and responses in forums

Test Case Description: This test case will confirm that learners are able to successfully post questions and responses in the dedicated discussion forums of the platform.

Test Suite: Discussion Forums

Test Priority: High

Preconditions: The learner is registered & logged in and the discussion forum page is accessible.

Test Data: Login credentials of a learner, sample question and response text.

Test Steps:

1. Log in with the learner credentials.
2. Navigate to a course's discussion forum.
3. Click on "Ask a Question" or "Reply to a Post".
4. Insert sample text.
5. Click on "Post?".

Postconditions: The post/question appears on the forum correctly.

Expected Result: The question or response is successfully posted in the forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC2_DF_02

Test Case Title: Verify that forums support threaded discussions

Test Case Description: This test case validates that the platform supports threaded discussions within forum posts.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions: A forum post exists and the learner is logged in.

Test Data: Login credentials of a learner, sample response text.

Test Steps:

1. Login with learner's credentials.
2. Navigate to an existing forum post.
3. Click on "Reply" for a specific comment.
4. Insert sample reply text.
5. Click on "Post?".

Postconditions: The reply gets nested under the targeted comment in a threaded structure.

Expected Result: Threaded discussions are supported within forum posts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC3_DF_03

Test Case Title: Verify that instructors can moderate discussions

Test Case Description: This test case ensures that instructors are able to moderate discussions by deleting inappropriate posts or comments within their course forum.

Test Suite: Discussion Forums

Test Priority: High

Preconditions: Instructor account & login, a post or comment exists for moderation.

Test Data: Instructor login credentials.

Test Steps:

1. Login with instructor credentials.
2. Navigate to their relevant course's discussion forum.
3. Identify a post or comment for moderation.
4. Click on "Delete" or "Edit".
5. Confirm moderation action.

Postconditions: The post/comment is appropriately edited or deleted.

Expected Result: Instructor is able to moderate (edit/delete) discussions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC4_DF_04

Test Case Title: Verify that learners receive notifications for new posts

Test Case Description: This test case verifies that learners receive appropriate notifications when a new post is made in their enrolled course forum.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions: A learner is logged in, a new post is made in the forum of an enrolled course.

Test Data: Login credentials of a learner.

Test Steps:

1. Login with learner's credentials.
2. Navigate to the notification area.
3. Check if there's a new notification for the recent post.

Postconditions: The notification area updates with new notifications.

Expected Result: Learners receive notifications for new posts in the discussion forums of their enrolled courses.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC5_DF_05

Test Case Title: Verify that forums are accessible on all devices

Test Case Description: This test case validates that the discussion forums are operable on various devices such as desktop, tablet and mobile device.

Test Suite: Discussion Forums

Test Priority: High

Preconditions: A device with internet connection; installed and functional web browser.

Test Data: No test data needed

Test Steps:

1. Open the web browser on the device.
2. Navigate to the online learning platform.
3. Log in with learner's credentials.
4. Navigate to a course's discussion forum.

Postconditions: The forum is operable & posts readable with appropriate formatting on various devices.

Expected Result: The discussion forums are accessible and operations works as expected on all devices.

Severity: Blocker

Priority: High

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:

Test Case ID: TC_01

Test Case Title: Verify that the platform supports video conferencing tools for live classes and webinars.

Test Case Description: This test case is designed to confirm that video conferencing tools function correctly, enabling learners to participate in live classes and webinars.

Test Suite: Live Classes and Webinars Suite

Test Priority: High

Preconditions: Learners and instructors have valid user accounts. A live class or webinar is scheduled.

Test Data: Username and password of a learner, instructor's username and password, and specific date and time for live class or webinar.

Test Steps:

1. Log in as a learner and navigate to the upcoming live class or webinar.
2. Check if the video conferencing tool interface appears.
3. Join the live class or webinar.
4. Ensure that audio and video transmission is clear and uninterrupted.

Postconditions: Learner leaves the live class or webinar.

Expected Result: Learner can successfully participate in a live class or webinar through the provided video

conferencing tools.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Compatibility Testing

Test Case Approach: Positive

Test Case ID: TC_02

Test Case Title: Verify that learners can ask questions in real-time during live classes and webinars.

Test Case Description: This test case checks the functionality of the platform's real-time Q&A system during live classes and webinars.

Test Suite: Live Classes and Webinars Suite

Test Priority: Medium

Preconditions: Learner is participating in a live class or webinar.

Test Data: No test data needed.

Test Steps:

1. Join a live class or webinar as a learner.
2. During the session, find and use the feature to ask questions.
3. Check if the instructor receives the question in real-time.

Postconditions: Learner leaves the live class or webinar.

Expected Result: Learner can post questions and they are received by the instructor in real-time.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

(Note: Continue to fill the remaining test cases with above format for the remaining acceptance criteria)

Data - 56

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Mobile Accessibility

Feature description -

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

Userstory title -

As a learner, I want mobile access to the platform so that I can learn on the go.

Acceptance criteria -

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_OLP_MA_01

Test Case Title: Verify that the platform is accessible on smartphones and tablets

Test Case Description: This test case verifies if the online learning platform is accessible and fully functional on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions: The learner has a smartphone or tablet with a good internet connection.

Test Data: No test data needed

Test Steps:

1. Open the device's web browser.
2. Enter the online learning platform's URL in the address bar.
3. On successful loading, validate the platform's functionalities.

Postconditions:

1. User logs out from the platform.

Expected Result: The online learning platform is fully accessible and functional on smartphones and tablets.

Severity: Blocker

Priority: High

Type of Testing: Compatibility Testing, Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_OLP_MA_02

Test Case Title: Verify that Mobile UI is optimized for touch navigation

Test Case Description: This test case is to ensure that the Mobile UI of the platform is easily navigable via touch.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions: The learner has accessed the learning platform on a touch-enabled device.

Test Data: No test data needed

Test Steps:

1. Open the online learning platform on mobile phone or tablet.
2. Try to navigate between various sections using touch.
3. Validate whether the touch response is working as expected.

Postconditions: None

Expected Result: The touch navigation on the platform works smoothly & accurately, with the platform responding correctly to touch inputs.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_OLP_MA_03

Test Case Title: Verify that course materials are accessible and readable on mobile devices

Test Case Description: The purpose of this test case is ensuring course documents, videos, and other resources are easily viewable and readable on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions: The learner has logged into his/her account on the platform using a mobile device.

Test Data: Access to various course contents.

Test Steps:

1. Access a course on the mobile device.
2. Browse through and open different types of course materials.

3. Validate the readability and accessibility of each type.

Postconditions: User logs off from the platform.

Expected Result: All course materials are accessible and readable without any issues on mobile devices.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_OLP_MA_04

Test Case Title: Verify that notifications are sent to mobile devices

Test Case Description: This test case is designed to confirm that mobile devices receive platform notifications correctly.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions: Learner has enabled platform notifications on the mobile device.

Test Data: None

Test Steps:

1. Perform an action on the platform that triggers a notification (e.g., course enrolment, lecture completion).
2. Wait for the notification to arrive on the mobile device.
3. Validate whether the notification arrives in a timely manner and displays proper content.

Postconditions: None

Expected Result: Notifications are correctly sent to the mobile devices and display the appropriate content.

Severity: Major

Priority: Medium

Type of Testing: Functional, Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_OLP_MA_05

Test Case Title: Verify that the platform supports offline access to downloaded materials

Test Case Description: Check if online learning platform provides offline access to the course materials downloaded on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions: Learner has downloaded course materials on mobile device.

Test Data: Downloaded course materials. Done

Test Steps:

1. Go into airplane mode or otherwise disable the mobile device's internet connection.
2. Open the app and try to access the downloaded materials.
3. Validate whether the materials are accessible and properly readable/viewable.

Postconditions: Turn mobile device's connection settings back on.

Expected Result: The downloaded materials should be accessible for offline viewing.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 57

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Resource Library

Feature description -

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

Userstory title -

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

Acceptance criteria -

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

Generated Test Cases:

Test Case 1

Test Case ID: TC01_RL

Test Case Title: Verify that the Resource Library includes e-books, articles, and reference guides.

Test Case Description: This test case verifies if the online Resource Library includes different types of resources - e-books, articles, and reference guides.

Test Suite: Resource Library

Test Priority: High

Preconditions: User logged in and navigates to Resource Library.

Test Data: No test data needed

Test Steps: 1. Navigate to Resource Library. 2. Review the types of resources available.

Postconditions: None

Expected Result: E-books, articles, and reference guides are included in the Resource Library.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC02_RL

Test Case Title: Verify that users can search for specific materials in the Resource Library.

Test Case Description: This test case validates if users can search for specific resources using the search function.

Test Suite: Resource Library

Test Priority: High

Preconditions: User logged in, navigates to Resource Library.

Test Data: Valid keyword or resource title for search

Test Steps: 1. Navigate to Resource Library. 2. Input keyword or resource title into the search bar. 3. Click on 'Search'.

Postconditions: User navigates to search results.

Expected Result: The search results show relevant resources based on the search input.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC03_RL

Test Case Title: Verify that the resources in the Library are categorized and tagged for easy navigation.

Test Case Description: This test case ensures that resources are properly categorized and tagged to aid navigation.

Test Suite: Resource Library

Test Priority: Medium

Preconditions: User logged in and navigates to Resource Library.

Test Data: No test data needed

Test Steps: 1. Navigate to Resource Library. 2. Browse through the resources catalog. 3. Check categories and tags associated with each resource.

Postconditions: None

Expected Result: All resources are categorized and tagged.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC04_RL

Test Case Title: Verify that learners can download resources for offline access.

Test Case Description: This test case validates the functionality that allows learners to download resources for offline use.

Test Suite: Resource Library

Test Priority: High

Preconditions: User logged in, navigates to Resource Library, a resource is selected.

Test Data: Resource file for download

Test Steps: 1. Navigate to Resource Library. 2. Select a resource. 3. Click 'Download'.

Postconditions: Resource file is downloaded on user's device.

Expected Result: Resource is successfully downloaded and can be accessed offline.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC05_RL

Test Case Title: Verify that instructors can recommend resources to learners.

Test Case Description: This test case checks if the feature allowing instructors to recommend resources to learners is working correctly.

Test Suite: Resource Library

Test Priority: Medium

Preconditions: User logged in as an Instructor, an appropriate resource for recommendation is selected.

Test Data: Reference to a resource for recommendation

Test Steps: 1. Login as an Instructor. 2. Navigate to Resource Library. 3. Select a resource. 4. Click 'Recommend' and select learners to recommend the resource to.

Postconditions: Learners receive the recommended resource.

Expected Result: Resource is successfully recommended to selected learners and they receive a recommendation alert.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Data - 58

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Peer Review

Feature description -

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

Userstory title -

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

Acceptance criteria -

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers remain anonymous to ensure unbiased feedback.']

Generated Test Cases:

Test Case 1:

- Test Case ID: TC_001_OPL

- Test Case Title: Verify that learners can submit assignments for peer review.
- Test Case Description: This case tests if the platform offers functionality to submit assignments for peer review, which is an important feature in the peer review process.
- Test Suite: Learner Functions
- Test Priority: High
- Preconditions:
 - The learner should be logged in.
 - An assignment should be ready for submission.
- Test Data: An assignment dummy file
- Test Steps:
 - Log in as a learner.
 - Navigate to the assignment submission page.
 - Upload the assignment.
 - Choose the option for peer review.
 - Submit the assignment.
- Postconditions: The assignment is submitted for peer review.
- Expected Result: The assignment should be successfully submitted for peer review, and there should be a confirmation message.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 2:

- Test Case ID: TC_002_OPL
- Test Case Title: Verify that peers can provide feedback on submitted assignments.
- Test Case Description: This case ensures that peers are able to successfully provide valuable feedback on

submitted assignments.

- Test Suite: Peer Review Functions
- Test Priority: High
- Preconditions:
 - Peer should be logged in.
 - An assignment should be available for review.
- Test Data: No test data needed
- Test Steps:
 - Log in as a peer.
 - Navigate to the assignment review page.
 - Choose an assignment to review.
 - Write feedback.
 - Submit the feedback.
- Postconditions: Feedback is submitted for the chosen assignment.
- Expected Result: Detailed feedback should be submitted successfully, and a confirmation message should be displayed.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 3:

- Test Case ID: TC_003_OPL
- Test Case Title: Verify that a structured review process guides the feedback.
- Test Case Description: This case checks the platform's laudable provision for a structured review system, guiding the feedback process appropriately.
- Test Suite: Peer Review Functions

- Test Priority: Medium
- Preconditions: User should be logged in.
- Test Data: No test data needed
- Test Steps:
 - Log into the platform.
 - Navigate to the assignment review page.
 - Begin to write feedback.
 - View the guidance/structure provided for writing feedback.
- Postconditions: The presence or absence of a guidance structure for feedback is determined.
- Expected Result: Guidelines or a structured process should be provided to guide the feedback process.
- Severity: Minor
- Priority: Medium
- Type of Testing: Usability Testing
- Test Case Approach: Positive

Test Case 4:

- Test Case ID: TC_004_OPL
- Test Case Title: Verify that learners can view feedback on their work.
- Test Case Description: This case tests if the learners can successfully view and receive detailed feedback on their submitted assignments.
- Test Suite: Learner Functions
- Test Priority: High
- Preconditions: The learner should have submitted an assignment and feedback should have been provided by a peer.
- Test Data: No test data needed
- Test Steps:
 - Log in as a learner.

- Navigate to the feedback page.
- Select an assignment.
- View the feedback.
- Postconditions: Learner can glean the feedback received on assignments.
- Expected Result: Feedback should be visible and accessible on the learner's dashboard.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 5:

- Test Case ID: TC_005_OPL
- Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback.
- Test Case Description: In this crucial case, we check if the identity of the peers providing feedback is kept anonymous to maintain unbiased evaluations.
- Test Suite: Security Functions
- Test Priority: High
- Preconditions: Peer should have provided feedback on an assignment.
- Test Data: No test data needed
- Test Steps:
 - Log in as a learner.
 - Navigate to the feedback page.
 - Select an assignment.
 - Check the feedback details.
- Postconditions: Anonymity of feedback provider is ensured.
- Expected Result: Reviewer's identification details should not be visible on the feedback page.
- Severity: Critical

- Priority: High
 - Type of Testing: Security Testing
 - Test Case Approach: Negative
-

Data - 59

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Certificates and Badges

Feature description -

Upon successful completion of courses or specific milestones, learners receive certificates and badges to recognize their achievements and accomplishments. These credentials provide tangible recognition of learners' efforts, skills, and expertise, enhancing their credentials and demonstrating their commitment to continuous learning and professional development.

Userstory title -

As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

Acceptance criteria -

['Learners earn certificates upon course completion.', 'Badges are awarded for specific achievements and milestones.', 'Certificates and badges are accessible through learner profiles.', 'Learners can share their certificates and badges on social media.', 'Instructors can issue custom badges for special achievements.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_OLP_01

Test Case Title: Verify that learners earn certificates upon course completion.

Test Case Description: Test that a certificate is automatically issued to the learner profile once they have successfully completed a course.

Test Suite: Course Completion and Certificate Generation

Test Priority: High

Preconditions: The learner has successfully completed a course.

Test Data: Course completion details (Course ID, Learner ID)

Test Steps:

1. Login as a Learner.
2. Complete a course.
3. Check the learner profile for a certificate.

Postconditions: The certificate is added to the learner profile.

Expected Result: The certificate is visible in the learner's profile after completing the course.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_OLP_02

Test Case Title: Verify that badges are awarded for specific achievements and milestones.

Test Case Description: Testing to check if the learner receives badges upon hitting specified milestones or achieving special feats within the course.

Test Suite: Milestones and Badge Awards

Test Priority: High

Preconditions: The learner has achieved a specific milestone.

Test Data: Milestone data (Milestone ID, Learner ID)

Test Steps:

1. Login as a Learner.
2. Achieve a milestone in a course.
3. Check the learner profile for the badge.

Postconditions: Badge is added to learner's profile.

Expected Result: The badge is visible in the learner's profile upon achieving a milestone.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_OLP_03

Test Case Title: Verify that learners can share their certificates and badges on social media.

Test Case Description: Testing to verify if learners are able to share their achieved certificates and badges on their chosen social media platforms.

Test Suite: Social Media Sharing

Test Priority: Medium

Preconditions:

1. The learner has earned a badge or certificate.
2. The learner has a social media account.

Test Data: No test data needed.

Test Steps:

1. Login as a Learner.
2. Navigate to the badges or certificates section.
3. Click on the share to social media button.
4. Login to social media account on the prompt.
5. Post the badge or certificate on the social media account.

Postconditions: The certificate or badge is shared on the learner's social media profile.

Expected Result: Successful posting of badge or certificate on the learner's social media profile.

Severity: Minor

Priority: Medium

Type of Testing: User Acceptance Testing, Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_OLP_04

Test Case Title: Verify that Instructors can issue custom badges for special achievements.

Test Case Description: Conduct a test to validate the ability of Instructors to issue custom badges when learners reach special achievements.

Test Suite: Instructor Badge Issuing

Test Priority: Medium

Preconditions:

1. The Instructor has an active account.
2. The learner has achieved a special feat.

Test Data: Learner ID, Badge details.

Test Steps:

1. Login as an Instructor.

2. Navigate to the badges creation section.
3. Create a custom badge for a learner based on their special achievement.
4. Issue the badge to the learner.

Postconditions:

1. Badge is added to the learner's profile.

Expected Result: The custom badge is visible in the learner's profile after issued by the instructor.

