Generated Test Cases via gemini-1.5-pro LLM Model and Zero-shot Prompting Technique

---

Data - 1

Project name -

Social Media Platform

Project description -

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

Feature name -

User Profiles

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for User Profile Feature:

Test Case ID: TP-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 the functionality of adding and editing different profile fields, ensuring users can personalize their profiles.

Test Suite: User Profile Management

Test Priority: High

Preconditions:

* User is logged in.

* User navigates to the profile editing section.

Test Data:

* Valid input data for all profile fields (name, bio, profile picture, location, interests).

* Modified data for each field.

Test Steps:

1. Input valid data into each profile field.

2. Save the profile.

3. Verify that the entered data is displayed correctly on the profile page.

4. Edit the data in each profile field.

5. Save the changes.

6. Verify that the modified data is reflected accurately on the profile page.

Postconditions:

* User profile reflects the latest saved changes.

Expected Result: Users should be able to add, edit, and save changes to all their profile information successfully.

Severity: Critical

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TP-002

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

Test Case Description: This test case validates the profile picture upload functionality by attempting to upload images in various standard formats.

Test Suite: User Profile Management

Test Priority: High

Preconditions:

* User is logged in.

* User is on the profile editing page.

Test Data: Image files in .jpg, .jpeg, .png, .gif formats.

Test Steps:

1. Attempt to upload an image in .jpg format.

2. Verify the image uploads successfully and displays correctly.

3. Repeat steps 1-2 for .jpeg, .png, and .gif formats.

Postconditions:

* User profile displays the uploaded image if the format is supported.

Expected Result: The platform should accept and display images in all tested standard formats.

Severity: Major

Priority: High

Type of Testing: Functional, Compatibility

Test Case Approach: Positive

Test Case ID: TP-003

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

Test Case Description: This test case ensures users can control the visibility of each profile field using privacy settings.

Test Suite: User Profile Privacy

Test Priority: High

Preconditions:

* User is logged in.

* User is in the privacy settings section of their profile.

Test Data: No test data needed.

Test Steps:

1. Navigate to each profile field's privacy setting.

2. Verify that options to control visibility (e.g., Public, Friends only, Private) are available.

3. Select different privacy settings for various fields.

4. Save the privacy settings.

5. Log out and view the profile as a public user and a friend.

6. Verify that the visibility of each field aligns with the chosen privacy settings.

Postconditions:

* User profile visibility reflects the chosen privacy settings.

Expected Result: Users should be able to set individual privacy settings for each profile field, and the platform should respect these settings.

Severity: Critical

Priority: High

Type of Testing: Functional, Security

Test Case Approach: Positive

Test Case ID: TP-004

Test Case Title: Verify that the profile is saved successfully.

Test Case Description: This test case checks the system's ability to save the user's profile information correctly.

Test Suite: User Profile Management

Test Priority: High

Preconditions:

* User is logged in.

* User has made changes to their profile.

Test Data: Any valid input data for the profile fields.

Test Steps:

1. Input data into various profile fields.

2. Click the "Save Profile" button.

3. Verify the system doesn't display any error messages.

4. Refresh the page or log out and log back in.

5. Verify that the entered profile data persists and is displayed correctly.

Postconditions:

* User profile reflects the saved information even after refreshing or re-logging.

Expected Result: The platform should successfully save the user's profile information to the database.

Severity: Blocker

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TP-005

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

Test Case Description: This test case ensures users receive feedback after saving their profile information.

Test Suite: User Profile Management

Test Priority: Medium

Preconditions:

* User is logged in.

* User has made changes to their profile.

Test Data: No test data needed.

Test Steps:

1. Make changes to the profile information.

2. Click the "Save Profile" button.

3. Verify that a confirmation message is displayed (e.g., "Profile saved successfully!").

Postconditions: None

Expected Result: The system should provide visual feedback to the user, confirming that their profile has been successfully saved.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Usability

Test Case Approach: Positive

Data - 2

Project name -

Social Media Platform

Project description -

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

Feature name -

News Feed

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - News Feed Feature

Test Case Creation Template:

| Field | Description |

|---|---|

| Test Case ID | (Provide a unique identifier for each test case) |

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

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

| Test Suite | (Name of the test suite or module) |

| Test Priority | (Priority level: High, Medium, Low) |

| Preconditions | (List any prerequisites before executing the test case, max 3 items, in an unordered list) |

| Test Data | (Specify data required for execution; if none, state "No test data needed") |

| Test Steps | (List steps for executing the test case, min 3 to max 6) |

| Postconditions | (Any follow-up actions after test execution, max 3 items, in an unordered list) |

| Expected Result | (Overall expected outcome of the test case) |

| Severity | (Blocker, Critical, Major, Minor, Trivial) |

| Priority | (High, Medium, Low) |

| Type of Testing | (See list in prompt) |

| Test Case Approach | (Positive, Negative, Destructive) |

Test Cases:

Test Case ID: NF-001

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

Test Case Description: This test case verifies that the news feed only displays posts from accounts that the user has chosen to follow. It checks for the presence of posts from followed accounts and the absence of posts from unfollowed accounts.

Test Suite: News Feed Functionality

Test Priority: High

Preconditions:

- User is logged in.

- User is following at least one account.

Test Data: No test data needed.

Test Steps:

1. Navigate to the news feed page.

2. Observe the posts displayed in the feed.

3. Verify that each post originates from an account the user follows.

4. Verify that no posts from unfollowed accounts are present.

Postconditions: None

Expected Result: Only posts from followed accounts are displayed in the news feed.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: NF-002

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

Test Case Description: This test case verifies the order in which posts are displayed in the news feed. It checks for both chronological and algorithmic ordering options, if applicable.

Test Suite: News Feed Functionality

Test Priority: High

Preconditions:

- User is logged in.
- User has a populated news feed.

Test Data: No test data needed.

Test Steps:

1. Navigate to the news feed page.
2. Identify the timestamp on multiple posts.
3. If chronological ordering is selected, verify that posts are displayed from newest to oldest.
4. If algorithmic ordering is selected, observe the order of posts and verify it aligns with the platform's documented algorithm (e.g., relevance, engagement).

Postconditions: None

Expected Result: Posts in the news feed are displayed according to the selected order - either chronologically or algorithmically.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: NF-003

Test Case Title: Verify that users can interact with posts in the news feed (like, comment, share).

Test Case Description: This test case verifies the functionality of user interactions with posts on the news feed. It checks for liking, commenting on, and sharing posts.

Test Suite: News Feed Interaction

Test Priority: High

Preconditions:

- User is logged in.

- News feed displays posts.

Test Data: Sample comment text.

Test Steps:

1. Navigate to the news feed page.

2. Select a post to interact with.

3. Like the post using the "like" button and verify the like count increments.

4. Comment on the post using the provided test data and verify the comment is added.

5. Share the post using the "share" button and verify the sharing options are presented.

Postconditions:

- Unlike the post.

- Delete the comment.

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

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: NF-004

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

Test Case Description: This test case verifies that the news feed updates dynamically without requiring manual refresh. It checks for new posts and interaction updates appearing in real-time.

Test Suite: News Feed Functionality

Test Priority: High

Preconditions:

- User is logged in.
- User has a friend or account they follow online and active.

Test Data: No test data needed.

Test Steps:

1. Navigate to the news feed page.
2. Coordinate with a friend or observe for a followed account's new post.
3. Observe the news feed for the new post to appear without manual refresh.
4. Have the friend like or comment on a post you can see and observe the interaction update in real-time.

Postconditions: None

Expected Result: The news feed dynamically updates in real-time to reflect new posts and interactions.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Performance Testing

Test Case Approach: Positive

Test Case ID: NF-005

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

Test Case Description: This test case verifies the functionality of filtering options in the news feed. It checks if users can narrow down the displayed content by selecting specific post types or topics.

Test Suite: News Feed Filtering

Test Priority: Medium

Preconditions:

- User is logged in.

- News feed displays diverse content types.

Test Data: No test data needed.

Test Steps:

1. Navigate to the news feed page.

2. Locate the filtering options (if available, these could be dropdown menus, buttons, or dedicated sections).

3. Select a specific post type filter (e.g., photos, videos, articles).

4. Verify that the news feed content updates to display only the selected type.

5. Repeat steps 3-4 for different filter options, including topic filters if available.

Postconditions: Reset any applied filters.

Expected Result: Users can successfully filter the news feed content based on type or topic, displaying only the chosen content.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 3

Project name -

Social Media Platform

Project description -

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

Feature name -

Messaging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Messaging Feature

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

Test Case Creation Template:

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

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| MSG-001 | Verify that users can send text messages. | This test case verifies that a user can successfully send a text message to another user. | Messaging | High | - User is logged in.
- User has at least one friend added. | No test data needed. | 1. Navigate to the messaging section.
2. Select an existing friend or start a new conversation.
3. Enter a text message in the message input field.
4. Click the "Send" button. | - The sent message should be cleared from the input field.
- The message should appear in the conversation history. | The text message is sent successfully and appears in the conversation history for both users. | Blocker | High | Functional, System, Acceptance, End-to-End | Positive |

| MSG-002 | Verify that users can send images and videos. | This test case ensures that users can send image and video files as attachments in private messages. | Messaging | High | - User is logged in.
- User has at least one friend added. | - A sample image file (e.g., .jpg, .png).
- A sample video file (e.g., .mp4, .mov). | 1. Navigate to the messaging section.
2. Select an existing friend or start a new conversation.
3. Click on the image/video attachment icon.
4. Select and upload the image/video file.
5. Click the "Send" button. | - The uploaded files should appear in the conversation history. | The image and video files are sent successfully and appear in the conversation history for both users. | Critical | High | Functional, System, Acceptance, End-to-End | Positive |

| MSG-003 | Verify that messages are sent in real-time. | This test case verifies that messages are delivered and displayed to the recipient immediately upon sending. | Messaging | High | - Two users are logged in.
- Users have an active private conversation. | No test data needed. | 1. User 1 sends a message to User 2.
2. Observe the conversation history on User 2's screen. | N/A | The message sent by User 1 appears immediately on User 2's screen without noticeable delay. | Critical | High | Performance, Functional, System, Acceptance | Positive |

| MSG-004 | Verify that users can start private and group chats. | This test case checks if users can initiate both private conversations with individual friends and group chats with multiple friends. | Messaging | Medium | - User is logged in.
- User has at least two friends added. | No test data needed. | 1. Navigate to the messaging section.
2. To start a private chat, select a friend from the contact list.
3. To start a group chat, select the option to create a new group and add multiple friends. | N/A | The user can successfully start both private conversations and group chats with selected friends. | Major | Medium | Functional, System,

Acceptance | Positive |

| MSG-005 | Verify that users receive notifications for new messages. | This test case ensures that users are notified when they receive a new message, even when not actively viewing the messaging section. | Messaging, Notifications | High | - Two users are logged in.
- User 2 is not actively viewing the messaging section. | No test data needed. | 1. User 1 sends a message to User 2.
2. Observe User 2's interface for a new message notification. | N/A | User 2 receives a visual or audible notification indicating a new message from User 1. | Critical | High | Functional, System, Acceptance | Positive |

Data - 4

Project name -

Social Media Platform

Project description -

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

Feature name -

Content Sharing

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Content Sharing on Social Media Platform

Test Case ID: TC_CS_001

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

Test Case Description: This test case verifies the functionality of creating and sharing a text post on the platform. It ensures that users can input text, successfully publish the post, and that the post is visible in the intended audience's feed.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged into the platform.

- User is on their home feed or profile page.

Test Data: "This is a test post for TC_CS_001."

Test Steps:

1. Click on the "Create Post" button.

2. Select the "Text Post" option.

3. Enter the test data in the text field.

4. Choose the desired audience (followers or public).

5. Click the "Post" button.

Postconditions:

- Verify the post appears on the user's profile.