Severity: Minor

Priority: Medium

Type of Testing: Administrative Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_OLP_05

Test Case Title: Verify that certificates and badges are inaccessible through unregistered learner profiles.

Test Case Description: Testing to ensure the system's security. Only registered and verified learners should be able to access certificates and badges. Unauthorized access should be prevented.

Test Suite: Access Control

Test Priority: High

Preconditions: Test with an unregistered user profile.

Test Data: No test data needed.

Test Steps:

1. Login as an unauthorized (unregistered) user.
2. Attempt to access the badges or certificates section.

Postconditions: No specific postconditions.

Expected Result: Unauthorized user should not be able to access badges or certificates.

Severity: Blocker

Priority: High

Type of Testing: Security Testing, Negative Testing

Test Case Approach: Negative

Data - 60

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Gamification Elements

Feature description -

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

Userstory title -

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

Acceptance criteria -

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

Generated Test Cases:

Test Case ID: TC_G01

Test Case Title: Verify that learners can earn points upon completing activities.

Test Case Description: This scenario tests the functionality of the points system and checks the proper assignment of points to learners after they finish learning activities.

Test Suite: Gamification

Test Priority: High

Preconditions:

- Learner account exists.
- Learner is enrolled in a course.
- Learning activity is available for completion.

Test Data: Name of the course, Name of the learning activity

Test Steps:

1. Log in to the online learning platform using a learner account.
2. Navigate to an ongoing course.
3. Complete a learning activity.

Postconditions:

- Points are added to the learner's account.
- Notification about earned points is displayed.

Expected Result: The learner's total points increase as per the points gained from the completed activity.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_G02

Test Case Title: Verify that learners achieve levels based on accumulated points.

Test Case Description: This scenario tests if the leveling system works as intended based on the amount of

accumulated points by the learners.

Test Suite: Gamification

Test Priority: High

Preconditions:

- Learner has earned points.
- Level thresholds are set.

Test Data: Learner's points

Test Steps:

1. Log in to the platform with a learner account.
2. Check the current level of the learner.
3. Earn points to cross the level threshold.

Postconditions:

- Learner's level is updated.
- Notification about level upgrade is displayed.

Expected Result: Learner's level increases after crossing the level threshold.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_G03

Test Case Title: Verify that leaderboards display top-performing users based on points and levels.

Test Case Description: This scenario verifies the leaderboard functionality based on the rankings of top-performing learners.

Test Suite: Gamification

Test Priority: Medium

Preconditions:

- Multiple learners have earned points and attained levels.
- Leaderboard feature is correctly implemented.

Test Data: Learners' scores and levels

Test Steps:

1. Log into the online learning platform using any account.
2. Navigate to the leaderboard page.

Postconditions:

- Leaderboard displays top learners based on points and levels.
- Leaderboard is updated in real time.

Expected Result: Leaderboard correctly displays the top-performing users based on points and levels.

Severity: Minor

Priority: Medium

Type of Testing: Non-Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_G04

Test Case Title: Verify that badges are awarded and displayed on learner's profile for reaching milestones.

Test Case Description: This scenario checks if badges are correctly assigned and displayed upon reaching predefined milestones.

Test Suite: Gamification

Test Priority: High

Preconditions:

- Learner achieved a milestone.
- Badge for that milestone is available.

Test Data: Milestone details, Badge details

Test Steps:

1. Log into the online learning platform using a learner account

2. Complete a milestone.

3. Check the profile for newly achieved badge.

Postconditions:

- Badge is added to learner's profile.
- Notification of badge reward is displayed.

Expected Result: Badges are correctly awarded and displayed on learner's profile after reaching a milestone.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_G05

Test Case Title: Verify that learners can view their progress in the gamification dashboard.

Test Case Description: This scenario tests if learners can view their gamification progress such as points, levels, and badges earned on a dedicated dashboard.

Test Suite: Gamification

Test Priority: Medium

Preconditions:

- Learner account is active.
- Learner has accumulated points and badges.

Test Data: Login credentials, Learner's gamification details

Test Steps:

1. Log in to the online learning platform using a learner account.
2. Navigate to the gamification dashboard.
3. Validate the visible information.

Postconditions: Gamification dashboard shows the learner's progress accurately.

Expected Result: Learners can view their points, levels, leaderboard rank, and badges on the gamification

dashboard.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Data - 61

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Multilingual Support

Feature description -

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

Userstory title -

As a learner, I want multilingual support so that I can access the platform in my preferred language.

Acceptance criteria -

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated

help and support resources.', 'Instructors can provide content in multiple languages.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_OLP_MLS_001

Test Case Title: Verify that the platform interface is available in multiple languages.

Test Case Description: Testing if the online learning platform correctly displays interfaces in the selected user language.

Test Suite: Multilingual Support

Test Priority: High

Preconditions: The user is logged into the platform.

Test Data: User profile with preferred language set to a non-default language.

Test Steps:

1. User logs in and navigates to profile setting.
2. User changes preferred language to a non-default language.
3. User saves settings and refreshes the page.

Postconditions: The user can continue to interact with the platform in the selected language.

Expected Result: The interface of the platform is properly translated into the selected language.

Severity: Major

Priority: High

Type of Testing: Localization Testing, Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_OLP_MLS_002

Test Case Title: Verify that the course materials are available in different languages.

Test Case Description: This test case validates if the course materials are provided in the user's selected language.

Test Suite: Multilingual Support

Test Priority: High

Preconditions: The user has an active account and is enrolled in a course.

Test Data: User account, Course materials in multiple languages.

Test Steps:

1. User logs in and navigates to the course materials section for an enrolled course.
2. User selects the preferred language of course materials.
3. User reviews the materials in the chosen language.

Postconditions: The user can proceed with the course in the chosen language.

Expected Result: All course materials render correctly in the selected language.

Severity: Major

Priority: High

Type of Testing: Localization Testing, Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_OLP_MLS_003

Test Case Title: Verify that Learners can select their preferred language at any time.

Test Case Description: This test case checks if the language selection functionality works accurately at any given time.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions: The user is logged in to the platform.

Test Data: User profile with preferred language set to a non-default language.

Test Steps:

1. User logs in to his account.
2. User navigates to user profile settings.
3. User changes the preferred language to a non-default language at different times.
4. User saves settings.

Postconditions: The user's selected language is saved properly for future sessions.

Expected Result: The platform permits selection of the preferred language at any time and reflects the change without error.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_OLP_MLS_004

Test Case Title: Verify that multilingual support includes translated help and support resources.

Test Case Description: Testing to ensure that help and support resources are comprehensively and accurately translated in the selected user language.

Test Suite: Multilingual Support

Test Priority: High

Preconditions: The user is logged into the platform.

Test Data: User profile with preferred language set to a non-default language.

Test Steps:

1. User logs in and switches the platform language to a non-default language.
2. User navigates to Help and Support section.

3. User views the Help and Support resources.

Postconditions: The user can contact support in their chosen language.

Expected Result: All Help and Support resources are correctly translated into the user-selected language.

Severity: Major

Priority: High

Type of Testing: Localization Testing, Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_OLP_MLS_005

Test Case Title: Verify that instructors can provide content in multiple languages.

Test Case Description: This test case ensures that instructors can add course materials in different languages.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions: The instructor is logged in to the platform.

Test Data: Instructor account, Course contents in different languages.

Test Steps:

1. Instructor logs in and navigates to course creation section.
2. Instructor adds course material in different languages.
3. Instructor saves the course.

Postconditions: Course materials in different languages are available for learners.

Expected Result: Instructors can successfully upload course content in different languages.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 62

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Accessibility Features

Feature description -

The platform prioritizes accessibility by incorporating features such as screen reader compatibility, adjustable font sizes, and color contrast options. These accessibility features ensure that learners with disabilities can access and navigate the platform with ease, promoting inclusivity and equitable access to educational opportunities.

Userstory title -

As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

Acceptance criteria -

['The platform is compatible with screen readers.', 'Learners can adjust font sizes for better readability.', 'Color contrast options are available for improved visibility.', 'Accessibility features are easy to activate and customize.', 'Support resources are available to assist learners with disabilities.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_OLP_A_001

Test Case Title: Verify that the platform is compatible with screen readers.

Test Case Description: The test is designed to ensure the Online Learning Platform's compatibility with various screen readers.

Test Suite: Accessibility Testing

Test Priority: High

Preconditions:

- User has a registered account
- User has a screen reader installed

Test Data: No test data needed

Test Steps:

- Log into the platform
- Turn on the screen reader
- Navigate through the site

Postconditions:

- Record the screen reader's interpretation
- Log out of the platform

Expected Result: The site content should be accurately read by the screen reader

Severity: Critical

Priority: High

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_OLP_A_002

Test Case Title: Verify that learners can adjust font sizes for better readability.

Test Case Description: This case is to test the feature that allows users to adjust the font sizes across the

platform for improved readability.

Test Suite: Usability Testing

Test Priority: High

Preconditions:

- User has a registered account
- User is logged in

Test Data: No test data needed

Test Steps:

- Go to platform settings
- Adjust the font size
- Visit various pages of the platform

Postconditions:

- Record the visibility of text with adjusted font size

Expected Result: The font size should be adjustable across all pages of the platform.

Severity: Major

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_OLP_A_003

Test Case Title: Verify that color contrast options are available for improved visibility.

Test Case Description: This test validates the presence and functionality of color contrast options to improve visibility for impaired users.

Test Suite: Accessibility Testing

Test Priority: High

Preconditions:

- User has a registered account
- User is logged in

Test Data: No test data needed

Test Steps:

- Go to platform settings
- Change the color contrast
- Visit various pages on the platform

Postconditions:

- Record how the color contrast change affects platform appearance

Expected Result: The platform should allow the adjustment of color contrast, and it should be effective across all its pages.

Severity: Major

Priority: High

Type of Testing: Accessibility Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_OLP_A_004

Test Case Title: Verify that accessibility features are easy to activate and customize.

Test Case Description: This test checks if the accessibility features on the online learning platform can be easily activated and customized by the users themselves.

Test Suite: Accessibility Testing

Test Priority: High

Preconditions:

- User has a registered account

- User is logged in

Test Data: No test data needed

Test Steps:

- Go to platform settings
- Try to activate and customize different accessibility features

Postconditions:

- Record how easy it is to activate and customize each feature

Expected Result: All the accessibility features should be easy to activate and customize for the user.

Severity: Critical

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_OLP_A_005

Test Case Title: Verify that support resources are available to assist learners with disabilities.

Test Case Description: This case is to ensure that ample support resources, such as guides and tutorials, are available to assist learners with disabilities in using the platform.

Test Suite: Functionality Testing

Test Priority: High

Preconditions:

- User has a registered account
- User is logged in

Test Data: No test data needed

Test Steps:

- Navigate to the support or help section of the platform

- Search for resources regarding the use of accessibility features

Postconditions:

- Record the availability and comprehensiveness of support resources

Expected Result: Support resources specific to the usage of accessibility features should be available and easy to understand.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 63

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Moderation

Feature description -

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

Userstory title -

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

Acceptance criteria -

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

Generated Test Cases:

Test Case 1:

Test Case ID: TCFD001

Test Case Title: Verify that instructors can edit, delete or highlight posts in discussions

Test Case Description: This test case focuses on checking the functionality of post moderation by instructors.

Test Suite: Discussion Suite

Test Priority: High

Preconditions:

- Instructor account is active
- The Course should have at least one active discussion
- Post in the discussion which needs moderation

Test Data: No test data needed

Test Steps:

1. Log in as an instructor
2. Navigate to active discussions
3. Choose a post to moderate
4. Use moderation tools to edit, delete or highlight posts

Postconditions:

- Changes are reflected immediately

- Record of moderation activity is generated

Expected Result: The instructor successfully modifies the post using the moderation tools

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TCFD002

Test Case Title: Verify that instructors can mute or ban disruptive users

Test Case Description: This test case checks the function of user control available to instructors for maintaining decorum.

Test Suite: Discussion Suite

Test Priority: High

Preconditions:

- Instructor account is active
- At least one disruptive user in the discussion

Test Data: Details of a disruptive user

Test Steps:

1. Log in as an instructor
2. Navigate to disruptive user profile
3. Use control settings to mute/ban the user

Postconditions:

- Affected user receives notification
- Record of control action is generated

Expected Result: The instructor successfully mutes/bans the disruptive user

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TCFD003

Test Case Title: Verify that moderation actions are logged and reported

Test Case Description: This test case examines whether the platform logs and reports moderation actions effectively.

Test Suite: Discussion Suite

Test Priority: High

Preconditions:

- Instructor account is active
- At least one active discussion

Test Data: Moderated discussion data

Test Steps:

1. Log in as an instructor
2. Initiate a moderation activity
3. Navigate to moderation logs

Postconditions:

- Log entry for each moderation activity exists
- Report displays recent activity

Expected Result: Moderation activities are successfully recorded and reported

Severity: Critical

Priority: High

Type of Testing: System Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TCFD004

Test Case Title: Verify that community guidelines are visible to all participants

Test Case Description: This test case checks the visibility and accessibility of community guidelines.

Test Suite: General Suite

Test Priority: Medium

Preconditions:

- Instructor and Student accounts are active

Test Data: No test data needed

Test Steps:

1. Log in from both accounts
2. Navigate to community guidelines page

Postconditions:

- User returns to discussion or course home page

Expected Result: Community guidelines are visible and accessible to all participants

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TCFD005

Test Case Title: Verify that notifications are sent for moderated actions

Test Case Description: This test case validates the system's ability to notify users about any moderation action taken.

Test Suite: Notification Suite

Test Priority: Medium

Preconditions:

- Instructor account is active
- At least one user available for moderation action

Test Data: No test data needed

Test Steps:

1. Log in as an instructor
2. Initiate moderation action on a user
3. User checks for notification

Postconditions:

- User confirms notification receipt

Expected Result: Notifications are correctly sent for any moderated actions

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Negative

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Analytics

Feature description -

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement, performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning experience to meet learners' needs effectively.

Userstory title -

As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

Acceptance criteria -

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.', 'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.', 'Instructors can export analytics reports for further analysis.']

Generated Test Cases:

Test Case 1

Test Case ID: TC_OLP_LAD_001

Test Case Title: Verify that the learning analytics dashboard displays engagement metrics accurately.

Test Case Description: This test case is designed to validate the functionality of the learning analytics

dashboard, ensuring that it accurately displays learner engagement metrics such as activity, course completion, and forum participation.

Test Suite: Learning Analytics Dashboard Suite

Test Priority: High

Preconditions:

- Instructor is logged into the platform.
- There is course data for the instructor to examine.

Test Data: Course data with varying engagement metrics.

Test Steps:

1. Log into the instructor account.
2. Navigate to the learning analytics dashboard.
3. Review the displayed engagement metrics.

Postconditions:

- Data verified for accuracy.
- Possible improvement in course or revision initiated (if needed).

Expected Result: The dashboard accurately displays learner engagement metrics.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_OLP_LAD_002

Test Case Title: Verify that performance trends are accurately visualized over time.

Test Case Description: This test case is designed to validate that performance trends, such as improvement or decline in learner scores, are accurately visualized in graphs or charts over specified timeframes.

Test Suite: Learning Analytics Dashboard Suite

Test Priority: High

Preconditions:

- Instructor is logged into the platform.
- There is course data for the instructor to examine.

Test Data: Course data with varying performance trends.

Test Steps:

1. Log into the instructor account.
2. Navigate to the learning analytics dashboard.
3. Review the visualized performance trends over chosen timeframes.

Postconditions:

- Data verified for accuracy.
- Possible improvement in course or revision initiated (if needed).

Expected Result: The platform accurately visualizes performance trends over time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_OLP_LAD_003

Test Case Title: Verify that instructors can access detailed reports on learner progress.

Test Case Description: This test case is designed to validate the accessibility of detailed reports on learner progress, including course completion, module scores, and overall assessment performance.

Test Suite: Learning Analytics Dashboard Suite

Test Priority: High

Preconditions:

- Instructor is logged into the platform.
- There is course data for the instructor to examine.

Test Data: Course data with varying learner progress.

Test Steps:

1. Log into the instructor account.
2. Navigate to the learning analytics dashboard.
3. Access the learner progress reports.

Postconditions:

- Data verified for accuracy.
- Possible improvement in course or revision initiated (if needed).

Expected Result: Instructors can successfully access the detailed learner progress reports.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_OLP_LAD_004

Test Case Title: Verify that analytics data is updated in real-time.

Test Case Description: This test case is designed to verify the real-time update feature of the analytics data.

This ensures that instructors have access to the most current and accurate data.

Test Suite: Learning Analytics Dashboard Suite

Test Priority: High

Preconditions:

- Instructor is logged into the platform.

- A learner is actively participating in a course.

Test Data: Real-time data as learners progress through a course.

Test Steps:

1. Log into the instructor account.
2. Navigate to the learning analytics dashboard.
3. Observe as a learner progresses in a course, checking if data reflects these changes in real-time.

Postconditions:

- Data verified for real-time updates.

Expected Result: The analytics data is updated in real-time, reflecting learners' current progress and engagement.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_OLP_LAD_005

Test Case Title: Verify that instructors can export analytics reports for further analysis.

Test Case Description: This test case checks the functionality of the report export feature, ensuring instructors can download and analyze learner engagement and performance data outside of the platform.

Test Suite: Learning Analytics Dashboard Suite

Test Priority: High

Preconditions:

- Instructor is logged into the platform.
- There is course data for the instructor to examine.

Test Data: No test data needed.

Test Steps:

1. Log into the instructor account.
2. Navigate to the learning analytics dashboard.
3. Generate a analytics report and enact the export feature.

Postconditions:

- Exported analytics report available for further offline analysis.

Expected Result: Instructors can successfully export analytics reports in a chosen format (PDF, CSV, etc.)

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 65

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Learning Features

Feature description -

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and

promote peer-to-peer support and mentorship.

Userstory title -

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Acceptance criteria -

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

Generated Test Cases:

Test Case 1

Test Case ID: TC001

Test Case Title: Verify that learners can create and join groups for projects.

Test Case Description: This test case checks the functionality of creating and joining groups for projects from a learner's perspective.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner account is setup
- Active internet connection

Test Data: No test data needed

Test Steps:

1. Login to the learner's account.
2. Navigate to the 'Groups' section.
3. Click on 'Create New Group'.
4. Fill in the required details for the new group.
5. Click on 'Join' on any existing group.

Postconditions:

- Group created by the learner is added to the available groups.
- Learner is added to the member list of the joined group.

Expected Result: Learner successfully creates and joins groups.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC002

Test Case Title: Verify that collaboration tools are available for group communication.

Test Case Description: This test case verifies the presence and functionality of collaboration tools like chat, video conferencing, etc. for group communication.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is part of a group
- Active internet connection

Test Data: No test data needed

Test Steps:

1. Login to the learner's account.
2. Navigate to the group's discussion page.
3. Use various collaboration tools like chat and video conferencing.

Postconditions:

- Learner is able to communicate with the group.

Expected Result: All collaboration tools are functional.

Severity: Major

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC003

Test Case Title: Verify that learners can share resources and feedback within groups.

Test Case Description: This test case checks the functionality of sharing resources and providing feedback within a group.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is part of a group
- Active internet connection

Test Data: Group discussion topics, group resources

Test Steps:

1. Login to the learner's account.
2. Navigate to the group page.
3. Share a resource or provide feedback in the discussion.

Postconditions:

- Shared resource or feedback appears in the group's discussion feed.

Expected Result: Learners can share resources and feedback successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC004

Test Case Title: Verify that instructors can assign group projects and monitor progress.

Test Case Description: This test case ensures that instructors can assign projects to groups and monitor their progress.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Instructor account is setup
- Groups are created
- Active internet connection

Test Data: Group Project details

Test Steps:

1. Login to the instructor's account.
2. Navigate to the 'Assign Projects' page.
3. Assign a new project to a group.
4. Navigate to 'Monitor Progress' page.

Postconditions:

- The assigned group project appears in the group's project list.
- The group progress appears on 'Monitor Progress' page.

Expected Result: Instructor can assign projects and monitor group progress successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC005

Test Case Title: Verify that social sharing features allow learners to share achievements.

Test Case Description: This test case checks if learners can share their achievements using social sharing features.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner has completed a course/achievement
- Active internet connection

Test Data: Course completion data

Test Steps:

1. Login to the learner's account.
2. Navigate to 'My Achievements' page.
3. Click on 'Share'.

Postconditions:

- A post containing the achievement is generated on the chosen social media platform.

Expected Result: Learners can share their achievements on social media platforms successfully.

Severity: Minor

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:

Test Case ID: TC001

Test Case Title: Verify that instructors can create multimedia presentations using content authoring tools

Test Case Description: In this test case, we will verify the functionality of creating multimedia presentations by instructors using the content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- User must be logged in as an Instructor
- User must have relevant permissions to create a presentation

Test Data: Multimedia files such as audio, video, images

Test Steps:

1. Login as Instructor
2. Navigate to the content authoring tools section
3. Select 'Create a presentation'
4. Import or create the multimedia content and save the presentation

Postconditions:

- Persistent storage of the multimedia presentation
- Instructor should be able to view the created multimedia presentation in their content library

Expected Result: Instructor should be able to successfully create a multimedia presentation

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC002

Test Case Title: Verify that interactive modules can be developed and integrated into the course content

Test Case Description: This test case is designed to verify the capability of the content authoring tools to create and integrate interactive modules in course content

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- User must be logged in as an Instructor
- User must have relevant permissions to create an interactive module

Test Data: Interactive module data such as quizzes, clickable graphics, flashcards

Test Steps:

1. Login as Instructor
2. Navigate to the content authoring tools section
3. Choose to create an interactive module
4. Configure and save the interactive module

Postconditions:

- The interactive module is stored for future access
- The interactive module is viewable in the Instructor's content library

Expected Result: Instructor can successfully create and integrate an interactive module into the course content

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Further test cases will explore the content authoring tools' support for rich media resources, user-friendliness and intuitive nature, and the ability for instructors to preview and edit content before publishing.

Data - 67

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Paths and Curriculum

Feature description -

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence and continuity in learners' educational journeys.

Userstory title -

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

Acceptance criteria -

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

Generated Test Cases:

Test Case 1

Test Case ID: TC01_LP_01

Test Case Title: Verify that learning paths include sequential courses and modules

Test Case Description: Testing the learning paths feature to verify it provides sequential progression through courses and modules, facilitating a systematic learning journey.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Valid user account
- User logged into the online learning platform

Test Data: No test data needed

Test Steps:

1. Login to the online learning platform
2. Navigate to the Learning Paths section
3. Open a learning path
4. Check if courses and modules are presented in sequential order

Postconditions:

- User has verified the order of content within a learning path

Expected Result: Courses and modules within a learning path should be arranged sequentially

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC02_LP_02

Test Case Title: Verify that curricula are aligned with specific learning objectives

Test Case Description: Testing the curricula feature to ensure they are tailored to match specific learning objectives, thereby providing clear learning outcomes.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Valid user account
- User logged into the online learning platform

Test Data: No test data needed

Test Steps:

1. Login to the online learning platform
2. Navigate to a curriculum
3. Review the learning objectives of the curriculum
4. Compare curriculum content with stated learning objectives

Postconditions:

- User has confirmed the alignment of curriculum content with learning objectives

Expected Result: The curriculum content should correlate directly with the stated learning objectives

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC03_LP_03

Test Case Title: Verify that learners can track their progress within a learning path

Test Case Description: Testing the feature that allows learners to track their progress within the learning path, enhancing their control over their learning journeys.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Valid user account
- User logged into the online learning platform

- User enrolled in a learning path

Test Data: No test data needed

Test Steps:

1. Login to the online learning platform
2. Navigate to the Learning Paths section
3. Open a learning path the user has begun
4. Check the progress tracking feature for the learning path

Postconditions:

- User has verified the progress tracking feature for a particular learning path

Expected Result: The platform should accurately show the user their current progress in a learning path

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC04_LP_04

Test Case Title: Verify that instructors can customize learning paths for different needs

Test Case Description: Testing the feature allowing instructors to create and modify learning paths based on specific student or course needs.

Test Suite: Learning Paths and Curriculum

Test Priority: Medium

Preconditions:

- Valid instructor account
- Instructor logged into the online learning platform

Test Data: No test data needed

Test Steps:

1. Login to the online learning platform with an instructor account
2. Create or select a previously created learning path
3. Attempt to customize the learning path's content and structure
4. Save changes and preview the updated learning path

Postconditions:

- Instructor has customized a learning path

Expected Result: Instructors should be able to customize the structure and content of learning paths seamlessly

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC05_LP_05

Test Case Title: Verify that notifications are sent for upcoming courses or milestones in the learning path

Test Case Description: Testing the notification system within the learning paths feature to ensure that learners receive prompt notifications regarding upcoming courses or milestones.

Test Suite: Learning Paths and Curriculum

Test Priority: Medium

Preconditions:

- Valid user account
- User logged into the online learning platform
- User has opted in for notifications

Test Data: Learning path with scheduled upcoming course/module

Test Steps:

1. Login to the online learning platform
2. Navigate to the Learning Paths section
3. Select a learning path with an upcoming course or milestone
4. Wait for a notification of the upcoming course or milestone to arrive

Postconditions:

- User has received or expected a notification

Expected Result: Learners should receive timely notifications for upcoming courses or milestones in their learning paths

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 68

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Attendance Tracking

Feature description -

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner

engagement, identify areas for improvement, and provide targeted support and intervention as needed.

Userstory title -

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Acceptance criteria -

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

Generated Test Cases:

Test Case 1

Test Case ID: TCO01

Test Case Title: Verify that attendance is automatically recorded during live sessions.

Test Case Description: This test case is designed to validate if learners' attendance is recorded automatically during live sessions.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- The system is online.
- The instructor initiates a live session.
- Learners join the live session.

Test Data: User credentials for instructor and learner.

Test Steps:

1. Instructor starts a live session.
2. Learners log into the platform and join the live session.
3. Monitor if the system automatically starts tracking attendance.

Postconditions:

- Data logged for each participant upon joining the live session.

Expected Result: Learner's attendance is tracked automatically when they join a live session.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, End-to-End Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TCO02

Test Case Title: Verify that instructors can view attendance reports for each session.

Test Case Description: This test case is made to check if instructors can effectively access and view attendance reports after each live session.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- A live session has been conducted.
- Attendance has been tracked and recorded.

Test Data: Attendance data of learners from the live session.

Test Steps:

1. Instructor logs into the platform after the live session.
2. Instructor navigates to the attendance report section.
3. Verify if the instructor can access and view detailed attendance data.

Postconditions:

- Attendance report data is accessible for the instructor.

Expected Result: Instructors can access and view attendance reports accurately.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TCO03

Test Case Title: Verify that learners can see their attendance records.

Test Case Description: Ensuring that learners can see their own attendance records to monitor their performance and participation.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Learners have attended live sessions.
- Attendance has been tracked and recorded.

Test Data: Learner's attendance data.

Test Steps:

1. Learner logs into the platform.
2. Learner navigates to their personal attendance record.
3. Check if the learner can access and view their attendance.

Postconditions:

- Attendance data is accessible to the learner.

Expected Result: Learners can view their attendance records accurately.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TCO04

Test Case Title: Verify that attendance data is integrated with participation metrics.

Test Case Description: This test case checks if the attendance data is integrated with other participation metrics to ensure comprehensive learner performance evaluation.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Attendance and participation metrics data have been recorded.

Test Data: Attendance & participation metrics data.

Test Steps:

1. Log into the platform as an instructor.
2. Navigate to the attendance and participation section.
3. Verify if the attendance data is integrated with participation metrics.

Postconditions:

- Attendance and participation data integration confirmed.

Expected Result: Attendance data is accurately integrated with participation metrics.

Severity: Major

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TCO05

Test Case Title: Verify that notifications are sent for missed sessions.

Test Case Description: This test case checks if the system automatically sends notifications to learners and instructors for missed live sessions.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- A live session has been scheduled.
- Learner did not attend the live session.

Test Data: Session ID, Learner's User ID.

Test Steps:

1. A live session is conducted.
2. Check if the system records the absence of a learner.
3. Verify if notifications are sent to the learner and instructor about the missed session.

Postconditions:

- Notifications are sent for the missed session.

Expected Result: For missed sessions, notifications are sent successfully to both the learner and instructor.

Severity: Major

Priority: High

Type of Testing: Functional Testing, End-to-End 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:

Test Case 1

Test Case ID: T01

Test Case Title: Verify that content licenses are assigned to each resource

Test Case Description: This test case is designed to check if every uploaded resource is automatically assigned a content license.

Test Suite: Licensing Test Suite

Test Priority: High

Preconditions:

- User must have content creator privileges
- Content to be uploaded is available

Test Data: Sample educational material

Test Steps:

1. Log in as a content creator
2. Navigate to the course creation section and start uploading a resource
3. Once the upload is complete, view the details of the resource

Postconditions:

- The uploaded resource is available in the creator's dashboard
- A viewing of resource details is reflected in the system logs

Expected Result: The details of the uploaded resource should display a valid content license.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: T02

Test Case Title: Verify that usage rights and permissions are clearly defined.

Test Case Description: This test is designed to ensure that the right usage permissions and rights are set up and visible.

Test Suite: Rights Management

Test Priority: High

Preconditions:

- User is logged in and has content creator rights
- Content is available for manipulation

Test Data: Existing course content

Test Steps:

1. Log in as a content creator
2. Navigate to a course and its contents
3. View the rights and permissions of a resource

Postconditions:

- Course content accessed, and permissions viewed
- All activities are logged

Expected Result: For each content resource, clear rights and permissions should be visible and understandable.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: T03

Test Case Title: Verify that content creators can set restrictions on usage and distribution.

Test Case Description: This test case checks if content creators can restrict resource usage and distribution.

Test Suite: Rights Management

Test Priority: High

Preconditions:

- Logged in as a content creator
- Content available for alterations

Test Data: Existing course content

Test Steps:

1. Log in as a content creator
2. Navigate to a course and the content
3. Adjust and save changes to the usage restrictions

Postconditions:

- Course content adjusted with new restrictions
- Changes recorded in the system

Expected Result: Changes made by the content creator are saved and reflected on the resource.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: T04

Test Case Title: Verify that instructors and learners are informed of content licensing terms.

Test Case Description: This test case is intended to check if licensing terms are displayed to both instructors and learners when they access course content.

Test Suite: License Notification Test Suite

Test Priority: High

Preconditions:

- User is logged in
- Course content is available

Test Data: No test data needed

Test Steps:

1. Log in as a learner or instructor
2. Access course and select a content resource
3. Check for license information

Postconditions:

- Access of content by user is logged

Expected Result: License information should be visible when accessing the content.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: T05

Test Case Title: Verify that notifications are sent for license expiration or renewal.

Test Case Description: This test case is designed to ensure notifications are sent properly when a content license is about to expire or needs renewal.

Test Suite: License Notification Test Suite

Test Priority: High

Preconditions:

- User logged in with content creator rights
- Content with license due for expiration/renewal is available

Test Data: Content with approaching license expiry

Test Steps:

1. Log in as a content creator
2. Navigate to content resources list
3. Wait for notification

Postconditions:

- System has checked for license expiry dates
- System has sent out notifications

Expected Result: The content creator should receive a clear notification about the approaching license expiration/renewal.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 70

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Versioning

Feature description -

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

Userstory title -

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

Acceptance criteria -

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.', 'Notifications are sent for significant content updates.']

Generated Test Cases:

Test Case 1

Test Case ID: TC001

Test Case Title: Verify that instructors can create and save multiple versions of content.

Test Case Description: This test case verifies the ability of instructors to create and save different revisions of course materials without overwriting the previous version.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor account is active
- Course material is available for editing

Test Data: Text document for course material

Test Steps:

1. Log into Instructor Account
2. Navigate to Course
3. Open course content
4. Edit the course material
5. Save course as a new version

Postconditions:

-The updated version should be saved separately

-Versioning history updated

Expected Result: The updated course material version is saved separately without erasing the original version.

Versioning history shows the new version.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC002

Test Case Title: Verify that the history of versions is correctly maintained and accessible.

Test Case Description: This test verifies the system's ability to correctly maintain the history of content versions and allows the instructors to access them at any time.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor account is active
- Version history exist

Test Data: No test data needed

Test Steps:

1. Log into Instructor Account
2. Navigate to Course
3. Open content version history
4. Browse through different versions.

Postconditions:

- Version history display updated.

Expected Result: All the versions of the course material are shown in the version history and can be accessed by the instructor.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC003

Test Case Title: Verify that instructors can compare different versions of course content.

Test Case Description: This test is to confirm that instructors can compare various versions of course content side by side and highlight the differences.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor account is active
- At least two versions of course content exist

Test Data: No test data needed

Test Steps:

1. Log into Instructor account
2. Navigate to Course
3. Open version history
4. Select two versions for comparison

Postconditions:

-Differences between the two versions highlighted.

Expected Result: The system should highlight the discrepancies between the versions of the course content selected.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC004

Test Case Title: Verify that instructors can restore to a previous version of course material.

Test Case Description: This test confirms that if required, instructors can revert back to an older version of course material.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor account is active
- History has at least one previous version of course material

Test Data: No test data needed

Test Steps:

1. Log into Instructor account
2. Navigate to Course
3. Open version history
4. Select the previous version
5. Click on restore

Postconditions:

- Course material reverts to the chosen version.

Expected Result: The selected previous version of course material replaces the current version.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC005

Test Case Title: Verify that notifications are sent for significant content updates.

Test Case Description: This test case validates that the system sends out notifications to the appropriate recipients whenever there's a significant update in the course content.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- Instructor account is active
- Recent significant update in course content

Test Data: No test data needed

Test Steps:

1. Log into Instructor account
2. Create a significant update in the course content
3. Save as a new version

4. Check the mail for notification

Postconditions:

-Notification sent successfully.

Expected Result: Instructors receive a notification confirming the update in course content.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 71

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

API Integration

Feature description -

The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

Userstory title -

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

Acceptance criteria -

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

Generated Test Cases:

Test Case 1

Test Case ID: API-01

Test Case Title: Verify that the APIs are available for key platform functionalities

Test Case Description: This test case is to confirm that the APIs for every key functionality such as course enrollment, progress tracking, taking quizzes and completion certifications are available.

Test Suite: API Integration Suite

Test Priority: High

Preconditions:

- * The platform is fully developed and functional
- * The API endpoints have been clearly defined
- * Authentication is set up and functioning

Test Data: No test data needed

Test Steps:

1. Review API documentation
2. Identify key functionalities of the platform

3. Confirm that APIs are provided for each functionality

Postconditions:

- * Create a report of any missing APIs
- * Forward the report to the development team

Expected Result: All key functionalities have corresponding APIs

Severity: Major

Priority: High

Type of Testing: API Testing, Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: API-02

Test Case Title: Verify that comprehensive API documentation is provided

Test Case Description: This test case is to validate that the API documentation clearly explains each API endpoint, its usage, parameters, response format, error codes and examples.