- Verify the post appears in the feed of the selected audience.

Expected Result: The text post is successfully created and shared with the chosen audience.

Severity: Critical

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_CS_002

Test Case Title: Verify that users can share content with followers or publicly.

Test Case Description: This test case checks the audience selection functionality when sharing content. It ensures users can choose to share posts either with their followers only or make them publicly viewable.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in.

- User has created a text post or uploaded media content.

Test Data: No test data needed.

Test Steps:

1. Navigate to the content sharing section.

2. Create a new post or select existing content for sharing.

3. Locate the audience selection option (e.g., dropdown menu, radio buttons).

4. Choose to share the content with "Followers only."

5. Post the content.

6. Repeat steps 3-5, this time choosing to share "Publicly."

Postconditions:

- Verify the first post is visible only to the user's followers.

- Verify the second post is visible to all users on the platform.

Expected Result: Users can successfully share content with their followers or publicly based on their preference.

Severity: Critical

Priority: High

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: TC_CS_003

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

Test Case Description: This test case ensures users have the ability to modify or remove their previously shared content. It checks for the presence and functionality of edit and delete options for posts.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in.
- User has at least one existing post shared.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user's profile and locate an existing post.
2. Click on the post options menu (usually represented by three dots).
3. Select the "Edit" option.
4. Make changes to the post content.
5. Save the changes and verify the post reflects the modifications.
6. Open the post options menu again.
7. Select the "Delete" option.
8. Confirm the deletion when prompted.

Postconditions:

- Verify the post is updated with the edited content.

- Verify the post is successfully removed from the user's profile.

Expected Result: Users can modify or remove their shared posts using the edit and delete functionalities.

Severity: Critical

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TC_CS_004

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

Test Case Description: This test case verifies the functionality of adding hashtags and tagging other users within posts. It ensures these features work correctly and enhance content searchability and user interaction.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- User is logged in.

- User is creating a new post.

Test Data:

- Hashtag: #TestPost

- Existing username for tagging: [Username of another account]

Test Steps:

1. Start creating a new post.

2. Type the hashtag "#TestPost" within the post content.

3. Type "@" followed by the existing username for tagging (e.g., @[Username of another account]).

4. Post the content.

Postconditions:

- Verify the hashtag appears as a clickable link in the post.

- Verify the tagged username appears as a clickable link to the other user's profile.

Expected Result: The post successfully incorporates both the hashtag and the user tag, linking them appropriately.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: TC_CS_005

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

Test Case Description: This test case checks the feedback mechanisms for posts, ensuring users can like, comment on, and share others' content. It also verifies that the original poster receives notifications or visual cues reflecting these interactions.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- Two users are logged in (User A and User B).

- User A has shared a post.

Test Data: No test data needed.

Test Steps:

1. Log in as User B and navigate to User A's post.

2. Like the post using the "Like" button.

3. Comment on the post using the comment section.

4. Share the post using the "Share" option.

Postconditions:

- Verify User A receives notifications for each interaction (like, comment, share).

- Verify the post displays the like count, comments, and share count accurately.

Expected Result: Users can successfully interact with posts through likes, comments, and shares, and the content creator receives appropriate feedback.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Integration

Test Case Approach: Positive

Data - 5

Project name -

Social Media Platform

Project description -

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

Feature name -

Privacy Settings

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Privacy Settings

Test Case ID: PS-001

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

Test Case Description: This test case verifies that a user can successfully navigate to the privacy settings and change their profile visibility to Public, Private, or Restricted.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User is on their profile page.

Test Data: No test data needed.

Test Steps:

1. Navigate to the Privacy Settings page.

2. Select the "Public", "Private", or "Restricted" option for profile visibility.

3. Save the changes.

4. Verify that the selected privacy setting is reflected on the user's profile.

Postconditions:

- The user's profile visibility should reflect the selected option.

Expected Result: The user should be able to set their profile to any of the three privacy settings successfully.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: PS-002

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

Test Case Description: This test verifies that users can modify their existing privacy settings whenever they choose.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in.

- User has previously set their privacy settings.

Test Data: No test data needed

Test Steps:

1. Navigate to the Privacy Settings page.

2. Change one or more privacy settings.

3. Save the changes.

4. Verify that the new settings are reflected on the user's profile.

Postconditions:

- The user's profile should reflect the updated privacy settings.

Expected Result: The user should be able to change their privacy settings successfully at any time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: PS-003

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

Test Case Description: This test case checks if users can define the visibility (Public, Friends Only, Private) of each new post they create.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in.

- User is creating a new post.

Test Data: No test data needed

Test Steps:

1. Create a new post.

2. Select the desired visibility option (Public, Friends Only, Private) for the post.

3. Publish the post.

4. Verify that the post visibility is applied correctly based on the chosen option.

Postconditions:

- The new post should reflect the selected visibility setting.

Expected Result: The user should be able to control the visibility of their individual posts.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: PS-004

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

Test Case Description: This test case checks if the system applies the changed privacy settings instantly, restricting/granting access to the user's profile and content accordingly.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in.
- User has changed their privacy settings.

Test Data: No test data needed

Test Steps:

1. User A changes their privacy setting to "Private".
2. User B, who is not friends with User A, attempts to view User A's profile and posts.
3. Verify that User B cannot access User A's restricted content.
4. User A changes their privacy settings back to "Public".
5. User B refreshes User A's profile.
6. Verify that User B can now view User A's profile and posts.

Postconditions:

- The system should immediately reflect changes in privacy settings.

Expected Result: The system should enforce the new privacy settings immediately, restricting or granting access based on the user's configuration.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive, Negative

Test Case ID: PS-005

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

Test Case Description: This test verifies that the system sends a notification to the user upon successful modification of their privacy settings.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in.

- User is on the Privacy Settings page.

Test Data: No test data needed

Test Steps:

1. User makes a change to their privacy settings.

2. Save the changes.

3. Verify that a confirmation message appears on the screen.

4. Verify that the user receives an email or in-app notification confirming the privacy setting changes.

Postconditions:

- The user should be informed about changes made to their privacy settings.

Expected Result: The user should be notified about any successful changes to their privacy settings via on-screen message and email/in-app notification.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 6

Project name -

Social Media Platform

Project description -

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

Feature name -

Notifications

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Notifications Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
NOTIF-001	Verify that users receive notifications for likes, comments, and shares.	This test case verifies that the system generates and delivers notifications to users when their posts receive likes, comments, or shares from other users.	Notifications	High	- User A is logged in. - User B is logged in. - User A has at least one public post.	No test data needed	1. User B likes, comments on, and shares User A's post. 2. Observe User A's notification panel.	- Clear all notifications.	User A receives three separate notifications: one for the like, one for the comment, and one for the share.	Critical	High	Functional, System, Acceptance	Positive
NOTIF-002	Verify that users are notified of new friend requests and mentions.	This test case checks if users receive notifications when they get new friend requests and when other users mention them in posts or comments.	Notifications	High	- User A is logged in. - User B is logged in.	No test data needed	1. User B sends a friend request to User A. 2. User B creates a post mentioning User A using "@" symbol. 3. Observe User A's notification panel.	- User B cancels the friend request. - User B deletes the post mentioning User A.	User A receives two separate notifications: one for the friend request and one for the mention.	Critical	High	Functional, System, Acceptance	Positive
NOTIF-003	Verify that notifications are displayed in real-time.	This test case ensures that notifications appear promptly without any noticeable delay after the triggering action occurs.	Notifications	High	- User A is logged in. - User B is logged in.	No test data needed	1. User B likes User A's post. 2. Observe User A's notification panel for the notification.	- Clear all notifications.	The notification for the like appears on User A's notification panel instantly, without any significant delay.	Critical	High	Functional, Performance, Acceptance	Positive
NOTIF-004	Verify that users can customize notification preferences.	This test case examines the											

functionality allowing users to tailor their notification settings, choosing which types of notifications they want to receive. | Notifications, Settings | Medium | - User A is logged in. | No test data needed | 1. Navigate to User A's notification settings. 2. Toggle off notifications for "Likes" 3. Have User B like User A's post. 4. Observe User A's notification panel. | - Re-enable notifications for "Likes". | User A does not receive a notification for the like on their post after disabling "Likes" notifications. | Major | Medium | Functional, Usability, Acceptance | Negative |

| NOTIF-005 | Verify that notifications include a link to the relevant activity. | This test case checks if each notification provides a direct link to the specific post, comment, or profile related to the notification. | Notifications | Medium | - User A is logged in. - User B is logged in. | No test data needed | 1. User B comments on User A's post. 2. User A clicks on the received notification. | - Clear all notifications. | Clicking the notification redirects User A to the specific post where User B left the comment. | Major | Medium | Functional, Usability, Acceptance | Positive |

Data - 7

Project name -

Social Media Platform

Project description -

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

Feature name -

Search Functionality

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Search Functionality

Test Case ID: SEARCH-001

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

Test Case Description: This test case verifies that the search functionality correctly returns relevant results when a user enters a keyword in the search bar.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged into the platform.
- User is on a page with an active search bar.

Test Data:

- Keyword: "Technology"

Test Steps:

1. Navigate to the search bar.
2. Enter the keyword "Technology".
3. Press Enter or click the search icon.

Postconditions: None

Expected Result:

- The platform displays a list of results containing the keyword "Technology" in the post content, profile information, or associated hashtags.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SEARCH-002

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

Test Case Description: This test case validates that the search functionality accurately returns results for searches using hashtags.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged into the platform.

- User is on a page with an active search bar.

Test Data:

- Hashtag: "#Travel"

Test Steps:

1. Navigate to the search bar.

2. Enter the hashtag "#Travel".

3. Press Enter or click the search icon.

Postconditions: None

Expected Result:

- The platform displays a list of posts and profiles using the hashtag "#Travel".

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SEARCH-003

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

Test Case Description: This test case ensures the search functionality returns a variety of content types including profiles, posts, and topics related to the search query.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged into the platform.

- User is on a page with an active search bar.

Test Data:

- Search Term: "Photography"

Test Steps:

1. Navigate to the search bar.

2. Enter the search term "Photography".

3. Press Enter or click the search icon.

Postconditions: None

Expected Result:

- The results page displays a mix of profiles related to photography, posts containing photography content, and potentially suggested topics or communities focused on photography.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SEARCH-004

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

Test Case Description: This test case ensures users can refine their search results by filtering for specific content types like posts, profiles, photos, videos etc.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User has performed a search and received results.

Test Data: None

Test Steps:

1. Observe the search results page for available filter options.

2. Select a filter option, such as "Profiles" or "Videos".

Postconditions: None

Expected Result:

- The search results update dynamically to display only the content type selected in the filter.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SEARCH-005

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

Test Case Description: This test case verifies the search bar provides real-time suggestions based on the user's input, improving search efficiency.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged into the platform.

- User is on a page with an active search bar.

Test Data: None

Test Steps:

1. Navigate to the search bar.

2. Begin typing a keyword or hashtag.

Postconditions: None

Expected Result:

- A dropdown menu appears beneath the search bar, displaying relevant suggestions based on the entered characters. This could include matching keywords, hashtags, profiles, or topics.

Severity: Minor

Priority: Medium

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 Cases for Hashtags and Tagging Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
TC_HT_001	Verify that users can add hashtags to their posts.	This test case verifies the functionality of adding hashtags to posts and their visibility in the feed.	Hashtags and Tagging	High	- User is logged in. - User is creating a new post. - User has access to the hashtag symbol (#).	No test data needed	1. Navigate to the post creation screen. 2. Enter text for the post. 3. Add one or more hashtags within the post text (e.g., #travel, #photography). 4. Publish the post. 5. Verify that the hashtags are displayed correctly within the post in the feed.	No specific postconditions.	Hashtags are successfully added to the post and visible in the feed.