Test Suite: API Integration Suite

Test Priority: High

Preconditions:

- * The API documentation is complete
- * All API endpoints are correctly defined

Test Data: API documentation

Test Steps:

1. Review API documentation
2. Confirm that all required elements are explained in detail
3. Check for examples and proper formatting

Postconditions:

- * Adequate API documentation

Expected Result: The API documentation is comprehensive and easy to understand

Severity: Critical

Priority: High

Type of Testing: Documentation Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: API-03

Test Case Title: Verify that secure access protocols are implemented and working as expected

Test Case Description: This test assesses if secure access protocols are in place to prevent unauthorized access to APIs.

Test Suite: API Integration Suite

Test Priority: High

Preconditions:

- * APIs are developed and functional
- * Secure access protocols are implemented

Test Data: API token credentials

Test Steps:

1. Investigate secure access protocol implementation
2. Authenticate with token credentials
3. Attempt unauthorized API access

Postconditions:

- * If unauthorized API access is possible, report a critical security issue

Expected Result: Unauthorized API access should be denied with appropriate error code

Severity: Blocker

Priority: High

Type of Testing: Security Testing, API Testing

Test Case Approach: Negative

Test Case 4

Test Case ID: API-04

Test Case Title: Verify that third-party applications can authenticate and access APIs

Test Case Description: This test checks if third-party applications can successfully connect to APIs with valid credentials.

Test Suite: API Integration Suite

Test Priority: Medium

Preconditions:

- * A third-party application with the necessary credentials is available
- * APIs are fully functional

Test Data: Third-party application credentials

Test Steps:

1. Enter the credentials into the third-party application
2. Attempt to connect to the API from the application
3. Check response

Postconditions:

- * Report any connectivity or authentication issues

Expected Result: Third-party applications should be able to authenticate and access APIs successfully.

Severity: Major

Priority: Medium

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: API-05

Test Case Title: Verify that rate limiting and monitoring features are in place

Test Case Description: This test case confirms that rate limiting and monitoring are implemented to prevent excessive requests and track API usage.

Test Suite: API Integration Suite

Test Priority: Low

Preconditions:

- * APIs are functional and access is correctly set up
- * Rate limiting and monitoring settings are configured

Test Data: No test data needed

Test Steps:

1. Send a number of API requests above the rate limit
2. Observe the response received
3. Check if API usage is recorded

Postconditions:

- * Report any deviations from expected behavior

Expected Result: Request over limit should be denied and usage should be recorded

Severity: Minor

Priority: Low

Type of Testing: Load Testing, API Testing

Test Case Approach: Negative

Data - 72

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

Test Case 1:

Test Case ID: 01

Test Case Title: Verify that the data visualization tools are available for analysis.

Test Case Description: This test case checks if the data visualization tools, including charts, graphs, heat maps, etc. are available for viewing and analysis.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- The tester has admin access to the platform
- The platform has data to visualize

Test Data: No test data needed

Test Steps:

1. Login to the platform with admin credentials
2. Go to the data analytics and insights section.
3. Check the availability of data visualization tools

Postconditions:

- Log out from the admin account

Expected Result: The data visualization tools are properly available and functional.

Severity: High

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: 02

Test Case Title: Verify that reports can be generated for learner performance metrics.

Test Case Description: This test case verifies the ability to generate reports based on different learner performance metrics.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- The tester has admin access
- There are learner profiles with performance metrics available

Test Data: Prepopulated learner performance metrics

Test Steps:

1. Login to the platform with admin credentials
2. Navigate to the analytics section
3. Select the option to generate a report based on learner performance
4. Review the generated report

Postconditions:

- Log out after reviewing the report

Expected Result: The platform properly generates and displays learner performance reports.

Severity: High

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: 03

Test Case Title: Verify that real-time analytics provide insights into engagement and outcomes.

Test Case Description: This test case checks if real-time analytics provide insights into user engagement and outcomes.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- The tester has admin access
- There are ongoing learner activities on the platform

Test Data: Real-time user interaction data

Test Steps:

1. Login to the platform with admin credentials

2. Go to the analytics section
3. Observe the real-time analytics dashboard
4. Check if insights are displayed for learner engagement and outcomes

Postconditions:

- Log out after reviewing

Expected Result: Real-time analytics accurately display insights into learner engagement and outcomes.

Severity: Major

Priority: High

Type of Testing: Non-Functional Testing, Performance Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: 04

Test Case Title: Verify that data can be exported for further analysis.

Test Case Description: This test case verifies the platform's capability to export data.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- The tester has admin access
- There are performance metrics available to export

Test Data: No test data needed

Test Steps:

1. Login to the platform with admin credentials
2. Go to the analytics and insights section
3. Select a set of data to be exported
4. Initiate the data-export process

Postconditions:

- Successfully downloaded data export file
- Log out from the platform

Expected Result: Data is successfully exported for further analysis.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: 05

Test Case Title: Verify that notifications are sent for significant data trends or anomalies.

Test Case Description: This test case aims to check if notifications are correctly sent when significant data trends or anomalies are detected.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- The tester has admin access
- An anomaly or significant trend is present in the data

Test Data: Prepopulated data with anomalies

Test Steps:

1. Login to the platform with admin credentials
2. Generate an analytic report observed for anomalies or significant trends
3. Check if a notification is raised

Postconditions:

- Received notification regarding the anomaly

- Log out from the system

Expected Result: Notifications for anomalies or significant trends in data are sent promptly.

Severity: Major

Priority: Medium

Type of Testing: Non-Functional Testing, Performance 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:

Test Case 1:

Test Case ID: TC_01

Test Case Title: Verify that administrators can create and assign user roles successfully

Test Case Description: This test case verifies whether the administrator can create and assign new user roles successfully.

Test Suite: User Roles and Permissions Module

Test Priority: High

Preconditions:

- The admin account must be logged in
- The required fields for creating user roles must be defined

Test Data: Admin credentials, User role details (Role Name, Description, Permissions, etc.)

Test Steps:

1. Log in as an Administrator
2. Navigate to the 'User Roles' section
3. Click on 'Create a new role'
4. Enter all the required fields and set the permissions
5. Click 'Save'
6. Assign the new role to a specific user

Postconditions:

- New role is added to the 'User Roles' section
- A confirmation message is shown for successful role assignment

Expected Result: The System should allow the admin to create a new user role and assign it successfully.

Severity: Critical

Priority: High

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

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_02

Test Case Title: Verify that each role's permissions are customizable

Test Case Description: This test case validates whether permissions can be customized successfully for each role.

Test Suite: User Roles and Permissions Module

Test Priority: High

Preconditions:

- Admin account must be logged in
- A user role to be customized must exist

Test Data: Admin credentials, Permission settings.

Test Steps:

1. Log in as an Administrator
2. Select a user role from the 'User Roles' section
3. Click 'Edit Permissions'
4. Change the current permissions and click 'Save'

Postconditions:

- Permissions are changed for the selected role
- System displays a confirmation message

Expected Result: The system should allow customizing permissions of a selected role successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing, System Testing, Regression Testing

Test Case Approach: Positive

(And so forth for rest of the test cases...)

Data - 74

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Feedback and Rating System

Feature description -

The platform includes a user feedback and rating system, allowing learners to provide feedback on courses, instructors, and learning materials. This feature enables continuous improvement by gathering user insights and ratings, helping instructors refine their content and approach.

Userstory title -

As a learner, I want to provide feedback and ratings so that I can share my experience and help improve the platform.

Acceptance criteria -

['Learners can rate courses and instructors on a scale.', 'Feedback forms are available for detailed comments.',

'Instructors can view and respond to feedback.', 'Aggregated ratings are displayed publicly.', 'Anonymous feedback options are available.']

Generated Test Cases:

Test Case 1

Test Case ID: 001

Test Case Title: Verify that learners can rate courses and instructors on a scale.

Test Case Description: This test case ensures that the platform allows learners to rate both courses and instructors using a defined scale.

Test Suite: User Rating System Suite

Test Priority: High

Preconditions:

- User is registered and is logged into the platform.
- User has completed at least one course.

Test Data: User login details, Course completion details

Test Steps:

1. Login using user credentials.
2. Go to completed courses.
3. Select a course and click on 'Give Rating'.
4. Rate the course and instructor on a scale.
5. Submit the ratings.

Postconditions:

- The user is returned to the course page.
- A confirmation message appears indicating successful submission of ratings.

Expected Result: Users can successfully rate both courses and instructors on a defined scale.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: 002

Test Case Title: Verify that detailed feedback forms are available for user comments.

Test Case Description: The test case validates the availability and functioning of feedback forms for users to leave detailed comments.

Test Suite: User Review and Feedback Suite

Test Priority: High

Preconditions:

- User is registered and logged into the platform.
- User has completed at least one course.

Test Data: User login details, Course completion details

Test Steps:

1. Login using user credentials.
2. Head to completed courses.
3. Select a course and click on 'Leave Feedback'.
4. Fill out the feedback form with detailed comments.
5. Click on 'Submit Feedback'.

Postconditions:

- User is returned to the course page.
- A 'Thank you' message is displayed acknowledging the feedback submission.

Expected Result: Users can successfully use detailed feedback forms to leave comments.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: 003

Test Case Title: Verify that instructors can view and respond to feedback.

Test Case Description: This test case ensures that instructors have the ability to view learner feedback and respond to it appropriately.

Test Suite: Feedback Management for Instructors Suite.

Test Priority: High

Preconditions:

- Instructor is registered and is logged into the platform.
- At least one feedback has been submitted by a user to one of the instructor's courses.

Test Data: Instructor login details.

Test Steps:

1. Login as instructor.
2. Go to 'My Courses'.
3. Select a course and click 'Feedback'.
4. View feedback comments.
5. Respond to the comments.

Postconditions:

- The response is recorded and published under the specific feedback.

Expected Result: Instructors can successfully view and respond to feedback.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: 004

Test Case Title: Verify that aggregated ratings are displayed publicly.

Test Case Description: This test case checks if the combined ratings of courses or instructors are publicly displayed on the platform.

Test Suite: User Rating System Suite

Test Priority: Medium

Preconditions: Ratings for courses or instructors exist.

Test Data: No test data needed.

Test Steps:

1. Go to the course page or instructor profile.
2. Check the display of the aggregated ratings.

Postconditions: Further browsing of the platform.

Expected Result: The aggregated ratings are publicly displayed.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: 005

Test Case Title: Verify that anonymous feedback options are available.

Test Case Description: This test case validates the ability for users to submit feedback anonymously.

Test Suite: User Review and Feedback Suite

Test Priority: Medium

Preconditions:

- User is registered and is logged into the platform.
- User has completed at least one course.

Test Data: User login details, Course completion details.

Test Steps:

1. Login using user credentials.
2. Navigate to completed courses.
3. Select a course and click on 'Leave Feedback'.
4. Opt for 'Submit Anonymously' option.
5. Submit the feedback.

Postconditions:

- User is returned to the course page.
- A 'Thank you' message is displayed acknowledging the anonymous feedback submission.

Expected Result: Users can successfully leave anonymous feedback.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 75

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Media Integration

Feature description -

The platform supports social media integration, allowing learners and instructors to share their achievements, courses, and updates on social media platforms. This feature promotes engagement and community building, enhancing the visibility of the platform's offerings.

Userstory title -

As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

Acceptance criteria -

['Learners can connect their social media accounts to the platform.', 'Achievements and course completions can be shared on social media.', 'Privacy settings control the visibility of shared content.', 'Instructors can promote courses and updates through social media.', 'Analytics track the reach and engagement of shared content.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_OLP_001

Test Case Title: Verify that learners can connect their social media accounts to the platform.

Test Case Description: This test case checks if learners are able to connect their social media profiles with the online learning platform.

Test Suite: Social Media Integration

Test Priority: High

Preconditions: The learner has registered and logged in to their account. The learner has an active social media account.

Test Data: Learner's social media login credentials.

Test Steps:

1. Navigate to the profile settings page.
2. Click on the 'Connect Social Media' button.
3. Enter login credentials for a social media account.
4. Click on 'Authorize?.'

Postconditions: Refresh the profile settings page.

Expected Result: The connected social media account should be displayed under the social media connections in profile settings.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_OLP_002

Test Case Title: Verify that achievements and course completions can be shared on social media.

Test Case Description: This test case verifies the functionality of sharing achievements and course completions on connected social media platforms.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

1. The learner has connected a social media account.

2. The learner has completed a course or has a recent achievement.

Test Data: No test data needed

Test Steps:

1. Go to the 'Achievements' section of the learner's profile.
2. Click on the 'Share' button next to a listed achievement or course.
3. Select the social media platform to share on.
4. Click 'Confirm'.

Postconditions: Navigate to the connected social media's profile page.

Expected Result: The shared achievement should be visible in the user's social media profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_OLP_003

Test Case Title: Verify that privacy settings control the visibility of shared content.

Test Case Description: This test case ensures that the platform respects the privacy of learners by enabling them to control who can see their shared content.

Test Suite: Social Media Integration

Test Priority: High

Preconditions: The learner has shared an achievement on social media.

Test Data: No test data needed.

Test Steps:

1. Navigate to the shared post on the social media platform.
2. Adjust the visibility settings of the post.

3. Confirm changes.

Postconditions: Log out from the social media platform and view the profile as a visitor.

Expected Result: Only people fitting the selected privacy setting should be able to view the shared content.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_OLP_004

Test Case Title: Verify that instructors can promote courses and updates through social media.

Test Case Description: This case checks whether an instructor is able to share course information and updates on their social media profiles to attract potential learners.

Test Suite: Social Media Integration

Test Priority: High

Preconditions: The instructor has a created course and has connected their social media profiles.

Test Data: Course information, instructor's social media credentials.

Test Steps:

1. Go to the 'My Courses' tab.
2. Click on the 'Share' button next to the desired course.
3. Choose the social media platform to share on.
4. Confirm sharing.

Postconditions: Navigate to the instructor's social media profile.

Expected Result: The course should be seen shared on the instructor's social media profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_OLP_005

Test Case Title: Verify that analytics track the reach and engagement of shared content.

Test Case Description: This test case checks the functionality of analytics provided by the online learning platform to track the reach and engagement of shared content on social media.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions: An achievement, update, or course has been shared on a connected social media platform.

Test Data: No test data needed.

Test Steps:

1. Go to the Analytics section of the profile.
2. Select the 'Social Media Engagement' tab.
3. Select the shared post for reviewing analytics.

Postconditions: No specific postcondition.

Expected Result: Analytics data regarding the reach and engagement of the selected post should be displayed.

Severity: Minor

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:

Test Case 1:

Test Case ID: FT-1

Test Case Title: Verify that the app records data on distance, duration, pace, and calories burned

Test Case Description: This test case seeks to ascertain that the app accurately records and displays parameters such as distance, duration, pace, and calories burned during an activity.

Test Suite: Activity Tracking Functionality

Test Priority: High

Preconditions: A user must have an account and a recorded activity.

Test Data: User login credentials and activity tracking data

Test Steps:

1. Login into the app.
2. Register any activity such as walking.
3. End the activity.
4. Review the recorded data for the activity.

Postconditions: The app has captured and displayed the activity parameters.

Expected Result: The app records and displays the distance, duration, pace, and calories burned accurately.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-2

Test Case Title: Verify that users can view activity summaries

Test Case Description: Testing that the app allows users to view summaries of their recorded activities.

Test Suite: Activity Tracking Functionality

Test Priority: High

Preconditions: A user must have an account and recorded activities.

Test Data: User login credentials and activity tracking data

Test Steps:

1. Login into the app.

2. Navigate to the Activity Summary section.
3. Review the summaries of recorded activities.

Postconditions: The user has viewed the summaries of their recorded activities.

Expected Result: Summaries for all recorded activities are visible and accurate.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-3

Test Case Title: Verify that activity tracking can be paused and resumed

Test Case Description: Testing if a user can pause an ongoing activity and subsequently resume it.

Test Suite: Activity Tracking Functionality

Test Priority: High

Preconditions: A user must have an account and an ongoing activity.

Test Data: User login credentials

Test Steps:

1. Login into the app.
2. Start any activity.
3. Pause the activity.
4. Resume the activity.

Postconditions: An ongoing activity was paused and then resumed.

Expected Result: The app should successfully pause and resume activity tracking without any loss or corruption of data.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-4

Test Case Title: Verify that users receive notifications for milestones

Test Case Description: This test verifies if the app sends notifications for achieved milestones.

Test Suite: Activity Tracking Functionality

Test Priority: Medium

Preconditions: The user must have an account, and the notification settings should be enabled.

Test Data: User login credentials and activity tracking data that achieves a milestone.

Test Steps:

1. Login into the app.
2. Record an activity that achieves a milestone.
3. Assess the notifications received.

Postconditions: The user receives a notification for a achieved milestone.

Expected Result: The app sends a notification when a milestone is achieved.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-5

Test Case Title: Verify that data is stored for historical analysis

Test Case Description: Testing if the app stores data for future reference and analysis.

Test Suite: Activity Tracking Functionality

Test Priority: High

Preconditions: A user must have an account and recorded activities.

Test Data: User login credentials and activity tracking data

Test Steps:

1. Login into the app.
2. Navigate to the historical data section.
3. Review the recorded activities.

Postconditions: The user has accessed and reviewed the historical data.

Expected Result: The app stores and retrieves user's historical activity data accurately.

Severity: Major

Priority: High

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:

Test Case 1:

Test Case ID: FT001

Test Case Title: Verify that the GPS Tracking accurately records route data.

Test Case Description: Test aims at validating the route data precision acquired through GPS tracking during an outdoor activity.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- The user is logged into the app.
- GPS is enabled on the device.
- The user is outdoors.

Test Data: GPS coordinates

Test Steps:

1. Log into the Fitness Tracker App.
2. Start any outdoor activity (e.g. running).
3. Follow the chosen route while the app tracks the movements.
4. After completing the route, stop the activity.
5. Verify the mapped route details displayed on the app.

Postconditions:

- Route data is available for review in the app.
- Activity is saved in the user's history.

Expected Result: Route details logged from GPS should match the actual route precisely.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT002

Test Case Title: Verify that users can view detailed maps and elevation profiles.

Test Case Description: Test aims at reassuring the user's ability to view detailed maps with elevation profiles during or after a tracked activity.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- A completed outdoor activity is available in the user's history.
- A graphical user interface (GUI) is properly working.

Test Data: Pre-recorded GPS coordinates

Test Steps:

1. Log into the Fitness Tracker App.
2. Open the history of outdoor activities.
3. Choose a completed outdoor activity.
4. Evaluate the map and elevation profile details.

Postconditions:

- The elevation profile and map have been observed by the user.

Expected Result: Users should be able to see detailed maps with elevation profiles for each recorded activity.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT003

Test Case Title: Verify that performance data includes location-based metrics.

Test Case Description: The test verifies that the app accurately provides location-based performance metrics after a completed outdoor activity.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- User has completed an outdoor activity with GPS tracking.

Test Data: Pre-recorded GPS coordinate and activity details.

Test Steps:

1. Log into the Fitness Tracker App.
2. Open the history of outdoor activities.

3. Select a previously completed activity.

4. Inspect the performance metrics.

Postconditions:

- Performance metrics have been observed by the user.

Expected Result: Location-based performance metrics should be available and accurate for each recorded activity.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT004

Test Case Title: Verify that routes can be saved and revisited.

Test Case Description: The test confirms that tracked outdoor activities routes can be saved, revisited, and reused.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- User has completed an outdoor activity with GPS tracking.

Test Data: Pre-recorded GPS coordinate and activity details.

Test Steps:

1. Log into the Fitness Tracker App.
2. Open the history of outdoor activities.
3. Select a previously completed activity.
4. Use the saved route for a new activity.

Postconditions:

- The saved route is available for a new activity.

Expected Result: All saved routes should be retrievable and reusable for new activities.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT005

Test Case Title: Verify that notifications are provided for deviations from planned routes.

Test Case Description: Test aims at verifying if the app provides notifications to the user when they deviate from the planned route during an outdoor activity.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- User plans a route for an outdoor activity.
- GPS is enabled on the device.
- The app permission for notifications is enabled.
- The user is outdoors.

Test Data: Planned route GPS coordinates

Test Steps:

1. Log into the Fitness Tracker App.
2. Plan a route for an outdoor activity.
3. Start the activity.
4. Deviate from the planned route.

5. Wait for the app to recognize the deviation.

6. Confirm the notification.

Postconditions:

- The user continues with the activity after receiving the notification.

Expected Result: Users should receive timely notifications for deviations from the planned route.

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:

Test Case 1:

Test Case ID: HRM_01

Test Case Title: Verify that heart rate data is recorded in real-time.

Test Case Description: This test case checks whether the app is effectively tracking and displaying the user's heart rate in real time.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is signed into the app.
- User has compatible wearable device that is turned on and connected to the app.

Test Data: Real-time heart rate data from wearable device

Test Steps:

1. User starts a workout.
2. User monitors heart rate on the app as they perform workout.
3. User checks if the heart rate displayed correlates with their workout intensity.

Postconditions:

- Workout session ends.
- Heart rate data is saved in the app.

Expected Result: Heart rate data is recorded and displayed in real-time on the app.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: HRM_02

Test Case Title: Verify that users can set and monitor target heart rate zones.

Test Case Description: This test case verifies the ability of the user to set and monitor goals in relation to their heart rate zones.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

- User is signed into the app.
- User has accurate knowledge of their personalized heart rate zones.

Test Data: Target heart rate zones data

Test Steps:

1. User attempts to set target heart rate zones in the app
2. User starts a workout.
3. User monitors if their actual heart rate stays within the target zones.

Postconditions:

- Workout session ends.
- Achievement of heart rate goals are recorded.

Expected Result: Users can successfully set and monitor their target heart rate zones during a workout.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: HRM_03

Test Case Title: Verify that historical heart rate data can be viewed.

Test Case Description: This test case checks the functionality of the heart rate history storing capability and ease of access.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

- User is signed into the app.
- User has previously completed workouts with heart rate data.

Test Data: Historical heart rate data

Test Steps:

1. User navigates to the heart rate history section in the app.
2. User checks for previous workout data.

Postconditions: N/A

Expected Result: Historical heart rate data from past workouts is accurately stored and accessible.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: HRM_04

Test Case Title: Verify that alerts are provided for abnormal heart rate readings.

Test Case Description: This test case aims to validate the app's ability to provide alert notifications whenever the user's heart rate exceeds or drops below healthy limits.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is signed in to the app.
- User has their wearable device connected and turned on.

Test Data: Heart rate data

Test Steps:

1. User starts a simulated task that deliberately provides abnormal heart rate readings exceeding safe limits.
2. User checks if the app sends an alert notification.

Postconditions:

- Simulated task ends.
- Notification is dismissed.

Expected Result: Alerts are successfully sent to the user whenever there are abnormal heart rate readings.

Severity: Blocker

Priority: High

Type of Testing: Non-Functional Testing

Test Case Approach: Negative

Test Case 5:

Test Case ID: HRM_05

Test Case Title: Verify that the app is compatible with various wearable devices.

Test Case Description: This test case checks the app's capability to integrate with different brand's wearable devices.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is signed into the app.
- User has different brands of wearable devices available for testing.

Test Data: Wearable device compatibility data

Test Steps:

1. User connects each wearable device to the app.
2. User monitors heart rate using the app, while wearing each device.

Postconditions:

- Heart rate monitoring ends.
- Heart rate data saved in the app.

Expected Result: The app successfully connects and retrieves data from all tested wearable devices.

Severity: Major

Priority: High

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Data - 79

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Calorie Tracking

Feature description -

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

Userstory title -

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

Acceptance criteria -

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC001

Test Case Title: Verify that users can log their daily food intake and calorie consumption.

Test Case Description: This test case ensures that users are able to log their daily food intake and the corresponding calorie count.

Test Suite: Calorie Tracking Suite

Test Priority: High

Preconditions:

- User is registered and logged into the app.
- User has food consumption for the day.

Test Data: Food Name: Salad, Calories: 300 cal

Test Steps:

1. Log in to the Fitness Tracker App.
2. Click on the "Log Food" button.
3. Enter "Salad".
4. Input "300" for Calories.
5. Click "Save."

Postconditions:

- Food and calorie data has been entered.
- System updates calorie consumption for the day.

Expected Result: Calorie log is updated with "Salad, 300 cal" and daily calorie consumption total includes these 300 calories.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that the app calculates total calories consumed and burned.

Test Case Description: This test case validates that the app accurately calculates and displays the total calories consumed and burned by the user.

Test Suite: Calorie Tracking Suite

Test Priority: High

Preconditions:

- User has consumed food and logged it.
- User has performed an exercise and burned calories.

Test Data: Food: Salad, Calories Consumed: 300 cal, Exercise Name: Running, Calories Burned: 200 cal.

Test Steps:

1. Log in to the Fitness Tracker App.
2. Input the food consumption (Salad, 300 cal).
3. Log exercise (Running).
4. Input calories burned (200 cal).
5. Review the dashboard for total calories consumed and burned.

Postconditions:

- Food intake and exercise have been logged.
- Calculations for total calories consumed and burned are displayed.

Expected Result: Total calories consumed are shown as 300 cal and total calories burned are shown as 200 cal.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003

Test Case Title: Verify that users can set daily calorie goals and track their progress.

Test Case Description: This test case ensures users can input their daily calorie goal and view their progress towards it.

Test Suite: Calorie Tracking Suite

Test Priority: High

Preconditions:

- User is registered and logged into the app.
- User has a caloric goal for the day.

Test Data: Daily Caloric Goal: 2000 cal

Test Steps:

1. Log in to the Fitness Tracker App.
2. Click on "Set daily caloric goal."
3. Input "2000" for the daily caloric goal.
4. Save the changes.
5. Review the calorie goal status.

Postconditions:

- Daily caloric goal has been set.
- User can view their progress towards the goal.

Expected Result: The daily calorie goal is set to 2000 cal. The progress bar indicates the percentage achieved based-on consumption logs.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

(Add more test cases as required to cover the rest of the acceptance criteria)

Data - 80

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

Test Case 1:

Test Case ID: FTC_01

Test Case Title: Verify that the app records sleep data including duration and quality.

Test Case Description: This test case aims to verify if the app is accurately tracking and recording user sleep data, especially in terms of sleep duration and quality.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions: User is signed in, and the app is linked to a wearable device.

Test Data: Sleep data input by user.

Test Steps: 1) Ensure that the wearable device is on and connected to the app. 2) User goes to sleep while wearing the device. 3) After the user wakes up, check the sleep data on the app.

Postconditions: Refresh the app to update sleep data.

Expected Result: The app shows accurate sleep data including the duration and quality of sleep.

Severity: Major

Priority: High

Type of Testing: Functional testing, Integration testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTC_02

Test Case Title: Verify that users can view detailed sleep stages and patterns.

Test Case Description: This test case verifies if users can access and view detailed breakdowns of their sleep stages and patterns, helping to identify any irregularities.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions: Sleep data has been collected by the app.

Test Data: No test data needed.

Test Steps: 1) Open the sleep data dashboard. 2) Navigate to the detailed sleep stages/ patterns section. 3) Check the information presented.

Postconditions: Logout of the app.

Expected Result: The user can view distinct sleep stages and patterns.

Severity: Major

Priority: High

Type of Testing: Usability testing, Functional testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTC_03

Test Case Title: Verify that the app provides insights into sleep disturbances.

Test Case Description: This test case aims to confirm if the app effectively analyzes and provides insights into the user's sleep disturbances.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions: Sleep data has been captured that indicates sleep disturbances.

Test Data: Sleep data indicating disturbances.

Test Steps: 1) Open the app. 2) Navigate to sleep insights. 3) Review the insights corresponding to sleep disturbances.

Postconditions: No direct actions necessary.

Expected Result: App identifies and provides insights into observed sleep disturbances.

Severity: Minor

Priority: Medium

Type of Testing: Functional testing, Data-Driven Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTC_04

Test Case Title: Verify that users receive recommendations for improving sleep quality.

Test Case Description: This test case checks whether the app provides valid and useful recommendations to the user based on their sleep data, aimed at improving sleep quality.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions: Sleep data has been collected and analyzed by the app.

Test Data: No test data needed.

Test Steps: 1) Open the app. 2) Navigate to health recommendations. 3) View the recommendations for improving sleep quality.

Postconditions: No direct actions necessary.

Expected Result: App provides practical recommendations for how the user can improve their sleep quality.

Severity: Minor

Priority: Medium

Type of Testing: Functional testing, Usability testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTC_05

Test Case Title: Verify that sleep data can be viewed in daily, weekly, and monthly summaries.

Test Case Description: This test case checks if users can view summaries of their sleep data at daily, weekly, and monthly intervals.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions: Sufficient sleep data has been collected over time.

Test Data: No test data needed.

Test Steps: 1) Open the app. 2) Navigate to the sleep summary section. 3) Toggle between the daily, weekly, and monthly view options.

Postconditions: Logout of the app.

Expected Result: Sleep data summaries for daily, weekly, and monthly intervals are displayed correctly.

Severity: Minor

Priority: Medium

Type of Testing: Functional testing, Usability 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:

Test Case 1:

Test Case ID: TC_WL_001

Test Case Title: Verify that users can log various workout types with details.

Test Case Description: This test case is designed to assess the ability of users to input different kinds of workout details such as type, duration, intensity, and additional comments.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User should have an active account on the Fitness Tracker App.
- User is logged in to the account.

Test Data: User inputs for workout type, duration, intensity, and notes.

Test Steps:

1. Login to the Fitness Tracker App.
2. Click on 'New Workout'.
3. Select 'type' from the dropdown.
4. Enter 'duration' and 'intensity'.
5. Input any relevant notes in the 'notes' field.
6. Click on 'Save'.

Postconditions:

- The workout details save successfully.

- User is returned to the homepage.

Expected Result: A new workout log has been successfully created.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_WL_002

Test Case Title: Verify that users can edit and delete workout entries.

Test Case Description: This test case checks the functionality of editing and deleting an existing entry of workouts.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User should have an active account on the Fitness Tracker App.
- User is logged in to the account.
- There exist previous entries of workouts.

Test Data: User inputs to edit workout details.

Test Steps:

1. Login to the Fitness Tracker App.
2. Go to 'Workout History'.
3. Click on an existing workout entry.
4. Edit the details and click on 'Save'.
5. Go back to 'Workout History'.
6. Click on the same workout entry and press 'Delete'.

Postconditions:

- The workout details are updated successfully.
- The workout entry is removed from the system.

Expected Result: The edited details of the workout are observed in the 'Workout History'. After being deleted, the entry is removed and no longer viewable.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_WL_003

Test Case Title: Verify that users can filter logs by date and type of workout.

Test Case Description: This test case verifies the ability for users to filter their workout logs based on the date and workout type.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User should have an active account on the Fitness Tracker App.
- User is logged in to the account.
- There exist previous entries of workouts.

Test Data: Filter parameters (date range, workout type)

Test Steps:

1. Login to the Fitness Tracker App.
2. Go to 'Workout History'.
3. Set filter parameters with specific date range and workout type.

4. Click on 'Apply Filters'.

Postconditions:

- Workout logs are filtered and displayed as per chosen parameters.

Expected Result: Only workout logs meeting the filter parameters are displayed.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_WL_004

Test Case Title: Verify that workout data is stored and viewable for tracking progress.

Test Case Description: This test case tests the visibility of previous workout logs within view of the user for the purpose of tracking regular workout progress.

Test Suite: Workout Progress Tracking

Test Priority: High

Preconditions:

- User should have an active account on the Fitness Tracker App.
- User is logged in to the account.
- There exist previous entries of workouts.

Test Data: No test data needed

Test Steps:

1. Login to the Fitness Tracker App.
2. Go to 'Workout History'.

Postconditions:

- User can view previous workout logs.

Expected Result: User can see saved workout logs in the 'Workout History'.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_WL_005

Test Case Title: Verify that workout input fields reject incorrect data.

Test Case Description: This test case verifies the system's ability to reject incorrect information input by a user.

Test Suite: Workout Data Quality Assurance

Test Priority: High

Preconditions:

- User should have an active account on the Fitness Tracker App.
- User is logged in to the account.

Test Data: Incorrect user inputs for workout type, duration, intensity, and notes.

Test Steps:

1. Login to the Fitness Tracker App.
2. Click on 'New Workout'.
3. Input incorrect data for type, duration, intensity, and notes fields.
4. Click on 'Save'.

Postconditions:

- An error message is displayed.
- The incorrect workout details are not saved.

Expected Result: An error message is displayed and incorrect workout data is rejected.

Severity: Critical

Priority: High

Type of Testing: Functional testing; Negative testing

Test Case Approach: Negative

Data - 82

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Goal Setting

Feature description -

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

Userstory title -

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

Acceptance criteria -

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT001

Test Case Title: Verify that the user is able to set goals for different fitness metrics.

Test Case Description: This test case ensures that the goals' function in the setting section is responsive and users can input different values in the fitness metric goals.

Test Suite: Goal Setting

Test Priority: High

Preconditions: The user is logged in and has access to the setting.

Test Data: Various types of fitness goals such as steps count, weight loss, calorie burn etc.

Test Steps:

1. Login to the Fitness Tracker App.
2. Navigate to the 'goals' section under 'settings'.
3. Enter desired values in the different fitness metric goals.
4. Save the changes.

Postconditions: None

Expected Result: The goals for different fitness metric should be set as per user input.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT002

Test Case Title: Verify that the progress tracking functionality shows accurate data.

Test Case Description: This test case ensures that the user's applied fitness metrics in the goal setting section are reflected in the progress tracking.

Test Suite: Goal Setting

Test Priority: High

Preconditions: The user has set fitness goals.

Test Data: No test data needed

Test Steps:

1. Login to the Fitness Tracker App.
2. Navigate to the 'Progress Tracking' section.
3. Verify if the set goals are accurately reflected.
4. Record some exercise activity and verify if the progress tracking updates as per the activity.

Postconditions:

1. Recording of exercise activity to check progress tracking.

Expected Result: The progress tracking should accurately reflect the set fitness goals and record associated activities.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT003

Test Case Title: Verify that the app sends motivational feedback and reminders.

Test Case Description: This test case checks if the user receives motivational feedback and reminders after achieving or falling behind on their set fitness goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions: The user has set fitness goals.

Test Data: Progress of user in achieving set goals.

Test Steps:

1. Login to the Fitness Tracker App.
2. Achieve or fall behind on some fitness goals.
3. Check for motivational feedback or reminders.

Postconditions: None

Expected Result: The app should send reminders or motivational feedback based on the user's progress.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT004

Test Case Title: Verify that users can adjust or update goals.

Test Case Description: This test case ensures that a user can adjust saved goals based on their changing fitness requirements.

Test Suite: Goal Setting

Test Priority: High

Preconditions: The user has set fitness goals.

Test Data: New fitness goal to be updated.