Blocker | High | Functional, UI | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| TC_HT_002 | Verify that users can tag other profiles, brands, or locations in their posts. | This test case checks if users can successfully tag other users, brands, or locations in their posts and if these tags are clickable. | Hashtags and Tagging | High | - User is logged in. - User is creating a new post. | No test data needed | 1. Navigate to the post creation screen. 2. Enter text for the post. 3. Type "@" followed by the username of another user, brand, or location. 4. Select the correct profile/page from the suggestions. 5. Publish the post. 6. Verify that the tag is displayed correctly and is clickable, leading to the tagged profile/page. | No specific postconditions. | Users are successfully tagged in the post and the tag links to their profile/page. | Blocker | High | Functional, UI | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| TC_HT_003 | Verify that hashtags and tags are clickable and lead to related content. | This test case ensures clicking on hashtags or tags in a post directs the user to the relevant content page. | Hashtags and Tagging | High | - User is logged in. - User is viewing a post with hashtags or tags. | No test data needed | 1. Navigate to a post containing hashtags and/or tags. 2. Click on a hashtag within the post. 3. Verify that the click redirects to a page displaying other posts with the same hashtag. 4. Go back to the original post. 5. Click on a tag within the post. 6. Verify that the click redirects to the profile/page of the tagged user, brand, or location. | No specific postconditions. | Clicking hashtags and tags redirects users to corresponding content pages. | Critical | High | Functional, UI | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| TC_HT_004 | Verify that users can search for content using hashtags. | This test case checks the functionality of the search bar in finding content based on specific hashtags. | Hashtags and Tagging | High | - User is logged in. - User is on the search page. | Test data: #Technology | 1. Navigate to the search page. 2. Enter a hashtag in the search bar (e.g., #Technology). 3. Initiate the search. 4. Verify that the search results page displays posts containing the searched hashtag. | No specific postconditions. | The search function displays relevant results for the entered hashtag. | Critical | High | Functional, Search Functionality | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| TC_HT_005 | Verify that the system suggests popular hashtags as users type. | This test case validates the system's ability to provide hashtag suggestions based on user input and trending topics. | Hashtags and Tagging | Medium | - User is logged in. - User is creating a new post. - User is typing in the post text area. | No test data needed | 1. Navigate to the post creation screen. 2. Start typing a word related to a potential hashtag (e.g., "travel"). 3. Observe if the system suggests relevant and popular hashtags (e.g., #travel, #travelphotography, #travelgram). 4. Verify that the suggestions change dynamically as the user continues typing. | No specific postconditions. | The system provides relevant and dynamic hashtag suggestions based on user input. | Minor | Medium | Functional, UI, Usability | Positive |

Data - 9

Project name -

Social Media Platform

Project description -

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

Feature name -

Emojis and Reactions

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Emojis and Reactions Feature

Test Case ID: TC_ER_001

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

Test Case Description: This test case verifies the functionality of reacting to posts using various emojis.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged into the platform.

- A post is available in the user's feed.

Test Data: No test data needed.

Test Steps:

1. Navigate to a post in the feed.

2. Hover over the reaction options (e.g., like button).

3. Select a desired emoji from the displayed list.

4. Verify that the selected emoji appears below the post as a user reaction.

5. Repeat steps 3-4 with different emojis.

Postconditions:

- User reactions are recorded and displayed correctly.

Expected Result: Users should be able to react to posts using a variety of emojis from the platform's selection.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TC_ER_002

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

Test Case Description: This test case verifies if users can successfully include emojis within their comments on posts and within direct messages.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged into the platform.

- User is viewing a post with the option to comment or is in a direct message conversation.

Test Data: No test data needed.

Test Steps:

1. Open the comment section of a post or a direct message conversation.

2. Tap on the designated emoji input field (if available) or use the device's keyboard to insert an emoji.

3. Type a comment or message containing the emoji.

4. Send the comment or message.

5. Verify that the emoji is displayed correctly within the sent message.

Postconditions:

- Emojis are successfully sent and displayed in comments and messages.

Expected Result: Emojis should be seamlessly integrated and displayed within user comments and messages.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TC_ER_003

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

Test Case Description: This test case verifies the functionality of displaying an accurate count of reactions received on a post.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

- User is logged into the platform.
- A post with existing reactions is available in the user's feed.

Test Data: No test data needed.

Test Steps:

1. Navigate to a post with existing reactions.
2. Observe the reaction display area below the post.
3. Verify that a numerical count is displayed next to each unique reaction emoji.
4. Verify that the count accurately reflects the total number of each type of reaction given to the post.

Postconditions:

- Reaction counts are updated in real-time and displayed correctly.

Expected Result: Users should be able to view a clear and accurate count of each type of reaction received on a post.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TC_ER_004

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

Test Case Description: This test case ensures the platform supports a diverse and up-to-date selection of emojis for user expression.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

- User is logged into the platform.

- User is in an area where they can input and send emojis (e.g., comments, messages).

Test Data: List of commonly used emojis across different categories (emotions, objects, symbols, etc.).

Test Steps:

1. Attempt to input and send a variety of emojis from the provided test data list.

2. Verify that the platform successfully displays the emojis as intended.

3. Check for any rendering issues or missing emojis.

Postconditions:

- A wide range of emojis are supported and rendered correctly.

Expected Result: The platform should support and accurately display a wide range of commonly used emojis, ensuring a comprehensive and inclusive user experience.

Severity: Minor

Priority: Medium

Type of Testing: Compatibility Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_ER_005

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

Test Case Description: This test case verifies the user's ability to retract or modify their previous reactions to posts.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged into the platform.
- User has previously reacted to a post.

Test Data: No test data needed.

Test Steps:

1. Navigate to a post where the user has previously reacted.
2. Locate the user's existing reaction below the post.
3. Click or tap on the existing reaction.
4. Verify that the reaction is either removed entirely or that options appear allowing the user to select a different reaction.
5. If changing the reaction, select a new emoji and verify that it replaces the previous one.

Postconditions:

- User reactions can be easily changed or removed without issue.

Expected Result: Users should have full control over their reactions, enabling them to change or remove them based on their preference.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 10

Project name -

Social Media Platform

Project description -

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

Feature name -

Live Streaming

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Live Streaming Feature

Test Case ID: LST-001

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

Test Case Description: This test case verifies the basic functionality of starting and stopping a live stream. It ensures that users can initiate a live stream and terminate it successfully.

Test Suite: Live Streaming Functionality

Test Priority: High

Preconditions:

- User is logged into the social media platform.

- User has necessary permissions to start a live stream.

- User's device has a stable internet connection.

Test Data: No test data needed

Test Steps:

1. Navigate to the live stream creation section.

2. Select "Start Live Stream".

3. Verify that the live stream starts successfully and the user interface displays the active streaming status.

4. Select "Stop Live Stream".

5. Verify that the live stream stops successfully and the user interface reflects the inactive streaming status.

Postconditions:

- The live stream is terminated.

- The recorded stream (if saved) is accessible.

- User interface returns to pre-stream state.

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

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: LST-002

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

Test Case Description: This test case ensures that users have control over the privacy settings of their live streams. It verifies that users can choose to make their live streams public or restricted to their followers only.

Test Suite: Live Streaming Functionality

Test Priority: High

Preconditions:

- User is logged into the social media platform.

- User is in the process of initiating a live stream.

Test Data: No test data needed

Test Steps:

1. Navigate to the live stream creation section.

2. Locate the privacy settings for the live stream.

3. Select "Public" and verify that the setting is applied correctly.

4. Select "Followers Only" and verify that the setting is applied correctly.

5. Start the live stream and verify the chosen privacy setting is reflected in the live stream's visibility.

Postconditions: None

Expected Result: Users should be able to successfully set their live streams as public or limited to followers.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Security Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: LST-003

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

Test Case Description: This test case tests the interaction features available to viewers during a live stream. It verifies that viewers can post comments and use reactions in real-time.

Test Suite: Live Streaming Interaction

Test Priority: High

Preconditions:

- User is logged into the social media platform.
- User is viewing an active live stream.

Test Data: Sample comment: "Great stream!"

Test Steps:

1. Locate the comment section of the live stream.
2. Type a comment and submit it.
3. Verify that the comment appears in the live chat.
4. Select a reaction emoji.
5. Verify that the reaction is displayed in the live stream.

Postconditions: None

Expected Result: Viewers should be able to successfully post comments and use reactions during a live stream.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: LST-004

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

Test Case Description: This test case checks the live stream interface for essential information like viewer count and real-time comments. This ensures viewers and streamers are aware of audience engagement.

Test Suite: Live Streaming Interface

Test Priority: Medium

Preconditions:

- User is logged into the social media platform.

- User is viewing an active live stream.

Test Data: No test data needed

Test Steps:

1. Access an active live stream.

2. Verify that the number of viewers is displayed and updates in real-time as viewers join or leave.

3. Observe the live comment section.

4. Verify that new comments from viewers appear in real-time during the live stream.

Postconditions: None

Expected Result: The live stream interface should correctly display the viewer count and live comments.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: LST-005

Test Case Title: Verify that users can save their live streams for later viewing.

Test Case Description: This test case examines the functionality of saving a live stream after it ends. It ensures users can access and view their past live streams.

Test Suite: Live Stream Saving & Archiving

Test Priority: Medium

Preconditions:

- User is logged into the social media platform.

- User has just finished a live stream.

Test Data: No test data needed

Test Steps:

1. After ending the live stream, locate the option to save the stream.

2. Select the "Save" option.

3. Verify that the platform confirms the live stream is being saved.

4. Navigate to the user's video library or profile section.

5. Verify that the saved live stream is accessible for later viewing.

Postconditions: Saved live stream is accessible in the user's video library or profile.

Expected Result: Users should be able to save their live streams and access them for later viewing.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Data - 11

Project name -

Social Media Platform

Project description -

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

Feature name -

Stories

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Stories Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
ST-001	Verify that users can create stories with photos, videos, and text overlays.	This test case verifies the core functionality of story creation, ensuring users can add different media types and customize them with text.	Stories	High	- User account created and logged in. - Camera and microphone permissions granted (if applicable). - Active internet connection.	- Sample photo - Sample video - Text overlay content	1. Navigate to the story creation screen. 2. Select the option to add a photo from the gallery. 3. Add the sample photo. 4. Select the option to add text overlay and input the sample text. 5. Repeat steps 2-4 for video. 6. Post the story.	- Story is visible in user's profile and to followers. - Used media is deleted from the device's temporary storage.	Story is successfully created and displayed with both photo, video, and text overlay.	Blocker	High	Functional, Acceptance	Positive

| ST-002 | Verify that stories disappear after 24 hours. | This test case validates the temporary nature of stories, confirming they are automatically removed after the defined time limit. | Stories | High | - User account created and logged in. - Story created within the last 24 hours. | No test data needed | 1. Note the time of story creation. 2. Wait for 24 hours. 3. Check user's profile and followers' feeds for the story's visibility. | No specific postconditions. | The created story is no longer visible after 24 hours. | Critical | High | Functional, Time-dependent, Acceptance | Positive |

| ST-003 | Verify that users can see who viewed their stories. | This test case confirms users can access the viewer list for their stories, ensuring transparency and engagement tracking. | Stories | Medium | - User account created and logged in. - Story created and at least one view registered. | No test data needed | 1. Navigate to the user's own story. 2. Open the viewer list. 3. Verify the displayed usernames match the actual viewers. | No specific postconditions. | The viewer list accurately reflects all users who have seen the story. | Minor | Medium | Functional, Acceptance | Positive |

| ST-004 | Verify that stories can be shared publicly or with selected followers. | This test case ensures users have control over their story's visibility, allowing them to choose between public sharing and limiting access to specific followers. | Stories | High | - User account created and logged in. - List of followers established. | No test data needed | 1. Navigate to the story creation screen. 2. Locate and adjust the privacy settings for story visibility. 3. Create a story with "Public" setting. 4. Create another story with "Selected Followers" setting, choosing specific followers. 5. Verify the visibility of both stories from different accounts (public, follower, and non-follower). | No specific postconditions. | Story visibility aligns with the selected privacy setting (Public or Selected Followers). | Critical | High | Functional, Security, Acceptance | Positive, Negative |

| ST-005 | Verify that users can add stickers and filters to their stories. | This test case confirms the availability and functionality of story enhancements, ensuring users can personalize their content with stickers and visual filters. | Stories | Medium | - User account created and logged in. | No test data needed | 1. Navigate to the story creation screen. 2. Add a photo or video. 3. Access the stickers menu and attempt to add various stickers. 4. Access the filter options and apply different filters. 5. Post the story and verify the applied enhancements. | No specific postconditions. | Stickers and filters are successfully added to the story and are visible as intended.