Test Steps:

1. Login to the Fitness Tracker App.
2. Navigate to the 'goals' section in 'settings'.
3. Adjust one or more fitness goals.
4. Save the changes.

Postconditions: None

Expected Result: The updated fitness goals should save accurately in the app.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT005

Test Case Title: Verify that the app provides a summary of goal attainment.

Test Case Description: This test case checks if the app shows a summary of the user's progress towards their set goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions: The user has set fitness goals.

Test Data: User's progress data.

Test Steps:

1. Login to the Fitness Tracker App.
2. Navigate to the 'Progress Summary' section.

3. Check if the progress summary accurately reflects the user's progress towards their goals.

Postconditions: None

Expected Result: The app should provide a summary of user's goal attainment accurately.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 83

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Plans

Feature description -

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

Userstory title -

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

Acceptance criteria -

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

Generated Test Cases:

Test Case 1

Test Case ID: TC_FIT_001

Test Case Title: Verify that users can select workout plans based on their fitness level.

Test Case Description: This test case is designed to validate that the app provides the functionality for users to choose workout plans that align with their specific fitness levels.

Test Suite: Workout Plans Module

Test Priority: High

Preconditions:

- The user has created a profile and logged in.
- The user has defined their fitness level in their profile.

Test Data: User profile data with defined fitness level

Test Steps:

1. Open the fitness tracker app.
2. Log in to the user account.
3. Navigate to the 'Workout Plans' section.
4. Browse the available workout plans.
5. Select a plan that aligns with the fitness level defined in the user profile.

Postconditions:

- The selected plan should be added to the user's profile.

Expected Result: The app allows users to choose workout plans that align with their fitness level.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_FIT_002

Test Case Title: Verify that the workout plans include detailed exercise routines and schedules.

Test Case Description: The purpose of this test case is to ensure that each workout plan provided by the app includes comprehensive information on exercise routines and schedules.

Test Suite: Workout Plans Module

Test Priority: High

Preconditions:

- The user has logged into the app.
- A workout plan has been selected.

Test Data: Detailed exercise routines and schedules

Test Steps:

1. Log in to the fitness tracker app.
2. Navigate to the 'Workout Plans' section.
3. Select a workout plan.
4. Review the exercise details and routines included in the plan.

Postconditions:

- The exercise routines for the selected workout plan have been displayed and reviewed.

Expected Result: All plans in the app's 'Workout Plans' section provide detailed information on exercises and schedules.

Severity: Minor

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_FIT_003

Test Case Title: Verify that users can track adherence to and progress within the workout plan.

Test Case Description: This test case checks the ability of the users to monitor their compliance with the workout plan and to see how far they have come since starting the plan.

Test Suite: Progress Tracking Module

Test Priority: High

Preconditions:

- The user is logged in.
- The user has a workout plan selected.
- The user has started following the workout plan.

Test Data: Progress data of the user based on the completed workouts

Test Steps:

1. Log in to the app.
2. Select the 'Progress Tracking' module.
3. Check the recorded adherence and progress.

Postconditions:

- The progress tracking report for the user is displayed.

Expected Result: The app provides detailed tracking of the user's adherence and progress within the workout plan.

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:

Test Case 1

Test Case ID: FT01

Test Case Title: Verify that a user can join and participate in community groups.

Test Case Description: The test aims to confirm the functionality of joining and participating in community groups.

Test Suite: Community Support

Test Priority: High

Preconditions: User must have a registered account on the Fitness Tracker App.

Test Data: User credentials, Group name

Test Steps:

1. Navigate to the community tab.
2. Search for a group to join.
3. Request to join the group.
4. Post a message in the group.

Postconditions: Clear the search bar, Log out

Expected Result: User successfully joins a group and posts a message.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: FT02

Test Case Title: Verify that users can share their progress updates and achievements.

Test Case Description: The test aims to confirm the functionality of sharing exercise progress and

achievements.

Test Suite: Community Support

Test Priority: High

Preconditions: User must have completed some fitness activity.

Test Data: Activity data.

Test Steps:

1. Navigate to the Progress tab.
2. Choose an activity to share.
3. Post the activity in the community/group.

Postconditions: Log out.

Expected Result: User successfully posts their progress and achievements in the community/group.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: FT03

Test Case Title: Verify that the app facilitates participation in community challenges.

Test Case Description: The test aims to confirm the functionality of participating in community challenges.

Test Suite: Community Support

Test Priority: Medium

Preconditions: User must be a part of a community/group which hosts challenges.

Test Data: Challenge data.

Test Steps:

1. Navigate to Challenges tab.

2. Select a challenge to participate in.

3. Click the Participate button.

Postconditions: Log out.

Expected Result: Challenge starts, capturing user's related activity data.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: FT04

Test Case Title: Verify that community activity and engagement are tracked.

Test Case Description: The test aims to confirm that the app facilitates tracking community activity and engagement.

Test Suite: Community Support

Test Priority: High

Preconditions: User must be part of a community group and have engaged in group activity.

Test Data: No test data needed.

Test Steps:

1. Navigate to community tab.
2. Select a group.
3. View activity log/engagement tab.

Postconditions: Log out.

Expected Result: A summary of the user's activities within the group is displayed.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: FT05

Test Case Title: Verify that users can offer and receive support and encouragement.

Test Case Description: The test is to confirm that users can post and receive motivation and encouragement from fellow members.

Test Suite: Community Support

Test Priority: Medium

Preconditions: User must be a part of a community group.

Test Data: Motivation message content.

Test Steps:

1. Navigate to the community tab.
2. Go to selected group.
3. Post a motivational message.
4. View responses and reactions to the posted message.

Postconditions: Log out.

Expected Result: User successfully posts a motivational message and receives responses.

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 -

Integration with Wearable Devices

Feature description -

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

Userstory title -

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

Acceptance criteria -

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT01

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: This test case verifies that the app is compatible with multiple wearable devices and can integrate with them successfully.

Test Suite: Wearable Integration

Test Priority: High

Preconditions: There are different wearable devices available and connected to the mobile.

Test Data: Different wearable device details and compatibility protocols

Test Steps:

1. Open the app.
2. Go to wearable devices integration page.
3. Choose a wearable device.
4. Confirm the integration.

Postconditions: Check the wearable device for confirmatory message of successful integration

Expected Result: The app should successfully integrate with the selected wearable device.

Severity: Critical

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT02

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: This test case verifies that the app can successfully receive and show activity data from the integrated wearable devices.

Test Suite: Data Sync

Test Priority: High

Preconditions: An integrated wearable device is in use and generating activity data.

Test Data: Activity data from the wearable device

Test Steps:

1. Use the wearable device for an activity.
2. Open the app.

3. Go to activity data.

4. Check the synced data.

Postconditions: Perform more activities and verify the sync

Expected Result: The app should show the data from the wearable device.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT03

Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: This test verifies that the app is able to update the user's data in real-time as it is collected from the wearable device.

Test Suite: Real-Time Updates

Test Priority: High

Preconditions: An integrated wearable device is in use

Test Data: Real-time activity data from the wearable device

Test Steps:

1. Use the wearable device for an activity.

2. Open the app.

3. Monitor real-time data.

Postconditions: The data on the app and the wearable device are compared

Expected Result: The app displays real-time data identical to the data on the wearable device.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT04

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: This test case confirms that the app is able to display data about steps taken, heart rate, and calories burned from the wearable device.

Test Suite: Data Types

Test Priority: High

Preconditions: The wearable device is worn and in use

Test Data: Steps, heart rate, and calories burned data from the wearable device

Test Steps:

1. Use the wearable device for an activity.
2. Open the app.
3. Check the activity data for steps, heart rate, and calories burned information.

Postconditions: Different physical activities are performed to collect data

Expected Result: The app displays accurate data about steps taken, heart rate, and calories burned.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT05

Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: This test case confirms that there are troubleshooting options available in the app to help users when connectivity issues arise.

Test Suite: Troubleshooting

Test Priority: Medium

Preconditions: There is a wearable device that has been integrated before.

Test Data: No test data needed

Test Steps:

1. Open the app.
2. Go to devices.
3. Disconnect a device.
4. Go to troubleshooting and follow the recommended steps.

Postconditions: Attempt to reconnect the device

Expected Result: The troubleshooting guide should help the user to resolve the issue and reconnect the device.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 86

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Water Intake Tracking

Feature description -

Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

Userstory title -

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

Acceptance criteria -

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC01

Test Case Title: Verify that users can log their daily water intake.

Test Case Description: This test aims to verify the functionality of logging daily water intake of the user.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User has installed the Fitness Tracker app.
- User is registered and logged in to the app.

Test Data: User water intake data.

Test Steps:

1. Login into app.
2. Navigate to the Water Intake Tracking section.

3. Enter the amount of water consumed.

4. Save the entry.

Postconditions:

- Water intake is logged in the user's account.

Expected Result: User's daily water intake is successfully logged.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC02

Test Case Title: Verify that the app calculates total water consumption for the day.

Test Case Description: Checking if the app can successfully calculate total water intake from the entries logged throughout the day.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has at least one water intake entry.

Test Data: Logged water intake entries.

Test Steps:

1. Login into app.
2. Navigate to the Water Intake Tracking section.
3. View daily total water consumption.

Postconditions:

- Total daily water intake is displayed.

Expected Result: The app successfully calculates and displays total daily water consumption.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC03

Test Case Title: Verify that users can set daily hydration goals.

Test Case Description: This test is to ensure that the application permits users to set daily hydration goals.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged in.

Test Data: Desired hydration goal.

Test Steps:

1. Login into the app.
2. Navigate to the Water Intake Tracking section.
3. Set the hydration goal.

Postconditions:

- The hydration goal is set and displayed on the user's account.

Expected Result: The user is able to set daily hydration goals successfully.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC04

Test Case Title: Verify if reminders are provided to encourage regular water intake.

Test Case Description: This test will check whether the app sends reminders to the user for regular water intake.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- The reminder function is enabled.

Test Data: No test data needed.

Test Steps:

1. Login into app.
2. Enable the reminder function.
3. Wait to receive a reminder.

Postconditions:

- User receives a reminder notification.

Expected Result: The app sends reminders on regular intervals to encourage water intake.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC05

Test Case Title: Verify that users can view hydration progress and trends.

Test Case Description: This test aims to validate that the application displays hydration progress and trends based on the data logged.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has logged water intake for more than one day.

Test Data: No test data needed.

Test Steps:

1. Login into app.
2. Navigate to the Water Intake Tracking section.
3. View the hydration progress and trends.

Postconditions:

- Hydration progress and trends are displayed.

Expected Result: User can successfully view their hydration progress and trends.

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:

Test Case 1

Test Case ID: FT1

Test Case Title: Verify that users can log daily meals and snacks.

Test Case Description: To ensure that users are able to log daily meals and snacks accurately. The focus is when the user inputs necessary information.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in to the Fitness Tracker app
- User is on the Nutrition Tracking screen

Test Data: Sample food items and snacks for data input.

Test Steps:

1. Navigate to the Nutrition Tracking screen.
2. Click on 'Add Meal'.
3. Enter details of meal (type of food, quantity).
4. Save the entry.

Postconditions:

- Entry is saved and displayed in the food diary
- Nutritional details are updated accordingly

Expected Result: The user's meal is successfully logged into the system, influencing the total daily intake data.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: FT2

Test Case Title: Verify that the app calculates macronutrient intake.

Test Case Description: To test the app's capability to calculate macronutrient intake correctly based on the user's food log entries.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User has added a meal in the Nutrition Tracking section

Test Data: Predefined data of macronutrients amount for specific foods

Test Steps:

1. View the nutritional details after logging a meal
2. Cross-check the calculated macronutrient data with preset value

Postconditions: Nutritional details area gets updated with macronutrient calculation.

Expected Result: The app accurately calculates and displays the user's macronutrient intake, reflecting the newly added meal.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: FT3

Test Case Title: Verify that users can set nutritional goals and track progress.

Test Case Description: To test if users can set their nutritional targets and monitor their progress.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is on the Nutrition Goals setting area of the application

Test Data: No test data needed

Test Steps:

1. Navigate to the Goal Setting area.

2. Configure target calorie count and macronutrient ratios.
3. Save these nutrition goals.
4. Track the progress over a couple of days.

Postconditions:

- Goals updates get reflected
- Progress report changes depending on food log entries and goals set

Expected Result: Nutritional goals are properly set and progress is tracked accurately.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: FT4

Test Case Title: Verify that app provides nutritional information and suggestions.

Test Case Description: To test if the app can provide users with helpful nutritional information and suggestions based on their dietary habits.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- User has a history of recorded meals in the Nutrition Tracking section

Test Data: No test data needed

Test Steps:

1. Navigate to the Suggestions area of Nutrition Tracking.
2. Review the presented information and recommendations.

Postconditions: Suggestions and nutritional information change based on the user's dietary habits and

preferences.

Expected Result: The app provides accurate and personalized food suggestions and nutritional information to the user.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: FT5

Test Case Title: Verify that users can view their nutritional history and trends.

Test Case Description: To test if the app allows users to visualize their past nutritional data and note any trends over time.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- User has a history of recorded meals in the Nutrition Tracking section

Test Data: No test data needed

Test Steps:

1. Navigate to the History/ Trends area of Nutrition Tracking.
2. Check if past nutritional information is present and trends are clearly visible.

Postconditions: User's past nutritional data and changes over time are clearly displayed

Expected Result: The user can easily access and view their nutritional history and detect trends.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Data - 88

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Progress Reports

Feature description -

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

Userstory title -

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

Acceptance criteria -

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

Generated Test Cases:

Test Case 1:

- Test Case ID: TC_FT_01
- Test Case Title: Verify that the app sends weekly and monthly progress reports to users.
- Test Case Description: This test case will check the functionality of automatic generation and dispatch of weekly and monthly progress reports to users.
- Test Suite: Progress Reports Module
- Test Priority: High
- Preconditions: User has an active account with ongoing recorded activities for the past month.
- Test Data: User activity data for the last month.
- Test Steps:
 1. Log in to the user account.
 2. Wait for the end of the week/month.
 3. Check user's email for progress report.
- Postconditions:
 1. Validate the received progress report.
 2. Archive the progress report in the user's app account.
- Expected Result: Weekly and monthly progress reports are sent to users without prompt.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 2:

- Test Case ID: TC_FT_02
- Test Case Title: Verify that progress reports include activity levels, workout performance, and goal attainment.
- Test Case Description: This test case evaluates the content of progress reports, ensuring they feature essential

fitness tracking information: activity levels, workout performance, and goal achievements.

- Test Suite: Progress Reports Module

- Test Priority: Medium

- Preconditions:

1. User has an active account with ongoing activities.

2. User received a progress report.

- Test Data: Latest user progress report.

- Test Steps:

1. Log in to the user account.

2. Open the latest progress report.

3. Review the report's content.

- Postconditions: None

- Expected Result: The progress report contains information on activity levels, workout performance, and goal attainment.

- Severity: Critical

- Priority: High

- Type of Testing: Functional Testing

- Test Case Approach: Positive

Test Case 3:

- Test Case ID: TC_FT_03

- Test Case Title: Verify that users can view detailed breakdowns of achievements and trends in their progress reports.

- Test Case Description: This test case checks the feature that allows users to access detailed information and analytics about their performance trends and achievements.

- Test Suite: Progress Reports Module

- Test Priority: Medium

- Preconditions: User has an active account and has received at least one progress report.
- Test Data: Latest user progress report.
- Test Steps:
 1. Log in to the user account.
 2. Open the latest progress report.
 3. Look for detailed breakdown and comprehensive analysis of their fitness data.
- Postconditions: None
- Expected Result: Progress reports offer comprehensive analysis and detailed breakdowns of user performance information.
- Severity: Major
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 4:

- Test Case ID: TC_FT_04
- Test Case Title: Verify that the app provides personalized recommendations based on report findings.
- Test Case Description: This test case inspects how the app identifies trends and developments in the user's fitness data and provides personalized recommendations for improvements.
- Test Suite: Progress Reports Module
- Test Priority: Medium
- Preconditions: User has an active account and received at least one progress report with recommended advice.
- Test Data: Latest user progress report.
- Test Steps:
 1. Log in to the user's account.
 2. Open the latest progress report.

3. Review the tailored recommendations based on user's fitness data.

- Postconditions: None
- Expected Result: The progress report contains personalized recommendations based on user's data.
- Severity: Major
- Priority: Medium
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 5:

- Test Case ID: TC_FT_05
- Test Case Title: Verify that users can customize the frequency and content of their reports.
- Test Case Description: This case verifies the settings for customizing the frequency and content of reports in the app.
- Test Suite: Progress Reports Module
- Test Priority: Medium
- Preconditions: User has an active account.
- Test Data: No test data needed.
- Test Steps:
 1. Log in to the user's account.
 2. Navigate to settings.
 3. Customize the frequency and content of reports.
 4. Save the changes.
 5. Check the changes in next received report.
- Postconditions: None
- Expected Result: Customized changes are saved and reflected in the next received report.
- Severity: Minor
- Priority: Medium

- Type of Testing: Functional 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:

Test Case 1:

- Test Case ID: FTC001
- Test Case Title: Verify that users are able to join various in-app challenges.
- Test Case Description: This test case checks the functionality of joining in-app fitness challenges provided in the application.
- Test Suite: In-App Challenges
- Test Priority: High
- Preconditions: An active user account, various in-app challenges are available.
- Test Data: User login credentials.
- Test Steps:
 1. Login to the application.
 2. Navigate to the 'Challenges' section.
 3. Select a challenge to join.
 4. Click on 'Join Challenge'.
- Postconditions: User is enrolled in the selected challenge.
- Expected Result: The user should be able to successfully join the in-app challenge.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 2:

- Test Case ID: FTC002
- Test Case Title: Verify that each challenge has clear objectives and rules.
- Test Case Description: This test case is intended to confirm the clarity of challenge objectives and rules.
- Test Suite: In-App Challenges
- Test Priority: Medium

- Preconditions: An active user account, joined at least one challenge.
- Test Data: No test data needed.
- Test Steps:
 1. Login to the application.
 2. Navigate to the 'Challenges' section.
 3. Select a challenge to review details.
- Postconditions: User understands the objectives and rules of the challenge.
- Expected Result: The selected in-app challenge should have clear objectives and rules presented.
- Severity: Minor
- Priority: Medium
- Type of Testing: Usability Testing
- Test Case Approach: Positive

Test Case 3:

- Test Case ID: FTC003
- Test Case Title: Verify that users are can track their progress within the challenge.
- Test Case Description: This test case checks that users can track their progress during a challenge.
- Test Suite: In-App Challenges
- Test Priority: High
- Preconditions: An active user account, user has joined at least one challenge.
- Test Data: User login credentials, ongoing challenge information.
- Test Steps:
 1. Login to the application.
 2. Navigate to the 'Challenges' section.
 3. Select an ongoing challenge.
 4. View progress details.
- Postconditions: User can see progress for the challenge.

- Expected Result: The user should be able to successfully view progress within the challenge.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 4:

- Test Case ID: FTC004
- Test Case Title: Verify that rewards are provided upon achieving challenge milestones.
- Test Case Description: This test case confirms that users are rewarded after achieving specific milestones in a challenge.
- Test Suite: In-App Challenges
- Test Priority: High
- Preconditions: An active user account, user has achieved a milestone in a challenge.
- Test Data: User login credentials, completed challenge milestone.
- Test Steps:
 1. Login to the application.
 2. Navigate to the 'Rewards' section.
 3. View earned rewards.
- Postconditions: User can see rewards for completed challenge milestones.
- Expected Result: The user should receive rewards upon achieving challenge milestones as promised.
- Severity: Major
- Priority: High
- Type of Testing: Functional Testing
- Test Case Approach: Positive

Test Case 5:

- Test Case ID: FTC005
- Test Case Title: Verify the functionality of leaderboards for competitions with friends.
- Test Case Description: This test case checks the functionality of in-app leaderboards in relation to challenge competitions with friends.
- Test Suite: In-App Challenges
- Test Priority: Medium
- Preconditions: An active user account, user has engaged in a competition with friends.
- Test Data: User login credentials, friend's list, ongoing or completed competitions.
- Test Steps:
 1. Login to the application.
 2. Navigate to the 'Leaderboards' section.
 3. View ongoing or past competitions.
- Postconditions: User can view rankings within leaderboards.
- Expected Result: The user should be able to view and correctly interpret rankings on competition leaderboards.
- Severity: Minor
- Priority: Medium
- Type of Testing: Usability 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:

Test Case 1:

Test Case ID: FTC001

Test Case Title: Verify that users can select which fitness metrics to display on their dashboard.

Test Case Description: This test case is designed to validate the functionality of user control over the selection and display of fitness metrics on their application dashboard.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions: User is logged into the app, user has recorded some metrics (like steps taken, calories burned).

Test Data: User profile, Activity data (like steps, calories).

Test Steps:

1. Log into app.
2. Navigate to dashboard.
3. Open dashboard customization settings.
4. Select desired fitness metrics to display.
5. Confirm changes.

Postconditions:

1. Dashboards data gets updated.
2. User returns to the main dashboard screen.

Expected Result: The selected metrics are displayed correctly on the users' dashboard.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTC002

Test Case Title: Verify the app allows customization of dashboard layout and theme.

Test Case Description: The purpose of this test case is to ensure that users can personalize the layout and theme of their dashboard within the application.

Test Suite: Dashboard Customization

Test Priority: Medium

Preconditions: User is logged into the app.

Test Data: No test data needed

Test Steps:

1. Log into the app.
2. Navigate to dashboard.
3. Open dashboard customization settings.
4. Modify layout and theme.
5. Save changes.

Postconditions: Dashboard layout and theme are updated.

Expected Result: The dashboard layout and theme should match the custom settings chosen by the user.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTC003

Test Case Title: Verify that user can save and update their dashboard settings.

Test Case Description: This Test case is to check if the app allows the user to save and update the dashboard settings.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions: User is logged into the app, user has modified dashboard settings.

Test Data: Dashboard Settings, User Profile.

Test Steps:

1. Log into app.
2. Navigate to dashboard.
3. Open dashboard customization settings.
4. Modify settings.

5. Save changes.

6. Reopen dashboard settings to verify the changes.

Postconditions: Dashboard settings are updated.

Expected Result: The dashboard settings should be saved and update correctly based on user modifications.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTC004

Test Case Title: Verify dashboard displays update in real-time.

Test Case Description: This Test case ensures that the application dashboard displays update in real-time as the user's metrics change.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions: User is logged in, user has set some fitness metrics.

Test Data: Current fitness metrics data.

Test Steps:

1. Log into the app.
2. Navigate to dashboard.
3. Perform physical activity to change fitness metrics.
4. Check if dashboard displays have updated.

Postconditions: Dashboard displays have updated.

Expected Result: The dashboard displays should update in real-time to reflect changes in user's fitness metrics.

Severity: Critical

Priority: High

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTC005

Test Case Title: Verify that user can reset the dashboard to default settings.

Test Case Description: This test case validates the functionality that allows users to reset their personalized dashboard settings to the default settings.

Test Suite: Dashboard Customization

Test Priority: Low

Preconditions: User is logged in, user's dashboard settings have been personalized.

Test Data: Default dashboard settings.

Test Steps:

1. Log into the app.
2. Navigate to dashboard.
3. Open dashboard customization settings.
4. Click on 'Reset to Default' button.
5. Confirm reset.

Postconditions: Dashboard settings have been reset to default.

Expected Result: All the user's dashboard settings should revert back to the default settings after executing reset.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 91

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Social Sharing

Feature description -

Users can share workout summaries, achievements, and progress updates on social media platforms, inspiring others and fostering accountability. Social sharing enables users to celebrate their successes, seek support from their social networks, and stay motivated on their fitness journey.

Userstory title -

As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

Acceptance criteria -

['Users can share workout summaries and achievements.', 'The app supports sharing on multiple social media platforms.', 'Users can customize the content and format of shared posts.', 'Sharing features are integrated seamlessly with social media.', 'Users can track the engagement of their shared posts.']

Generated Test Cases:

Test Case 1

Test Case ID: FT001

Test Case Title: Verify that users can share workout summaries and achievements on the fitness tracker app.

Test Case Description: This case tests if the user is able to share their workout summaries and achievements through the app.

Test Suite: Integration Testing

Test Priority: High

Preconditions:

- User is logged in to the app
- User has workout summaries and achievements to share

Test Data: User account credentials, workout summary, achievement data.

Test Steps:

1. Login to the app
2. Navigate to workout summary or achievements
3. Click on 'share' button
4. Select social media platforms to share on and click 'confirm'

Postconditions:

- Summary/Achievement is shared on selected platforms
- App returned to previous state

Expected Result: Workout summary or achievement is shared on the selected social media platform

Priority: High

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: FT002

Test Case Title: Verify that the app supports sharing on multiple social media platforms.

Test Case Description: This case tests if a user can share data simultaneously on multiple social media platforms.

Test Suite: System Testing

Test Priority: Medium

Preconditions:

- User is logged in to the app
- User has workout summaries and achievements to share

Test Data: Social media account credentials, workout summary, achievement data.

Test Steps:

1. Login to the app
2. Navigate to workout summary or achievements
3. Click on 'share' button
4. Select multiple social media platforms to share on and click 'confirm'

Postconditions:

- Summary/Achievement is shared on selected platforms
- App returned to previous state

Expected Result: The data is shared on all selected social media platforms.

Priority: High

Severity: Major

Type of Testing: System Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: FT003

Test Case Title: Verify that the sharing features are integrated seamlessly with social media platforms.

Test Case Description: This case tests if the sharing feature of the app works seamlessly with social media platforms.

Test Suite: Functional Testing

Test Priority: High

Preconditions:

- User is logged in to the app
- User has workout summaries and achievements to share

Test Data: Social media account credentials, workout summary, achievement data.

Test Steps:

1. Login to the app
2. Navigate to workout summary or achievements
3. Click on 'share' button
4. Select a social media platform to share on and click 'confirm'

Postconditions:

- Summary/Achievement is shared on selected platform
- User is redirected back to the app

Expected Result: Workout summary or achievement is posted on the selected social media platform without issues

Priority: High

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: FT004

Test Case Title: Verify that users can customize the content and format of shared posts.

Test Case Description: This case tests if the user is able to modify the content and format of the post before sharing.

Test Suite: Function Testing

Test Priority: Medium

Preconditions:

- User is logged in to the app
- User has workout summaries and achievements to share

Test Data: User account credentials, workout summary, achievement data, custom text.

Test Steps:

1. Login to the app
2. Navigate to workout summary or achievements
3. Click on 'share' button
4. Edit the content and format of the post
5. Click 'Confirm'

Postconditions:

- Customized summary/achievement is shared
- App returned to previous state

Expected Result: The customized post is shared on the selected social media platforms.

Priority: Medium

Severity: Minor

Type of Testing: Functional testing

Test Case Approach: Positive

Test Case 5

Test Case ID: FT005

Test Case Title: Verify that users can track the engagement of their shared posts.

Test Case Description: This case tests if the user is able to track likes, comments, and shares on their posts made through the app.

Test Suite: Regression Testing

Test Priority: Low

Preconditions:

- User is logged in to the app
- User has shared a workout summary or achievement

Test Data: User account credentials, previously shared workout summary or achievement.

Test Steps:

1. Login to the app
2. Navigate to the post engagement tracking section
3. Select a previously shared post

Postconditions:

- User views post engagement stats
- App returned to previous state

Expected Result: User is able to view the engagement stats of their shared posts

Priority: Low

Severity: Trivial

Type of Testing: Usability 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:

Test Case 1:

Test Case ID: TC_FTA_01

Test Case Title: Verify that users can log rest days and recovery activities.

Test Case Description: This test case ensures that users can successfully input rest days and recovery activities into the app.

Test Suite: Rest & Recovery

Test Priority: High

Preconditions:

- User is registered and logged in.
- User has completed a physical activity.

Test Data: User activity logs

Test Steps:

1. Open the app.
2. Navigate to the Rest & Recovery section.
3. Input rest day.
4. Log recovery activity.

Postconditions:

- Data is saved in user profile.
- Activity log is updated.

Expected Result: App allows input of rest days and recovery activities, with data reflected accurately in the user profile and activity log.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_FTA_02

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 correctly logs and updates resting heart rate and sleep quality metrics.

Test Suite: Rest & Recovery

Test Priority: High

Preconditions:

- User is registered and logged in.
- Device connected to the app has these monitoring capabilities.

Test Data: Health metrics data

Test Steps:

1. Open the app.
2. Navigate to the Rest & Recovery section.
3. Check resting heart rate and sleep quality metrics.

Postconditions:

- Metrics data logged correctly.
- User's health report is updated.

Expected Result: The app accurately tracks and logs resting heart rate and sleep quality.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_FTA_03

Test Case Title: Verify that users receive recovery recommendations based on their activities.

Test Case Description: This test case aims to ensure the app provides suitable recovery suggestions based on the user's logged physical activities.

Test Suite: Rest & Recovery

Test Priority: High

Preconditions:

- User is registered and logged in.

- User has logged physical activities.

Test Data: User activity logs

Test Steps:

1. Open the app.
2. Navigate to the Rest & Recovery section.
3. Check the recovery recommendations.

Postconditions:

- Recommendations are based on user's activity data.

Expected Result: User receives appropriate recovery recommendations.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_FTA_04

Test Case Title: Verify that the app provides alerts on insufficient recovery time.

Test Case Description: This test case checks if the app successfully warns users when their logged recovery periods are less than recommended.

Test Suite: Rest & Recovery

Test Priority: High

Preconditions:

- User is registered and logged in.
- User has logged physical activities and recovery periods.

Test Data: User activity logs

Test Steps:

1. Open the app.
2. Navigate to the Rest & Recovery section.
3. Check if the insufficiency alert is triggered.

Postconditions:

- Alert triggered upon detection of insufficient recovery periods.

Expected Result: Alerts regarding insufficient recovery periods are received on time.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_FTA_05

Test Case Title: Verify that users can view trends and history of their recovery data.

Test Case Description: This test case ensures that the app correctly displays historical recovery data and identifies discernible trends.

Test Suite: Rest & Recovery

Test Priority: Medium

Preconditions:

- User is registered and logged in.
- User has logged physical activities and recovery data for a certain period.

Test Data: User activity logs

Test Steps:

1. Open the app.
2. Navigate to the Rest & Recovery section.
3. Check the trends and history of recovery data.

Postconditions:

- Display of all past recovery data.
- Identified trends in data.

Expected Result: User is able to view historical recovery data and detect trends.

Severity: Minor

Priority: Medium

Type of Testing: Non-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:

Test Case 1:

Test Case ID: TC001_AudioCoaching

Test Case Title: Verify that the user can access a variety of audio-guided workouts.

Test Case Description: This test case ensures that different types of audio-guided workouts are available to the user.

Test Suite: Workout Functionality

Test Priority: High

Preconditions:

- User is registered and logged in to the app.
- Internet connection is stable.
- Audio speakers or headphones are functional.

Test Data: User login details.

Test Steps:

1. Log into the Fitness tracker app.
2. Navigate to the 'Workouts' page and select 'Audio Coaching'.
3. Browse through the available audio-guided workouts.

Postconditions:

- Log out of the app.

Expected Result: The user is able to browse and access various types of audio-guided workouts.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002_AudioCoaching

Test Case Title: Verify that audio coaching includes motivational messages and coaching cues.

Test Case Description: This test checks if the audio-guided workouts contain motivational messages and coaching cues to guide the user.

Test Suite: Content Validation

Test Priority: High

Preconditions:

- User is logged into the app.
- User has started an audio-guided workout.

Test Data: No test data needed

Test Steps:

1. Log into the app.
2. Start an audio-guided workout.
3. Listen to the audio coaching.

Postconditions:

- End the workout.

Expected Result: The audio coaching includes motivational messages and coaching cues.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC003_AudioCoaching

Test Case Title: Verify that users can choose coaching based on workout type and intensity.

Test Case Description: This test verifies that the app allows users to select the type and intensity of the workout in the audio coaching feature.

Test Suite: Workout Functionality

Test Priority: High

Preconditions:

- User has successfully logged into the app.
- The audio coaching feature is enabled.

Test Data: User-defined workout type and intensity

Test Steps:

1. Open the 'Workouts' section and click on 'Audio Coaching'.
2. Select a preferred workout type and intensity level.
3. Start the workout and listen to the audio coaching.

Postconditions:

- Log out from the app.

Expected Result: User is able to select and start a workout based on preferred type and intensity, receiving appropriate audio coaching.

Severity: Major

Priority: High

Type of Testing: Functional testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004_AudioCoaching

Test Case Title: Verify that audio coaching integrates seamlessly with other app features.

Test Case Description: This test ensures that the audio coaching feature does not interfere with other features of the app and works in harmony.

Test Suite: Integration Testing

Test Priority: Medium

Preconditions:

- User is logged into the app.
- User has initiated a workout with audio coaching.

Test Data: No test data needed

Test Steps:

1. Start a workout with audio coaching.
2. While workout is ongoing, use other features such as step tracking, heart rate monitoring.
3. Observe app behavior.

Postconditions:

- End the workout.

Expected Result: Audio coaching and other features work simultaneously without any disruptions.

Severity: Critical

Priority: Medium

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005_AudioCoaching

Test Case Title: Verify that users can control audio settings and preferences.

Test Case Description: This test checks whether users can alter audio settings within the app during a workout.

Test Suite: User Interface Testing

Test Priority: Medium

Preconditions:

- User is logged into the app.
- User has started a workout with audio coaching.

Test Data: No test data required

Test Steps:

1. Start a workout with audio coaching.
2. Navigate to the Audio Settings within the app.
3. Change the volume, mute/unmute or switch between speakers and headphones.

Postconditions:

- End the workout.

Expected Result: User can successfully control audio settings as per preference.

Severity: Major

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Data - 94

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Health Apps

Feature description -

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

Userstory title -

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

Acceptance criteria -

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

Generated Test Cases:

Test Case ID: FTC-001

Test Case Title: Verify that the app integrates with popular health and wellness apps.

Test Case Description: This test case is designed to ensure that the fitness tracker app is compatible with major health and wellness applications on both iOS and Android platforms.

Test Suite: Integration

Test Priority: High

Preconditions:

- The user has a fitness tracker app installed.

- The user has a popular health app installed.
- The user is logged in to the fitness tracker app.

Test Data: Health app details (name, integration endpoints, data format)

Test Steps:

1. Open the fitness tracker app.
2. Navigate to 'Settings'.
3. Click on 'Connect to Health Apps'.
4. Select a health app from the list.
5. Confirm the integration.

Postconditions:

- Data from the selected health app is imported to the fitness tracker app.
- The chosen health app is listed under 'Connected Apps'.

Expected Result: The fitness tracker app integrates seamlessly with the chosen health app, and health data is synced and consolidated correctly.

Severity: Major

Priority: High

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: FTC-002

Test Case Title: Verify that the app provides a consolidated view of health metrics.

Test Case Description: This test checks whether health data from connected apps is accurately compiled and displayed in a unified, understandable format.