| Minor | Medium | Functional, Usability, Acceptance | Positive |

Data - 12

Project name -

Social Media Platform

Project description -

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

Feature name -

Analytics and Insights

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Analytics and Insights

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
AAI-001	Verify that users can view data on follower demographics	This test case verifies that users can access and view detailed information about their follower demographics.	Analytics and Insights	High	- User account created <br> - User has followers	No test data needed	1. Log in as a user. <br> 2. Navigate to the Analytics section. <br> 3. Select the Follower Demographics view.	None	The user should be able to see a breakdown of their followers based on age, gender, location, and other relevant demographic information.	Critical	High	Functional, Usability	Positive
AAI-002	Verify that post reach and engagement metrics are available	This test case verifies that users can access data on the reach and engagement of their posts, including likes, comments, shares, and views.	Analytics and Insights	High	- User account created <br> - User has published posts	No test data needed	1. Log in as a user. <br> 2. Navigate to the Analytics section. <br> 3. Select a specific post to view its analytics.	None	The user should be able to view metrics such as the number of users reached, likes, comments, shares, and potentially other engagement metrics like saves or link clicks.	Critical	High	Functional	Positive
AAI-003	Verify that users can track audience growth trends	This test case checks the functionality of tracking follower growth over different time periods.	Analytics and Insights	Medium	- User account created	No test data needed	1. Log in as a user. <br> 2. Navigate to the Analytics section. <br> 3. Select the Audience Growth view. <br> 4. Choose different time ranges (e.g., past week, past month, past year).	None	The user should see a graphical representation or data table showing their follower growth trends over the selected time periods.	Major	Medium	Functional, Usability	Positive
AAI-004	Verify that analytics are displayed in a user-friendly format	This test case assesses the clarity and usability of the analytics dashboard, ensuring ease of understanding and navigation.	Analytics and Insights

Medium | - User account created | No test data needed | 1. Log in as a user. <br> 2. Navigate to the Analytics section. <br> 3. Explore different analytics views and data representations. | None | The analytics dashboard should present data in a clear, concise, and visually appealing way, using charts, graphs, and tables that are easy to understand and navigate. | Minor | Medium | Usability | Positive |

| AAI-005 | Verify that data can be filtered by time range and content type | This test case verifies the filtering functionality of the analytics data. | Analytics and Insights | Medium | - User account created <br> - User has different types of posts (text, image, video) | No test data needed | 1. Log in as a user. <br> 2. Navigate to the Analytics section. <br> 3. Apply filters for different time ranges (e.g., last week, last month). <br> 4. Apply filters for different content types (e.g., text posts, images, videos). | None | The displayed analytics data should be filtered accurately based on the selected time range and content type. | Major | Medium | Functional | Positive |

Data - 13

Project name -

Social Media Platform

Project description -

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

Feature name -

Events and Groups

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

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

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
EG-001	Verify that users can create and join events.	This test case verifies the functionality of creating and joining events on the platform.	Events and Groups	High	- User is logged in.<br> - User has an active account.<br>- User navigates to the "Events" section.	Event Name: Test Event<br>Event Description: This is a test event.<br>Event Date/Time: Future date and time.<br>Event Location: Online/Offline Location	1. Navigate to the "Events" section.<br>2. Click on "Create Event" button.<br>3. Fill in the event details (Name, Description, Date/Time, Location).<br>4. Click on "Create" button.<br>5. Verify the event is created and displayed correctly.<br>6. Search for the created event and click on "Join Event."	- Event is created successfully.<br>- User is added to the event attendee list.	User should be able to successfully create an event and join the created event.	Critical	High	Functional, Acceptance	Positive
EG-002	Verify that users can join interest-based groups.	This test case checks the functionality allowing users to find and join groups based on their interests.	Events and Groups	High	- User is logged in.<br>-

User has an active account.<br>- User navigates to the "Groups" section. | Group Name: Test Group<br>Group Description: This is a test group for [Interest]. | 1. Navigate to the "Groups" section.<br>2. Use the search bar to find groups related to a specific interest (e.g., "Photography").<br>3. Select a group from the search results.<br>4. Click on the "Join Group" button.<br>5. Verify that the user is added to the group member list. | - User is added to the selected group.<br>- User can access group content. | User should be able to successfully find and join a group based on their interest. | Critical | High | Functional, Acceptance | Positive |

| EG-003 | Verify that group discussions can be organized by topic. | This test case ensures that group admins or members can categorize discussions within a group using topics. | Events and Groups | Medium | - User is logged in.<br>- User is a member/admin of a group.<br>- User navigates to a group's "Discussions" section. | Topic Name: Test Topic<br>Discussion Title: Test Discussion | 1. Navigate to the "Discussions" section of a joined group.<br>2. Click on "Create New Topic" (if available to the user role).<br>3. Enter a relevant Topic Name (e.g., "Upcoming Events").<br>4. Create a new discussion and select the newly created topic.<br>5. Verify the discussion is correctly categorized under the chosen topic. | - New topic is successfully created.<br>- Discussion is displayed under the correct topic. | Users should be able to organize and find discussions within a group using topics. | Major | Medium | Functional, Usability | Positive |

| EG-004 | Verify that event details include date, time, location, and description. | This test case confirms that all essential information regarding an event is displayed accurately. | Events and Groups | High | - User is logged in.<br>- User can view an event (created or public). | N/A | 1. Navigate to the "Events" section.<br>2. Select an event from the list of available events.<br>3. Verify that the event details page displays the following information accurately:<br>   - Event Name<br>   - Event Description<br>   - Event Date and Time<br>   - Event Location (if applicable) | N/A | All event details should be present and accurately displayed on the event page. | Critical | High | Functional, Usability | Positive |

| EG-005 | Verify that users can RSVP to events and see the list of attendees. | This test case checks the functionality of responding to event invitations and viewing the attendee list. | Events and Groups | High | - User is logged in.<br>- User can view an event with RSVP option enabled. | N/A | 1. Navigate to the "Events" section.<br>2. Select an event with RSVP enabled.<br>3. Click on the "RSVP" button.<br>4. Choose an

RSVP option (e.g., "Going," "Interested," "Not Going").<br>5. Verify that the user's RSVP is registered correctly.<br>6. Check if the event page displays a list of attendees with their respective RSVP responses. | - User's RSVP status is updated.<br>- Attendee list reflects accurate RSVP responses. | Users should be able to RSVP to events and view the list of attendees with their respective RSVP responses. | Critical | High | Functional, Acceptance | Positive |

Data - 14

Project name -

Social Media Platform

Project description -

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

Feature name -

Monetization Options

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Monetization Options:

Test Case ID: TC_Monetization_001

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

Test Case Description: This test case verifies that eligible users can successfully navigate to the monetization settings and submit an application for monetization options.

Test Suite: Monetization

Test Priority: High

Preconditions:

- User account created and logged in.

- User meets minimum eligibility criteria for monetization (e.g., follower count, content views).

Test Data: No test data needed.

Test Steps:

1. Navigate to the user profile settings.

2. Locate and select the "Monetization" or equivalent option.

3. Verify that the application form for monetization options is accessible.

4. Fill out the application form with valid information.

5. Submit the application.

Postconditions:

- Confirmation message displayed upon successful application submission.

- Application status trackable in user profile.

Expected Result: The user should be able to successfully apply for monetization options.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_Monetization_002

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

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

Test Suite: Monetization

Test Priority: High

Preconditions:

- User is logged in.
- User has successfully applied for monetization.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Monetization" section within the user profile settings.
2. Check for the availability of "Advertising Revenue" as a monetization option.
3. Verify the presence of information and settings related to enabling and managing advertising revenue.
4. Check for the availability of "Sponsored Content Partnerships" as a monetization option.
5. Verify the presence of information and settings related to partnering for sponsored content.

Postconditions: None

Expected Result: The platform should clearly list "Advertising Revenue" and "Sponsored Content Partnerships" as available monetization options.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_Monetization_003

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

Test Case Description: This test case validates the payment process for users who have earned revenue through monetization.

Test Suite: Monetization

Test Priority: High

Preconditions:

- User is logged in.

- User has earned revenue through monetization options.

- User has configured payment settings.

Test Data: Valid payment information (e.g., bank account details, payment processor account).

Test Steps:

1. Navigate to the "Earnings" or "Payments" section in the user profile.

2. Initiate a payment request for the earned amount.

3. Enter or select the preferred payment method.

4. Confirm the payment request.

Postconditions:

- Payment request confirmation displayed.

- Payment status trackable in the user's transaction history.

Expected Result: The platform should successfully process the payment request and transfer the earned revenue to the user's designated payment method.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_Monetization_004

Test Case Title: Verify that detailed earnings reports are available to users.

Test Case Description: This test case checks that users have access to comprehensive reports detailing their earnings from monetization.

Test Suite: Monetization

Test Priority: High

Preconditions:

- User is logged in.

- User has earned revenue through monetization options.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Earnings" or "Reports" section within the user profile.

2. Verify that earnings reports are available for different time periods (e.g., daily, weekly, monthly).

3. Check that the reports display detailed breakdowns of earnings sources (e.g., advertising, sponsored content).

4. Ensure reports include information like total earnings, payment deductions, and net earnings.

Postconditions: None

Expected Result: Users should be able to access detailed and accurate reports that provide comprehensive information on their earnings.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_Monetization_005

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

Test Case Description: This test case checks that the platform clearly outlines the criteria users need to meet for monetization eligibility.

Test Suite: Monetization

Test Priority: High

Preconditions: No preconditions

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Monetization" section or related help documentation.
2. Verify that the eligibility criteria for monetization are explicitly stated.
3. Ensure the requirements are specific, measurable, attainable, relevant, and time-bound (SMART).
4. Check for clarity in language and easy comprehension of the eligibility criteria.

Postconditions: None

Expected Result: The platform should clearly and unambiguously present all monetization eligibility requirements to users.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing, Acceptance Testing

Test Case Approach: Positive

Data - 15

Project name -

Social Media Platform

Project description -

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

Feature name -

Explore or Discover Tab

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Explore/Discover Tab - Social Media Platform

Test Case ID: EXP-001

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

Test Case Description: This test case verifies that the Explore tab populates with content deemed trending or popular based on platform algorithms.

Test Suite: Explore Tab Functionality

Test Priority: High

Preconditions:

* User is logged into the platform.

* User navigates to the Explore tab.

Test Data: No test data needed.

Test Steps:

1. Observe the content displayed on the Explore tab.

2. Verify that the displayed content consists of posts, accounts, or topics with high engagement (likes, comments, shares).

3. Compare the displayed content with trending topics or popular content from other sections of the platform.

Postconditions: None

Expected Result: The Explore tab should prominently feature content with significantly higher engagement metrics compared to average content, indicating trending or popular nature.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EXP-002

Test Case Title: Verify that content recommendations on the Explore tab are personalized based on user interests.

Test Case Description: This test case validates the personalization algorithm by checking if the displayed content aligns with the user's interaction history and declared interests.

Test Suite: Explore Tab Functionality

Test Priority: High

Preconditions:

* User is logged into the platform.

* User has a history of interactions (likes, comments, shares, follows) with specific content categories or accounts.

* User has defined interests in their profile settings.

Test Data: No test data needed.

Test Steps:

1. Navigate to the Explore tab.

2. Analyze the recommended content categories and accounts.

3. Verify if the recommendations align with the user's past interactions and defined interests.

Postconditions: None

Expected Result: The Explore tab should prioritize content recommendations that match the user's established interaction patterns and specified interests.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Data-Driven Testing

Test Case Approach: Positive

Test Case ID: EXP-003

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

Test Case Description: This test case ensures the functionality of content filtering options within the Explore tab, allowing users to refine their discovery experience.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

* User is logged into the platform.

* User is on the Explore tab.

Test Data: No test data needed.

Test Steps:

1. Locate and access the content filtering options (category or topic selection).

2. Choose a specific category or topic filter.

3. Verify that the displayed content dynamically updates to show only content matching the selected filter.

Postconditions: None

Expected Result: The Explore tab content should successfully filter, displaying content exclusively from the chosen category or topic.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EXP-004

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

Test Case Description: This test case ensures that the Explore tab dynamically refreshes with new trending or popular content at reasonable intervals.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

* User is logged into the platform.

* User is on the Explore tab.

Test Data: No test data needed.

Test Steps:

1. Note the current time and the displayed content on the Explore tab.

2. Wait for a predetermined interval (e.g., 30 minutes to an hour).

3. Return to the Explore tab and observe if there are any new content updates.

Postconditions: None

Expected Result: The Explore tab should reflect new trending or popular content over time, indicating a dynamic refresh mechanism.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EXP-005

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

Test Case Description: This test case confirms the user's ability to interact with and retain content discovered on the Explore tab through saving or following options.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

* User is logged into the platform.

* User is on the Explore tab.

Test Data: No test data needed.

Test Steps:

1. Identify a piece of content (post, account, or topic) on the Explore tab.

2. Locate and utilize the "Save" or "Follow" action associated with the content.

3. Verify that the content is successfully saved or followed, either through on-screen confirmation or by checking the appropriate user profile section.

Postconditions:

* Unsave or unfollow the content used in testing to maintain data integrity.

Expected Result: Users should be able to interact with content on the Explore tab and retain access to it for future reference.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 16

Project name -

Social Media Platform

Project description -

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

Feature name -

Accessibility Features

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Accessibility Features on Social Media Platform

Test Case ID: AC-001

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

Test Case Description: This test case aims to confirm that all images on the platform have appropriate alt text, enabling users with screen readers to understand the context and content of the images.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:
* User is logged into the platform.
* User is using a screen reader.

Test Data: No test data needed.

Test Steps:
1. Navigate to different sections of the platform (e.g., newsfeed, profiles, messages).
2. Use the screen reader to access images displayed in these sections.
3. Verify that the screen reader announces meaningful and descriptive alt text for each image.
4. Check if the alt text accurately reflects the image content and context.

Postconditions: None

Expected Result: The screen reader should announce clear and descriptive alt text for all images, enabling users to understand the visual content.

Severity: Major

Priority: High

Type of Testing: Accessibility Testing

Test Case Approach: Positive

Test Case ID: AC-002

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

Test Case Description: This test case verifies the platform's compatibility with popular screen reader software, ensuring users with visual impairments can navigate and interact with all features effectively.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:
* User has a screen reader installed (e.g., NVDA, JAWS).
* Screen reader is active and running.

Test Data: List of common screen reader software.

Test Steps:

1. Open the platform using different browsers (e.g., Chrome, Firefox, Edge).
2. Navigate through various platform functionalities (e.g., posting, commenting, profile editing) using the chosen screen reader.
3. Observe if the screen reader accurately announces all elements, controls, and content on the platform.
4. Check for any inconsistencies or errors in screen reader feedback.

Postconditions: None

Expected Result: The screen reader should accurately announce all platform elements and functionalities, allowing for seamless navigation and interaction.

Severity: Blocker

Priority: High

Type of Testing: Compatibility Testing, Accessibility Testing

Test Case Approach: Positive

Test Case ID: AC-003

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

Test Case Description: This test verifies the platform's keyboard navigation functionality, enabling users who cannot use a mouse to navigate through the platform's functionalities.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

* User is logged into the platform.

Test Data: No test data needed.

Test Steps:

1. Attempt to navigate through various sections of the platform (e.g., newsfeed, profiles, settings) using only keyboard controls (e.g., tab, arrows keys, enter).
2. Test core functionalities like posting, commenting, liking, and sharing using keyboard shortcuts.
3. Verify that focus is clearly indicated visually for each interactive element.

Postconditions: None

Expected Result: All platform sections and functionalities should be accessible and navigable using only the keyboard.

Severity: Major

Priority: High

Type of Testing: Usability Testing, Accessibility Testing

Test Case Approach: Positive

Test Case ID: AC-004

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

Test Case Description: This test case checks if the platform offers sufficient color contrast and adjustable font sizes to cater to users with visual impairments.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:
* User is logged into the platform.

Test Data: No test data needed.

Test Steps:
1. Navigate to the platform's accessibility settings or display options.
2. Verify options are available to adjust:
    * Text size (e.g., small, medium, large).
    * Color contrast (e.g., light mode, dark mode, high contrast mode).
3. Apply different settings and observe the changes in the platform's visual appearance.
4. Use a color contrast checker tool to ensure sufficient contrast ratios between text and background colors in different modes.

Postconditions: None

Expected Result: The platform should provide options to adjust text size and color contrast according to user preferences, meeting accessibility standards.

Severity: Major

Priority: High

Type of Testing: Usability Testing, Accessibility Testing

Test Case Approach: Positive

Test Case ID: AC-005

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

Test Case Description: This test case ensures that comprehensive documentation exists to guide users on accessing and utilizing the platform's accessibility features.

Test Suite: Documentation

Test Priority: Medium

Preconditions:
* Access to the platform's help center or support documentation.

Test Data: No test data needed.

Test Steps:

1. Locate the section on accessibility features within the platform's documentation.

2. Verify that the documentation covers:

    * Available accessibility features.

    * How to enable and customize these features.

    * Keyboard shortcuts.

    * Contact information for accessibility support.

3. Review the clarity, completeness, and accuracy of the provided information.

Postconditions: None

Expected Result: The platform documentation should offer clear and comprehensive information on accessing and utilizing all available accessibility features.

Severity: Minor

Priority: Medium

Type of Testing: Documentation Testing

Test Case Approach: Positive

Data - 17

Project name -

Social Media Platform

Project description -

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

Feature name -

Safety and Moderation Tools

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Safety and Moderation Tools:

Test Case ID: TC_SM_001

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

Test Case Description: This test case aims to verify the effectiveness of the content moderation algorithms in identifying and filtering various types of inappropriate content, including text, images, and videos. This includes testing against a pre-defined list of offensive keywords, phrases, and potentially harmful imagery.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

- The platform is up and running.
- A user account is available.
- A list of prohibited content is defined.

Test Data: Various text posts, images, and videos containing inappropriate content as defined by the platform's guidelines.

Test Steps:

1. Log in to the platform as the test user.
2. Attempt to post various test data items (text, images, videos) containing inappropriate content.
3. Observe the system's response.
4. Verify that the content is successfully flagged and blocked from being posted.
5. Check for appropriate error messages or warnings displayed to the user.

Postconditions:

- The inappropriate content is not published on the platform.
- The user might receive a warning based on the platform's policy.

Expected Result: The content moderation algorithms successfully detect and block all predefined instances of inappropriate content.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Negative

Test Case ID: TC_SM_002

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

Test Case Description: This test case ensures that users can easily find and utilize the reporting mechanism for harmful or inappropriate content. It involves testing different reporting paths and verifying the platform's response to submitted reports.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

- The platform is up and running.

- A user account is available.

- Content to report exists (can be pre-populated).

Test Data: No test data needed

Test Steps:

1. Log in to the platform as the test user.

2. Navigate to a piece of content that requires reporting.

3. Locate and utilize the "Report" functionality.

4. Select the appropriate reason for reporting from the provided options.

5. Submit the report.

6. Optionally, check for a confirmation message or notification.

Postconditions:

- The report is successfully submitted.

- The platform takes appropriate action based on its policies.

Expected Result: The reporting mechanism functions smoothly, allowing users to report content without any hindrance.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_SM_003

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

Test Case Description: This test verifies the functionality of the user blocking feature, ensuring that users can block other users and that the blocking functionality works as expected, restricting interactions between the blocker and the blocked user.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

- The platform is up and running.
- Two separate user accounts are available.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as the first user.
2. Navigate to the profile of the second user.
3. Locate and utilize the "Block User" functionality.
4. Confirm the blocking action.
5. Attempt to interact with the blocked user's content and profile.

Postconditions:

- The blocked user no longer appears in search results or recommendations for the first user.
- Interaction with the blocked user's content is restricted.

Expected Result: The blocking function successfully prevents any interaction between the two users.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Negative Testing

Test Case Approach: Positive

Test Case ID: TC_SM_004

Test Case Title: Verify that comment filters prevent offensive language.

Test Case Description: This test case ensures that the comment filtering system effectively identifies and blocks or flags comments containing offensive language, based on a predefined list of inappropriate words and phrases.

Test Suite: Safety and Moderation

Test Priority: Medium

Preconditions:

- The platform is up and running.
- A user account is available.
- A list of offensive words and phrases is predefined.

Test Data: A list of comments containing offensive language.

Test Steps:

1. Log in to the platform as the test user.
2. Navigate to a post that allows comments.
3. Attempt to post comments containing words and phrases from the predefined list.
4. Observe the system's response to each comment.
5. Verify that the system successfully flags or blocks the offensive comments.

Postconditions:

- Offensive comments are not published without moderation.
- Users may receive warnings for attempting to post offensive language.

Expected Result: The comment filter effectively prevents the posting of offensive language in the comments section.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Negative

Test Case ID: TC_SM_005

Test Case Title: Verify that community guidelines are enforced consistently.

Test Case Description: This test assesses the platform's consistency in enforcing its community guidelines across different scenarios. This involves testing various content types and user actions to ensure uniform application of the guidelines.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

- The platform is up and running.
- Access to multiple user accounts with varying levels of activity is available.
- A clear understanding of the platform's community guidelines is established.

Test Data: Various types of content and user actions that might violate community guidelines (e.g., hate speech, spam, harassment).

Test Steps:

1. Utilizing different user accounts, engage in actions and post content that pushes the boundaries of the community guidelines.
2. Monitor the platform's response to each action and piece of content.
3. Document the outcomes of each test case, noting whether actions were flagged, removed, or resulted in warnings or account suspensions.
4. Analyze the results to identify any inconsistencies in the application of the community guidelines.

Postconditions:

- Documentation of inconsistencies is compiled for review and potential platform improvements.

Expected Result: The platform demonstrates consistency in applying its community guidelines across different user actions and content types.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Exploratory Testing

Test Case Approach: Positive, Negative, Destructive

Data - 18

Project name -

Social Media Platform

Project description -

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

Feature name -

Multi-Language Support

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Multi-Language Support

Test Case ID: MLS-001

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

Test Case Description: This test case verifies that users can successfully change the platform's interface language to their preferred language from a list of available options.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

* User is logged into the platform.

* User navigates to the 'Settings' or 'Preferences' section.

Test Data: No test data needed.

Test Steps:

1. Navigate to the language settings section.

2. Click on the language dropdown menu.

3. Select a different language from the available options.

4. Save the changes.

Postconditions:

* Verify that the interface language has changed to the selected language.

* Verify that this change persists across different pages and sessions.

Expected Result: The user interface should successfully switch to the selected language, and this change should be reflected consistently across the platform.

Severity: Critical

Priority: High

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

Test Case Approach: Positive

Test Case ID: MLS-002

Test Case Title: Verify that the platform offers content translation features.

Test Case Description: This test case verifies that the platform provides users with the ability to translate posts and messages into different languages.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

* User is logged into the platform.

* User is viewing a post or message in a language different from their preferred language.

Test Data: A post or message in a foreign language.

Test Steps:

1. Navigate to a post or message in a different language.

2. Locate and click on the "Translate" button or option.

3. Observe if the content is translated into the user's preferred interface language.

Postconditions: None.

Expected Result: The post or message content should be accurately translated into the user's preferred interface language.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: MLS-003

Test Case Title: Verify that posts and messages can be translated into different languages.

Test Case Description: This test case checks that the platform allows users to translate posts or messages into a variety of languages besides their preferred interface language.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

* User is logged into the platform.

* User is viewing a post or message they want to translate.

Test Data: A post or message in one language and a selection of several different target languages.

Test Steps:

1. Open a post or message.

2. Locate and click on the ?Translate? button or option.

3. Select a target language from the list of available languages.

4. Observe if the post or message content is translated into the selected target language.

Postconditions: None.

Expected Result: The content should be translated accurately into the selected target language, demonstrating the platform's ability to handle various language translations.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: MLS-004

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

Test Case Description: This test case verifies that once a user sets their preferred language, the platform remembers this preference across different sessions and applies it consistently to all features.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

* User is logged into the platform.

Test Data: No test data needed.

Test Steps:

1. Navigate to the language settings and select a preferred language.

2. Log out of the platform.

3. Log back into the platform.

4. Navigate to different sections of the platform, including features like messaging, newsfeed, and profile settings.

Postconditions: None.

Expected Result: The platform should display all features and content in the selected preferred language throughout the new session.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: MLS-005

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

Test Case Description: This test case verifies that the platform supports a significant number of languages to cater to a diverse user base.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

* User navigates to the language settings section.

Test Data: No test data needed.

Test Steps:

1. Open the language selection dropdown menu.

2. Count the number of languages available.

3. Observe the variety and representation of languages offered.

Postconditions: Document the supported languages for future reference and potential expansion.

Expected Result: The platform should offer a substantial list of languages, ideally representing major languages spoken globally. The more languages offered, the better the platform's accessibility to a wider audience.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 19

Project name -

Social Media Platform

Project description -

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

Feature name -

Cross-Platform Integration

Feature description -

Cross-platform integration allows users to connect and share content across different social media platforms and external services. This includes features like single sign-on (SSO) capabilities, cross-posting of content, integration with external media sources, and the ability to share platform content on other social networks, enhancing user convenience and content reach.

Userstory title -

As a user, I want cross-platform integration so that I can connect and share content across multiple platforms.

Acceptance criteria -

['Users can link their accounts with other social media platforms.', 'Content can be cross-posted to other platforms.', 'Users can sign in using single sign-on (SSO).', 'Integration with external media sources is available.', 'Content sharing across platforms is seamless.']

Generated Test Cases:

## Test Cases for Cross-Platform Integration:

Test Case ID: CPI-001

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

Test Case Description: This test case verifies the functionality of linking user accounts with other social media platforms such as Facebook, Twitter, and Instagram. It checks if the linking process is successful, authorization is handled correctly, and the user's profile reflects the linked accounts.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is logged into their social media platform account.

- User has active accounts on other social media platforms (e.g., Facebook, Twitter).

Test Data: None

Test Steps:

1. Navigate to the user profile settings.

2. Locate the "Linked Accounts" or similar section.

3. Select the desired social media platform to link.

4. Follow the on-screen prompts to authorize the connection.

5. Verify that the linked account appears in the "Linked Accounts" section.

Postconditions:

- The user's account should be successfully linked to the selected platform.

- The user should be able to share content directly to the linked platform.

Expected Result: The user should be able to successfully link their account with other social media platforms.

Severity: Critical

Priority: High

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: CPI-002

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

Test Case Description: This test case checks the functionality of cross-posting content from the platform to linked social media accounts. It verifies if the content is successfully shared on other platforms with accurate formatting and linked back to the original post.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is logged into their social media platform account.

- User has linked their account with at least one other social media platform.

Test Data: A sample text post with an optional image or video.

Test Steps:

1. Create a new post on the platform.

2. Select the option to cross-post to linked accounts.

3. Choose the desired platform(s) to share the post on.

4. Publish the post.

5. Verify that the post appears on the chosen linked platform(s).

Postconditions:

- The post should be successfully shared on the selected linked platforms.

- The shared post should include accurate formatting and link back to the original post.

Expected Result: Content should be successfully cross-posted to other platforms with correct formatting and linking.

Severity: Critical

Priority: High

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: CPI-003

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

Test Case Description: This test case verifies the single sign-on (SSO) functionality for user login. It ensures that users can successfully log in to the platform using credentials from supported third-party platforms like Google or Facebook.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- SSO is enabled for the platform.

- User has an active account with a supported third-party platform.

Test Data: None

Test Steps:

1. Navigate to the platform's login page.

2. Select the option to sign in using a supported third-party platform (e.g., Google, Facebook).

3. Enter the login credentials for the selected platform.

4. Authorize the platform to access the necessary information.

5. Verify that the user is successfully logged into the platform.

Postconditions:

- User should be logged in and redirected to the platform's main page.
- The user's profile should be populated with information from the third-party account (if applicable).

Expected Result: Users should be able to successfully log in using single sign-on with supported third-party platforms.

Severity: Critical

Priority: High

Type of Testing: Functional, Security

Test Case Approach: Positive

Test Case ID: CPI-004

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

Test Case Description: This test case checks the integration with external media sources such as YouTube, Spotify, or news feeds. It verifies if users can successfully embed content from these sources into their posts and if the embedded content displays correctly.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

- User is logged into their social media platform account.

Test Data: Valid URLs from supported external media sources (YouTube, Spotify, news articles).

Test Steps:

1. Create a new post on the platform.
2. Paste the URL from a supported external media source into the post.
3. Publish the post.
4. Verify that the content from the external media source is successfully embedded.
5. Verify that the embedded content displays correctly and is functional within the post.

Postconditions: The embedded content should be visible and functional within the platform.

Expected Result: Users should be able to embed content from supported external media sources into their posts, and the embedded content should function as intended.

Severity: Major

Priority: Medium

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: CPI-005

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

Test Case Description: This test case aims to ensure that the overall experience of connecting and sharing content across platforms is smooth and user-friendly. It assesses various aspects like UI consistency, error handling, and the intuitiveness of the linking and sharing process.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

- User is logged into their social media platform account.

Test Data: N/A

Test Steps:

1. Attempt to link accounts with various social media platforms, including successful and unsuccessful attempts.
2. Share different types of content (text, images, videos) to linked platforms.
3. Navigate through different stages of account linking, content sharing, and error messages (if any).
4. Observe the overall user experience, focusing on ease of use, clarity of instructions, and handling of potential issues.

Postconditions: N/A

Expected Result: The overall process of connecting and sharing content across platforms should be seamless and user-friendly, with intuitive UI elements, clear error messages, and consistent performance.

Severity: Minor

Priority: Medium

Type of Testing: Usability, User Acceptance

Test Case Approach: Positive, Negative

Data - 20

Project name -

Social Media Platform

Project description -

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

Feature name -

Gamification and Rewards

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Gamification and Rewards Feature

Test Case ID: TC_GR_001

Test Case Title: Verify that users earn points and badges for specific activities.

Test Case Description: This test case verifies that users are awarded points and badges for performing various actions on the platform, like creating a profile, adding friends, posting content, and participating in challenges.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- A user account is created and logged in.

- The platform has predefined point and badge awarding rules for specific activities.

Test Data: No test data needed

Test Steps:

1. Log in as a registered user.

2. Complete profile information (e.g., add a profile picture, bio).

3. Send a friend request to another user.

4. Post different types of content (text, image, video).

5. Participate in an active challenge and complete it.

Postconditions:

- Check the user's profile for awarded points and badges.

- Verify points and badges are awarded according to predefined rules.

Expected Result: The user should receive points and badges corresponding to each completed activity.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_GR_002

Test Case Title: Verify that the leaderboard accurately displays top users based on points or achievements.

Test Case Description: This test case ensures the leaderboard functionality displays users ranked correctly based on accumulated points or specific achievements. This involves verifying the sorting, updating, and accuracy of displayed user data.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- Multiple users exist on the platform with varying points and achievements.
- A leaderboard feature is implemented and accessible.

Test Data: No test data needed

Test Steps:

1. Access the Leaderboard section within the application.
2. Observe the ranking of users based on points.
3. Trigger point changes for specific users (e.g., through activity completion).
4. Refresh the Leaderboard and verify the ranking update reflects the changes.
5. Check if filters for different achievements are available and functioning correctly.

Postconditions: None

Expected Result: The leaderboard should accurately reflect the ranking of users based on points or selected achievement criteria, updating dynamically as points change.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_GR_003

Test Case Title: Verify that challenges are available, completable, and award rewards upon completion.

Test Case Description: This test case focuses on the functionality of challenges within the platform. It confirms that challenges are accessible, track user progress, and deliver appropriate rewards upon successful completion.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- The platform has a dedicated section for active challenges.
- Challenges have defined objectives, durations, and rewards.

Test Data: No test data needed

Test Steps:

1. Navigate to the Challenges section in the application.
2. Choose an available challenge and review its objectives and reward.
3. Participate in the challenge and make progress towards its completion.
4. Observe the progress tracking mechanism within the challenge.
5. Complete all objectives of the challenge within the given timeframe.

Postconditions:

- Verify the user receives the pre-defined reward for completing the challenge.

- Ensure completed challenges are marked accordingly in the user's challenge history.

Expected Result: Users should be able to participate in, track progress on, and complete challenges, receiving the appropriate reward upon successful completion.

Severity: Major

Priority: High

Type of Testing: Functional Testing, End-to-End Testing

Test Case Approach: Positive

Test Case ID: TC_GR_004

Test Case Title: Verify that rewards for users include various types like virtual goods, discounts, or exclusive content.

Test Case Description: This test case examines the diversity and proper delivery of rewards within the platform. It ensures that users can receive and utilize different types of rewards earned through their activities and achievements.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- Users have earned rewards of various types (e.g., virtual goods, discounts, exclusive content).

- Mechanisms for redeeming and utilizing rewards are in place.

Test Data:

- A list of available virtual goods.

- Discount codes or coupons.

- Links to exclusive content.

Test Steps:

1. Acquire different types of rewards through challenges or activities.

2. Attempt to redeem available virtual goods through the designated system.

3. Apply any earned discount codes or coupons during a purchase or action where applicable.

4. Access and utilize any exclusive content unlocked through reward redemption.

Postconditions: None

Expected Result: Users should be able to successfully redeem, access, and utilize different types of rewards, experiencing their intended value within the platform.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_GR_005

Test Case Title: Verify that users can access a dedicated section to view their progress, achievements, and earned rewards.

Test Case Description: This test case validates the functionality of a dedicated section for users to track their progress, review earned achievements, and manage acquired rewards. This ensures users have a centralized location to monitor their gamification journey.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- A user account with activity history and earned rewards.

- A dedicated "Achievements" or "Rewards" section in the user profile.

Test Data: No test data needed

Test Steps:

1. Log in as a user with existing activity history.
2. Navigate to the designated "Achievements" or "Rewards" section within the user profile.
3. Verify the accurate display of earned points and badges.
4. Check for a clear and organized presentation of completed challenges and earned rewards.
5. Confirm the availability of information regarding unredeemed rewards and their expiration (if applicable).

Postconditions: None

Expected Result: The dedicated section should provide a comprehensive and user-friendly overview of the user's gamification progress, achievements, and earned rewards within the platform.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 21

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Creation

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Task Management Tool - Task Creation Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
TC_001	Verify that the user can create a task with all required fields filled	This test case verifies that a user can successfully create a task by entering all the required information, including task title, description, due date, and priority level.	Task Creation	High	- User is logged in. - User is on the "Create Task" page.	Task Title: Test Task 1<br> Description: This is a test task.<br> Due Date: 2024-01-01<br> Priority: High	1. Enter "Test Task 1" in the "Task Title" field. 2. Enter "This is a test task." in the "Description" field. 3. Select "2024-01-01" in the "Due Date" field. 4. Select "High" from the "Priority" dropdown. 5. Click the "Create Task" button.	- Verify the task is created and displayed in the task list. - Verify all task details are accurately saved.	The task is successfully created and displayed in the task list with the entered information.	Critical	High	Functional, Acceptance	Positive
TC_002	Verify that the user can assign the task to a specific team member	This test case checks if the user can assign the created task to a specific team member during task creation.	Task Creation	High	- User is

logged in. - User is on the "Create Task" page. - At least one other team member exists. | Task Title: Test Task 2<br> Description: This is a test task for assignment. <br> Due Date: 2024-01-05<br> Priority: Medium<br> Team Member: [Select a team member from the list] | 1. Enter task details (Title, Description, Due Date, Priority). 2. Select a team member from the "Assign To" dropdown list. 3. Click the "Create Task" button. | - Verify the task is displayed in the task list. - Verify the assigned team member is correctly associated with the task. | The task is created and assigned to the selected team member. | Major | High | Functional, Acceptance | Positive |

| TC_003 | Verify that the system saves and displays the created task | This test case validates that the system saves all task details entered by the user during creation and displays them correctly in the task list view. | Task Creation | High | - User is logged in. - User is on the "Create Task" page. | Task Title: Test Task 3<br> Description: This task verifies saving and display functionality.<br> Due Date: 2024-01-10<br> Priority: Low | 1. Enter all required task details. 2. Click the "Create Task" button. 3. Navigate to the task list view. | - Verify the newly created task is visible in the task list. | The newly created task is displayed in the task list with all the entered details. | Critical | High | Functional, Acceptance | Positive |

| TC_004 | Verify that the user can edit task details after creation. | This test case confirms that users can modify any details of a previously created task, such as title, description, due date, or assigned team member. | Task Creation | High | - User is logged in. - A task has been created by the user. | No test data needed. | 1. Navigate to the task list view. 2. Click on the existing task to open the "Edit Task" view. 3. Modify any task detail (title, description, due date, priority, assigned team member). 4. Click the "Save Changes" button. | - Verify the task details are updated successfully in the system. - Verify the updated task details are reflected in the task list view. | The edited task details are saved correctly and reflected in both the task details view and the task list view. | Major | High | Functional, Acceptance | Positive |

| TC_005 | Verify that the system validates required fields during task creation | This test case ensures that the system enforces data validation rules and prevents task creation if mandatory fields like task title are left empty. | Task Creation | High | - User is logged in. - User is on the "Create Task" page. | No test data needed. | 1. Leave the "Task Title" field empty. 2. Attempt to create the task by clicking the "Create Task" button. | N/A | An error message is displayed, indicating that the "Task Title" field is required and the task is not created. |

Critical | High | Functional, Negative, Acceptance | 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: TMT-TA-001

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

Test Case Description: This test case verifies that a user can successfully assign a new task to a specific team member from the list of available team members.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* User has created a new task.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Tasks' section.

2. Click on the created task to open task details.

3. Locate the 'Assignee' field.

4. Select a team member from the dropdown list.

5. Save the task.

Postconditions:

* Verify the assigned team member is displayed in the 'Assignee' field.

* Verify that the task appears in the assignee's task list.

Expected Result: The task is successfully assigned to the selected team member.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TMT-TA-002

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

Test Case Description: This test case verifies that upon task assignment, the assignee receives a notification informing them about the assigned task, the task details, and the assigner's information.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* User has created a new task.

* Notifications are enabled for the assignee.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Tasks' section.

2. Create a new task and assign it to a team member.

3. Log in as the assigned team member.

Postconditions:

* Verify the assignee's notification center.

Expected Result: The assignee receives a notification about the newly assigned task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TMT-TA-003

Test Case Title: Verify that the system should display assigned tasks in the assignee's task list.

Test Case Description: This test case validates that when a task is assigned to a team member, the task appears in their personal task list view.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* User has created a new task.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Tasks' section.

2. Assign the created task to a specific team member.

3. Log in as the assigned team member.

4. Navigate to the 'My Tasks' or equivalent section.

Postconditions: None

Expected Result: The assigned task is displayed in the assignee's task list.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TMT-TA-004

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

Test Case Description: This test case ensures that users can effectively reassign an existing task from one team member to another, updating the assignee and reflecting the change in both users' task lists.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* A task is already assigned to a team member.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Tasks' section.

2. Open the details of the assigned task.

3. Locate the 'Assignee' field.

4. Select a different team member from the dropdown list.

5. Save the changes to the task.

Postconditions:

* Verify the previous assignee no longer sees the task in their list.

* Verify the new assignee now has the task in their task list.

Expected Result: The task is successfully reassigned to the new team member.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: TMT-TA-005

Test Case Title: Verify that task assignment history should be tracked.

Test Case Description: This test case verifies that the system logs and displays the task assignment history, including the original assignee, the new assignee, and the timestamp of each reassignment.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

* User is logged into the Task Management Tool.

* A task has been reassigned at least once.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Tasks' section.

2. Open the details of the task with reassignment history.

3. Locate the 'Task History' or a similar section within the task details.

Postconditions: None

Expected Result: The 'Task History' section displays a clear log of all assignment changes, including dates and assignees.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 23

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Deadline Reminders

Feature description -

Users receive reminders and notifications for upcoming task deadlines, ensuring that tasks are completed on time and preventing delays. Reminders can be customized based on users' preferences and urgency levels.

Userstory title -

As a user, I want to receive reminders for upcoming task deadlines so that I can complete tasks on time.

Acceptance criteria -

['Users can set reminders for tasks based on deadlines.', 'Reminders can be customized by urgency level.', 'The system sends notifications for upcoming deadlines.', 'Users can choose the delivery method for reminders (e.g., email, in-app).', 'Reminder settings can be adjusted or disabled.']

Generated Test Cases:

## Test Cases for Task Deadline Reminders

Test Case ID: TDL-001

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

Test Case Description: This test case verifies that users can successfully set reminders for individual tasks with different deadlines.

Test Suite: Reminders Functionality

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.

- User has created a new task.

Test Data: No test data needed

Test Steps:

1. Navigate to the newly created task.

2. Locate the "Add Reminder" or "Set Reminder" option.

3. Select a date and time for the reminder that aligns with the task deadline.

4. Save the reminder.

Postconditions:

- The reminder is successfully saved and associated with the task.

- The user can view the set reminder within the task details.

Expected Result: The reminder is set for the selected date and time, linked to the task.

Severity: Critical

Priority: High

Type of Testing: Functional, User Interface (UI) Testing

Test Case Approach: Positive

Test Case ID: TDL-002

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

Test Case Description: This test case verifies that users can customize the urgency level of task reminders.

Test Suite: Reminders Functionality

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.

- User is on the "Reminder Settings" or equivalent page.

Test Data: No test data needed

Test Steps:

1. Access the reminder settings for a specific task or globally.

2. Locate the "Urgency Level" or similar option.

3. Choose from available urgency levels (e.g., High, Medium, Low).

4. Save the changes to the reminder settings.

Postconditions:

- The selected urgency level is saved and applied to future reminders.

- The user can see the chosen urgency level reflected in their settings.

Expected Result: Reminders are customizable based on urgency level, allowing for personalized notifications.

Severity: Major

Priority: High

Type of Testing: Functional, Usability Testing

Test Case Approach: Positive

Test Case ID: TDL-003

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

Test Case Description: This test case verifies the system's ability to send timely notifications for upcoming task deadlines.

Test Suite: Notifications and Reminders

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.

- A task with a reminder set for a future date exists.

Test Data: Specific date and time for the reminder slightly in the future.

Test Steps:

1. Create a task and set a reminder for a specific date and time in the near future.

2. Wait for the set reminder time.

3. Observe if the system generates a notification.

Postconditions: None

Expected Result: The system successfully sends a notification to the user at the specified reminder time, indicating the upcoming task deadline.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TDL-004

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

Test Case Description: This test case verifies that users can select their preferred delivery method for task deadline reminders.

Test Suite: Reminders Functionality

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.

- User is on the "Notification Settings" or a similar page.

Test Data: No test data needed.

Test Steps:

1. Navigate to the notification settings within the application.

2. Locate the section for "Reminder Delivery Method" or similar.

3. Choose from available options (e.g., email, in-app notifications, both).

4. Save the preferred delivery method.

Postconditions:

- The selected delivery method is saved successfully.

- Future reminders are sent via the chosen method(s).

Expected Result: Users can select and save their desired delivery method for receiving task deadline reminders.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TDL-005

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

Test Case Description: This test case verifies the functionality for users to adjust or disable reminder settings according to their needs.

Test Suite: Reminders Functionality

Test Priority: Medium

Preconditions:

- User is logged into the Task Management Tool.
- User has previously set reminders.

Test Data: No test data needed

Test Steps:

1. Access the "Reminder Settings" for a specific task or globally.
2. Locate the option to "Edit" or "Disable" existing reminders.
3. Adjust the date, time, or urgency level as needed, or choose to disable the reminder entirely.
4. Save the changes made to the reminder settings.

Postconditions:

- Changes to reminder settings are saved successfully.
- Disabled reminders are no longer active.

Expected Result: Users can easily modify or turn off reminders for tasks, providing flexibility in managing notifications.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive, Negative (attempting to disable an already disabled reminder)

Data - 24

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Task Prioritization Feature

Test Case ID: TP-001

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

Test Case Description: This test case verifies that users can successfully assign different priority levels (High, Medium, Low) to newly created tasks.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged into the application.
- User has navigated to the task creation screen.

Test Data: No test data needed.

Test Steps:

1. Access the task creation form.
2. Locate the priority level dropdown or selection control.
3. Select each priority level (High, Medium, Low) for different tasks.
4. Save each task with the selected priority level.

Postconditions:

- Tasks are successfully created and visible in the task list.
- Created tasks display the assigned priority level.

Expected Result: Users should be able to set the desired priority level for each new task without encountering any errors.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TP-002

Test Case Title: Verify that tasks can be sorted based on priority levels.

Test Case Description: This test case checks if the application correctly sorts tasks in ascending or descending order based on their assigned priority (High > Medium > Low).

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in and viewing a list of tasks.

- Tasks with different priority levels exist in the list.

Test Data: No test data needed.

Test Steps:

1. Locate the sorting functionality for the task list.

2. Select "Priority" as the sorting criteria.

3. Verify that the tasks are sorted correctly in ascending order (High to Low).

4. Change the sorting order to descending (Low to High).

5. Verify that the tasks are now sorted correctly in descending order.

Postconditions: None.

Expected Result: The application should accurately sort and display tasks based on their assigned priority levels in both ascending and descending orders.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TP-003

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

Test Case Description: This test case ensures users can modify the priority level of an existing task and the system reflects the change correctly.

Test Suite: Task Management

Test Priority: High

Preconditions:

- User is logged in and viewing a task list.

- At least one task exists with an assigned priority.

Test Data: No test data needed.

Test Steps:

1. Select an existing task from the list.

2. Locate the option to edit the task details.

3. Change the priority level of the task to a different level.

4. Save the changes made to the task.

5. Verify that the task now reflects the updated priority level.

Postconditions: None.

Expected Result: The system should successfully update and display the new priority level of the task across all relevant views.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TP-004

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

Test Case Description: This test confirms the application visually differentiates tasks with different priorities using distinct visual cues (e.g., color coding, labels, icons).

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged in and viewing a list of tasks.

- Tasks with different priority levels are present.

Test Data: No test data needed.

Test Steps:

1. Observe the visual presentation of tasks in the list.

2. Verify that tasks with different priorities have distinct visual indicators.

3. Check for consistency in visual cues across different views and sections.

Postconditions: None.

Expected Result: Tasks should be easily distinguishable based on their priority levels through clear and consistent visual cues.

Severity: Minor

Priority: Medium

Type of Testing: UI Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TP-005

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

Test Case Description: This test ensures that any task created without an explicitly set priority level is automatically assigned a "Medium" priority.

Test Suite: Task Management

Test Priority: Medium

Preconditions:

- User is logged into the application.

- User is creating a new task.

Test Data: No test data needed.

Test Steps:

1. Create a new task without selecting any priority level.

2. Save the task without specifying a priority.

3. Navigate to the task list and locate the recently created task.

4. Verify that the task displays a "Medium" priority level.

Postconditions: None.

Expected Result: The system should automatically assign a "Medium" priority to any task created without an explicit priority selection.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

Data - 25

Project name -

Task Management Tool

Project description -

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

Feature name -

Subtasks

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Task Management Tool - Subtasks Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
ST-001	Verify that users can create subtasks under main tasks	This test case verifies the functionality of creating subtasks associated with a main task.	Subtasks	High	- User is logged in. - A main task exists.	No test data needed	1. Navigate to the main task. 2. Locate and select the "Add Subtask" option. 3. Enter a title for the subtask. 4. Click the "Create Subtask" button.	- Subtask is displayed under the main task.	Subtask is created successfully and displayed under the parent task.	Critical	High	Functional, Acceptance	Positive
ST-002	Verify that subtasks can have their own due dates and assignees	This test case validates that users can set individual due dates and assign responsible users to each subtask.	Subtasks	High	- User is logged in. - A main task with at least one subtask exists.	No test data needed	1. Navigate to the subtask. 2. Locate the "Due Date" field and select a date. 3. Locate the "Assignee" field and select a user from the list. 4. Save the changes made to the subtask.	- Due date and assignee are updated for the subtask.	Subtask due date and assignee are successfully set and saved.	Critical	High	Functional, Acceptance	Positive
ST-003	Verify that the system tracks the progress of each subtask	This test case checks if the system accurately reflects the completion status of individual subtasks.	Subtasks	High	- User is logged in. - A main task with multiple subtasks exists. - Some subtasks are marked as complete.	No test data needed	1. Navigate to the main task view. 2. Observe the progress bar or indicator for the main task. 3. Compare the displayed progress with the actual completion status of subtasks.	- Progress bar accurately reflects the

percentage of completed subtasks. | Main task progress visually reflects the completion status of its subtasks. | Major | High | Functional, Acceptance | Positive |
| ST-004 | Verify that subtasks can be marked as complete independently of the main task. | This test case ensures that completing a subtask does not automatically mark the parent task as complete. | Subtasks | High | - User is logged in. - A main task with at least one incomplete subtask exists. | No test data needed | 1. Navigate to the incomplete subtask. 2. Mark the subtask as complete. 3. Navigate back to the main task view. | - Subtask is marked as complete. - Main task remains in its original state (e.g., "In Progress"). | Subtask completion status is independent of the main task. | Major | High | Functional, Acceptance | Positive |
| ST-005 | Verify that dependencies between subtasks can be set. | This test case checks if users can establish dependencies between subtasks, ensuring one subtask is completed before another can begin. | Subtasks | High | - User is logged in. - A main task with at least two subtasks exists. | No test data needed | 1. Navigate to a subtask that should be blocked by another. 2. Locate and select the "Dependencies" option. 3. Choose the blocking subtask from the list of available subtasks. 4. Attempt to mark the dependent subtask as complete. | - Dependent subtask cannot be marked complete before the blocking subtask is finished. | Subtask dependencies are successfully enforced by the system. | Major | High | Functional, Acceptance | Positive |

Data - 26

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Dependencies

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Task Management Tool - Task Dependencies Feature

Test Case ID: TMT-TD-001

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

Test Case Description: This test case verifies that users can successfully establish dependency relationships between two or more tasks within the Task Management Tool.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* User has created at least two tasks.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.

2. Select a task and access its details.

3. Locate the "Dependencies" section or option.

4. Choose another task from the list to set as a dependency.

5. Save the task dependency.

Postconditions:

* Verify that the dependency is correctly displayed in the task details of both tasks.

Expected Result: The system should successfully link the selected tasks, establishing a dependency between them.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TMT-TD-002

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

Test Case Description: This test case ensures that the system clearly and accurately presents the established task dependencies within the detailed view of each related task.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* A dependency between two tasks has been previously set.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.

2. Select a task that has a dependency.

3. Open the task details view.

4. Locate the "Dependencies" section.

Postconditions: None

Expected Result: The "Dependencies" section should clearly display all dependent tasks, indicating their status (complete or incomplete).

Severity: Critical

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TMT-TD-003

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

Test Case Description: This test case ensures that the system enforces task dependencies by preventing a task from being marked as complete if it has incomplete dependent tasks.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* A dependency between two tasks exists where the dependent task is not yet complete.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.

2. Attempt to mark the task with dependencies as complete.

Postconditions: None

Expected Result: The system should prevent the user from marking the task as complete and display an error message or warning indicating the incomplete dependencies.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Negative

Test Case ID: TMT-TD-004

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

Test Case Description: This test case ensures that users can modify or delete existing task dependencies within the Task Management Tool.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

* User is logged into the Task Management Tool.

* A dependency between two tasks has been previously set.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.

2. Select the task with the existing dependency.

3. Access the task details view.

4. Locate the "Dependencies" section.

5. Choose to either edit the existing dependency (change the dependent task) or remove it completely.

6. Save the changes.

Postconditions:

* Verify that the system reflects the updated or removed dependency accurately in the task details of both

related tasks.

Expected Result: The system should allow modification or removal of the dependency and reflect these changes accordingly.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TMT-TD-005

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

Test Case Description: This test case checks the system's ability to provide timely notifications or alerts to users when a dependent task is approaching or has passed its deadline without being completed.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

* User is logged into the Task Management Tool.
* A dependency between two tasks exists with a defined deadline for the dependent task.
* The deadline for the dependent task is approaching or has passed.

Test Data: Specific date and time for the dependent task's deadline.

Test Steps:

1. Set the system date/time to a point where the deadline is nearing or past due.
2. Observe the system for any alerts or notifications related to the overdue dependent task.

Postconditions: None

Expected Result: The system should generate a clear alert or notification informing the user about the incomplete dependent task and its potential impact on the overall workflow.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, System Testing, Notification 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 Cases for Task Comments and Collaboration

Test Case ID: TC_001

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

Test Case Description: This test case verifies that a user can successfully add a text comment to an existing task.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged into the Task Management Tool.

* A task has been created.

Test Data: No test data needed

Test Steps:

1. Navigate to the task.

2. Locate the comment section.

3. Enter a text comment.

4. Click the "Add Comment" button.

Postconditions:

* The comment is displayed below the task.

* The comment is saved and persists across sessions.

Expected Result: The comment is successfully added to the task.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_002

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

Test Case Description: This test case verifies that users can add comments containing text, attachments (e.g., images, documents), and hyperlinks.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged into the Task Management Tool.

* A task has been created.

Test Data:

* Sample text comment: "This is a test comment."

* Sample attachment: Image file (e.g., .jpg, .png)

* Sample link: https://www.example.com

Test Steps:

1. Navigate to the task.

2. Locate the comment section.

3. Enter the sample text comment.

4. Upload the sample image file as an attachment.

5. Paste the sample link into the comment field.

6. Click the "Add Comment" button.

Postconditions:

* The comment, including the attachment and link, is displayed below the task.

* Clicking the link opens the correct webpage in a new tab.

Expected Result: The comment with text, attachment, and link is successfully added to the task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TC_003

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

Test Case Description: This test case verifies that users can reply to existing comments and edit their own comments within a task.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged into the Task Management Tool.

* A task with at least one existing comment exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task with the existing comment.

2. Click the "Reply" button below the comment.

3. Enter a reply comment.

4. Click the "Add Comment" button.

5. Click the "Edit" button on their original comment.

6. Modify the original comment.

7. Click the "Save" button.

Postconditions:

* The reply comment is displayed threaded under the original comment.

* The original comment reflects the applied edits.

Expected Result: Users can successfully reply to and edit their own comments.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_004

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

Test Case Description: This test case verifies that users receive notifications when new comments are added to a task they are assigned to or following.

Test Suite: Task Collaboration, Notifications

Test Priority: High

Preconditions:

* Two users (User A and User B) are logged into the Task Management Tool.

* User A is assigned to or following a task.

Test Data: No test data needed.

Test Steps:

1. User B navigates to the task and adds a new comment.

2. Observe User A's notification area within the application.

Postconditions: N/A

Expected Result: User A receives a notification alerting them to the new comment on the task.

Severity: Major

Priority: High

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_005

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

Test Case Description: This test case ensures that all comments within a task are displayed in chronological order, with the oldest comment at the top and the newest at the bottom.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

* The user is logged into the Task Management Tool.

* A task with multiple comments exists.

Test Data: No test data needed

Test Steps:

1. Navigate to the task with multiple comments.

2. Observe the order of the comments.

Postconditions: N/A

Expected Result: Comments are displayed in chronological order from oldest to newest.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Data - 28

Project name -

Task Management Tool

Project description -

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

Feature name -

File Attachments

Feature description -

Users can attach relevant files, documents, and resources to tasks, ensuring that all necessary information is accessible in one place. File attachments can include documents, spreadsheets, images, links, and other resources needed for task execution.

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for File Attachments in Task Management Tool

Test Case ID: TA-FA-001

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

Test Case Description: This test case verifies the functionality of attaching multiple files to a single task. It checks if the system allows selecting and uploading multiple files simultaneously and if all selected files are successfully attached to the task.

Test Suite: File Attachments

Test Priority: High

Preconditions:

* User is logged into the system.

* User has created a task.

Test Data:

* Multiple files of different types (e.g., .doc, .png, .xlsx).

Test Steps:

1. Navigate to an existing task.

2. Click on the 'Attach Files' button or designated area.

3. Select multiple files from the local system.

4. Click 'Open' or 'Upload' to attach the selected files.

5. Verify that all selected files are listed as attachments to the task.

Postconditions:

* Remove attached files.

Expected Result: All selected files should be successfully attached and displayed in the task's attachment section.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TA-FA-002

Test Case Title: Verify that attachments can include various file types.

Test Case Description: This test case validates the system's ability to handle attachments of different file types. It tests the upload functionality for common file types to ensure compatibility.

Test Suite: File Attachments

Test Priority: High

Preconditions:

* User is logged into the system.

* User has created a task.

Test Data:

* One file for each type: .doc, .docx, .xls, .xlsx, .ppt, .pptx, .pdf, .png, .jpg, .jpeg, .gif, .txt, .zip

Test Steps:

1. Navigate to an existing task.

2. Attempt to attach one file of each type listed in the Test Data.

3. Verify that each file is successfully uploaded and displayed.

Postconditions:

* Remove attached files.

Expected Result: The system should successfully upload and display all tested file types as attachments.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Compatibility Testing

Test Case Approach: Positive

Test Case ID: TA-FA-003

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

Test Case Description: This test case ensures that users can access and download attached files. It verifies the functionality of viewing and downloading attachments from a task.

Test Suite: File Attachments

Test