Test Suite: Integration

Test Priority: High

Preconditions:

- The user has successfully linked at least one health app.

- The health app has data that can be imported.

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app.
2. Navigate to the 'Dashboard'.
3. Review the consolidated health data.

Postconditions:

- Health metrics from the linked app are updated and accurately represented in the dashboard.
- The user can view and interpret data with ease.

Expected Result: The consolidation of health metrics from different apps is accurate, up-to-date, and represents meaningful insights.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

These are example test cases. You would need to repeat a similar process for all other acceptance criteria.

Data - 95

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

Feature name -

Body Composition Tracking

Feature description -

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and the effectiveness of exercise and nutrition interventions.

Userstory title -

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed decisions about my training and nutrition.

Acceptance criteria -

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set goals for body composition metrics.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_BCT_01

Test Case Title: Verify that users can log body composition metrics such as weight, body fat percentage, and muscle mass.

Test Case Description: This test case checks the ability of users to input and save different body composition data.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions: Users have installed the app and registered for an account

Test Data: Weight - 170 lbs, Body fat percentage - 18, Muscle mass - 60kg

Test Steps: 1. Launch the application and log in 2. Navigate to body composition tracking feature 3. Enter given test data 4. Click on save button.

Postconditions: Users can view the saved data on their profile

Expected Result: Input data is accurately displayed on user's profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_BCT_02

Test Case Title: Verify that the app calculates and displays BMI based on logged data.

Test Case Description: This test case is designed to confirm that the application properly calculates and shows BMI based on the user's input of height and weight.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions: Users have installed the app and registered for an account and logged body composition data

Test Data: Height - 180 cm, Weight - 75 kg

Test Steps: 1. Launch the app and log in 2. Navigate to body composition tracking feature 3. Users observe BMI calculated by app

Postconditions: User can view the calculated BMI on their profile

Expected Result: BMI value displayed is accurate based on the weight and height data provided.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_BCT_03

Test Case Title: Verify that users can view trends and history of body composition changes.

Test Case Description: This test case verifies that the application accurately tracks and displays a history of changes in body composition metrics.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions: Users have installed the app and registered for an account and logged body composition data over time

Test Data: No test data needed

Test Steps: 1. Launch the app and log in 2. Navigate to body composition tracking feature 3. Users view the history and trends in body composition changes

Postconditions: User can understand body composition trends over time

Expected Result: Recent and past body composition data is displayed appropriately.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_BCT_04

Test Case Title: Verify that the app provides insights and recommendations based on body composition data.

Test Case Description: This test case checks if the application generates useful insights and gives

recommendations based on logged body composition data

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions: Users have installed the app and registered for an account and logged body composition data over time

Test Data: No test data needed

Test Steps: 1. Launch the application and log in 2. Navigate to body composition tracking feature 3. Observe the insights or recommendations provided by the app based on body composition data

Postconditions: User can derive insights and follow recommendations to improve their fitness regimen.

Expected Result: App correctly generates actionable insights and recommendations based on user's body composition data.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_BCT_05

Test Case Title: Verify that users can set goals for body composition metrics.

Test Case Description: This test case validates whether users can specify goals for different body composition metrics.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions: Users have installed the app and registered for an account

Test Data: Goal for weight - 150 lbs, Goal for body fat percentage - 15, Goal for muscle mass - 65kg

Test Steps: 1. Launch the application and log in 2. Navigate to body composition tracking feature 3. Set goals

as per the test data and save it.

Postconditions: Goals are saved and users can view them on their profile

Expected Result: Goals are saved accurately and reflected in users' profile.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability 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:

Test Case 1:

Test Case ID: FT1

Test Case Title: Verify that users can view detailed performance metrics for each workout.

Test Case Description: This test validates the functionality of viewing the detailed performance metrics of a specific workout.

Test Suite: Workout Analysis

Test Priority: High

Preconditions: User must have completed at least one workout, User is logged into the app, User is on the "workout analysis" page.

Test Data: Past workout data.

Test Steps: 1. Log into the Fitness Tracker App. 2. Navigate to the "Workout Analysis" page. 3. Select a workout to analyze. 4. Inspect the performance metrics.

Postconditions: User returns to workout list or exits application.

Expected Result: The user should be able to view detailed performance metrics, and this information should match their workout records.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT2

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration.

Test Case Description: This test is to ensure the system correctly generates trends and insights based on workout intensity and duration.

Test Suite: Workout Analysis

Test Priority: High

Preconditions: User must have completed multiple workouts, User is logged in, User is on the Workout Analysis page.

Test Data: Multiple past workout data.

Test Steps: 1. Log in to the app. 2. Navigate to 'Workout Analysis'. 3. Choose to view trends/insights.

Postconditions: User navigates to other portions of the app or logs out.

Expected Result: The app should accurately display trends and insights based on the intensity and duration of past workouts.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT3

Test Case Title: Verify that users can track calorie burn and other health metrics.

Test Case Description: This test case validates that users can track their calorie burn and other health metrics successfully.

Test Suite: Health Metrics

Test Priority: High

Preconditions: User has completed a workout, User is logged in, User is on the Health Metrics page.

Test Data: Exercise data (calories burned, steps taken, heart rate etc.).

Test Steps: 1. Log in to the app. 2. Navigate to 'Health Metrics'. 3. Verify that health metrics are accurately displayed.

Postconditions: User navigates to other portions of the app or logs out.

Expected Result: App accurately displays calorie burn and other health metrics based on logged workouts.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT4

Test Case Title: Verify that the app offers recommendations based on workout analysis.

Test Case Description: This test validates the functionality of the app to make recommendations based on the analysis of users' workout.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions: User has completed a workout, User is logged in, User is on the Workout Analysis page.

Test Data: Exercise data (type of exercise, duration, intensity etc.).

Test Steps: 1. Log in to the app. 2. Navigate to 'Workout Analysis'. 3. Verify that recommendations are given based on the analysis.

Postconditions: User navigates to other portions of the app or logs out.

Expected Result: The app should provide recommendations based on the user's workout data accurately.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT5

Test Case Title: Verify that users can compare current workouts with past performance.

Test Case Description: This test validates the ability of the app to compare current workout data with past data.

Test Suite: Workout Analysis

Test Priority: High

Preconditions: User has multiple logged workouts, User is logged in, User is on the workout comparison page.

Test Data: Multiple workout data (current and past).

Test Steps: 1. Log into the application. 2. Navigate to the "Workout Comparison" page. 3. Initiate a comparison between current and past workout.

Postconditions: User navigates to other portions of the app or logs out.

Expected Result: The user should be able to view and understand the comparison between their current workouts and past performances.

Severity: Major

Priority: High

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:

Test Case ID: TC_FITAPP_001

Test Case Title: Verify that the app provides current weather conditions and forecasts

Test Case Description: The test case will verify that the app can retrieve and display the current weather conditions and forecasts for the user's location.

Test Suite: Functional Testing

Test Priority: High

Preconditions:

- The user is logged in.
- Location services are enabled.

Test Data: User's location coordinates

Test Steps:

1. Log into the fitness tracker app
2. Navigate to the Weather page
3. Confirm that current weather conditions and forecasts are displayed in real-time

Postconditions:

- User logs out of the app.

Expected Result: The user can view the real-time current weather conditions and forecasts for their location.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TC_FITAPP_002

Test Case Title: Verify that users can view weather details for different locations

Test Case Description: The test case will confirm that users can manually input a different location to view weather details.

Test Suite: Functional Testing

Test Priority: Medium

Preconditions:

- User is logged in
- Weather page is accessible

Test Data: Different valid location coordinates

Test Steps:

1. Log into the fitness tracker app
2. Navigate to Weather page
3. Manually input a different location
4. Confirm that the weather details change based on the entered location

Postconditions:

- User logs out from the app

Expected Result: User can view weather details for manually inputted different locations.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TC_FITAPP_003

Test Case Title: Verify the app provides personalized recommendations based on weather conditions

Test Case Description: This test case is designed to verify that the app can provide personalized workout recommendations based on the current weather conditions.

Test Suite: Functional Testing

Test Priority: High

Preconditions:

- User is logged in
- Current weather details are available

Test Data: Current weather conditions

Test Steps:

1. Log into the fitness tracker app
2. Navigate to the Weather page
3. Review suggested workout routines

Postconditions:

- User logs out from the app

Expected Result: The app generates personalized workout recommendations based on the current weather conditions.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_FITAPP_004

Test Case Title: Verify that weather data is updated in real-time

Test Case Description: The test case will confirm that the app consistently updates and displays weather information in real-time.

Test Suite: Performance Testing

Test Priority: High

Preconditions:

- User is logged in
- Weather page is open

Test Data: Real-time weather data

Test Steps:

1. Log into the fitness tracker app
2. Navigate to the Weather page
3. Note the displayed time and weather details
4. Check back after a few minutes and compare the time and weather details

Postconditions:

- User logs out of the app

Expected Result: User observes that the weather details on the app are updated in real-time.

Severity: Critical

Priority: High

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case ID: TC_FITAPP_005

Test Case Title: Verify that users can receive weather alerts for extreme conditions

Test Case Description: The test case will verify that the app sends weather alerts to the user when there are extreme weather conditions.

Test Suite: Functional Testing

Test Priority: High

Preconditions:

- User is logged in
- User has enabled weather alerts in the app settings

Test Data: Extreme weather conditions

Test Steps:

1. Simulate extreme weather conditions
2. Log into the fitness tracker app
3. Check if a weather alert for extreme conditions is received

Postconditions:

- User logs out of the app

Expected Result: User receives a weather alert for extreme conditions.

Severity: Critical

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:

Test Case 1:

Test Case ID: TC01_MoodTracking

Test Case Title: Verify that users can log their mood, energy levels, and motivation before and after workouts.

Test Case Description: This test case validates the functionality of mood logging in the fitness tracker app.

Test Suite: MoodTrackingSuite

Test Priority: High

Preconditions: The user is registered and logged into the application.

Test Data: Mood information (happy, motivated), energy levels (high, medium, low), motivation scale (1-10)

Test Steps: 1) User logs in to the app. 2) User performs a workout. 3) User logs their mood, energy, and motivation before and after the workout.

Postconditions: Mood, energy, and motivation data is stored and displayed on user profile.

Expected Result: The app correctly logs and displays the user's mood, energy levels, and motivation.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC02_MoodTracking

Test Case Title: Verify that the app provides insights into mood and energy trends over time.

Test Case Description: This case checks if the app accurately aggregates and presents mood and energy level trends over time.

Test Suite: MoodTrackingSuite

Test Priority: High

Preconditions: The user has logged several workout sessions with mood and energy data.

Test Data: Multiple mood, energy, and motivation data points over time.

Test Steps: 1) User logs in to the app. 2) User navigates to the 'Mood Trends' section. 3) User reviews the displayed mood and energy trends.

Postconditions: None.

Expected Result: The app accurately displays trends in mood and energy levels over time.

Severity: Major

Priority: High

Type of Testing: Data-Driven Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC03_MoodTracking

Test Case Title: Verify that the app can provide recommendations based on mood and energy data.

Test Case Description: This case verifies if the app correctly provides personalized workout recommendations based on the user's mood and energy data.

Test Suite: MoodTrackingSuite

Test Priority: High

Preconditions: The user has logged several workout sessions with mood and energy data.

Test Data: Various mood, energy, and motivation data points.

Test Steps: 1) User logs in to the app. 2) User navigates to the 'Recommendations' section. 3) User reviews the personalized workout recommendations.

Postconditions: None.

Expected Result: The app accurately gives personalized workout recommendations based on mood and energy data.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC04_MoodTracking

Test Case Title: Verify that the app supports journaling for personal reflections on mood and workouts.

Test Case Description: This case tests if the app allows the user to write personal reflections on their mood and workouts in a journal format.

Test Suite: MoodTrackingSuite

Test Priority: Medium

Preconditions: User is registered and logged in to the application.

Test Data: Personal reflections text data.

Test Steps: 1) User logs into the app. 2) User navigates to the 'Mood Journal'. 3) User writes and saves a personal reflection.

Postconditions: Journal entry is saved and displayed in the mood journal section.

Expected Result: The app successfully allows the user to write and save personal reflections in a journal format.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC05_MoodTracking

Test Case Title: Verify that users can set goals for improving mood and energy levels.

Test Case Description: This case tests if the app facilitates setting goals related to mood and energy

improvement.

Test Suite: MoodTrackingSuite

Test Priority: High

Preconditions: The user is registered and logged into the application.

Test Data: Goal information (i.e., increase motivation to 7/10)

Test Steps: 1) User logs into the app. 2) User navigates to 'Mood Goals'. 3) User sets a goal for improving mood and energy levels.

Postconditions: Goal data is saved and displayed under 'Mood Goals' section.

Expected Result: The app successfully allows the user to set goals related to mood and energy improvement.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive.

Data - 99

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Creation

Feature description -

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

Userstory title -

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

Acceptance criteria -

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', 'Recipes are saved in the user's profile.', 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

Generated Test Cases:

Test Case 1

Test Case ID: 001

Test Case Title: Verify that users can input and save ingredients and cooking instructions for a recipe.

Test Case Description: This case tests the user's ability to input and save necessary information for creating a recipe on the platform.

Test Suite: Recipe Creation

Test Priority: High

Preconditions: User must be registered and logged into the platform.

Test Data: Mock data with ingredients and cooking instructions for a test recipe.

Test Steps:

1. Log into the platform
2. Navigate to the 'Create Recipe' page
3. Input ingredients and cooking instructions
4. Save the recipe

Postconditions:

- Recipe should be saved in user's profile
- Message indicating successful recipe creation should be displayed

Expected Result: Recipe is successfully saved with given ingredients and cooking instructions.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: 002

Test Case Title: Verify that users can upload photos for their recipes.

Test Case Description: This test case checks the functionality to upload images for the recipe on the platform.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User must be logged in
- User has created a recipe

Test Data: Image file for the recipe.

Test Steps:

1. Open a created recipe
2. Click on 'Add Photo'
3. Select the image file and upload
4. Save Changes

Postconditions:

- Photo displays with the recipe
- Message indicating successful upload and save should be displayed

Expected Result: Photo is successfully uploaded and saved with the recipe.

Severity: Minor

Priority: High

Type of Testing: Functional Testing, Compatibility Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: 003

Test Case Title: Verify that recipes are accessible to other users.

Test Case Description: This case tests if a user's published recipe can be viewed by other users on the platform.

Test Suite: Recipe Sharing

Test Priority: High

Preconditions:

- User must have created a recipe
- Other users are registered and logged into the platform.

Test Data: No test data required.

Test Steps:

1. User A creates and publishes a recipe
2. User B logs into the platform
3. User B searches and opens User A's recipe

Postconditions: The platform displays User A's recipe details to User B.

Expected Result: Other users can access and view published recipes.

Severity: Major

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: 004

Test Case Title: Verify that a confirmation is received after successful recipe submission.

Test Case Description: This case verifies that the system sends out a confirmation when users successfully create a recipe.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions: User should be logged in and have filled out the 'create recipe' form.

Test Data: No test data needed.

Test Steps:

1. Fill out the 'Create Recipe' form
2. Click on 'Submit Recipe'
3. Check for confirmation message

Postconditions: Confirmation message displayed saying the recipe was submitted successfully.

Expected Result: User receives confirmation after successful recipe submission.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: 005

Test Case Title: Verify that the recipe input form rejects invalid data.

Test Case Description: This case tests the validation of the recipe input form, ensuring it rejects any data input that doesn't meet specified criteria.

Test Suite: Recipe Creation

Test Priority: High

Preconditions: User is logged in and filling out the Recipe creation form.

Test Data: Mock data (including invalid data) for a recipe.

Test Steps:

1. Start filling out the 'Create Recipe' Form.
2. Input invalid data (text in a numeric field, improper measurements, etc.).
3. Attempt to submit the form.

Postconditions: A message indicating input is invalid is displayed and correct data is requested.

Expected Result: The recipe input form rejects invalid data and requests rectification.

Severity: Major

Priority: High

Type of Testing: Negative Testing

Test Case Approach: Negative

Data - 100

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Discovery

Feature description -

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

Userstory title -

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance criteria -

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC1-RDP1 (Recipe Discovery Platform- Test Case 1)

Test Case Title: Verify that users can search for recipes using keywords and ingredients.

Test Case Description: The intent is to confirm that the search function on the platform works correctly when users search for recipes using keywords or ingredients.

Test Suite: Recipe Search Suite

Test Priority: High

Preconditions:

- User is registered on the platform
- User is logged in

Test Data: Test data includes the search keywords and ingredients (e.g., "chicken", "pasta").

Test Steps:

1. User logs into the platform.
2. User enters the keyword or ingredient in the search bar.
3. User clicks the 'Search' button.

Postconditions:

- User is presented with a set of search results.

Expected Result: The search results should correspond to the input keyword or ingredient.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC2-RDP2

Test Case Title: Verify the relevance and accuracy of search results.

Test Case Description: In this test, we're checking to see if a search using the platform's search functionality yields accurate and relevant results.

Test Suite: Recipe Search Suite

Test Priority: High

Preconditions:

- User is registered on the platform.
- User is logged in.
- User has performed a search with specific keywords or ingredients.

Test Data: N/A

Test Steps:

1. User logs into the platform.

2. User performs a keyword/ingredient search.

3. User reviews the search results.

Postconditions:

- User has reviewed the search results

Expected Result: Search result recipes should be relevant to the user's search keywords or ingredients.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Repeat this template and adjust according to the remaining acceptance 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."
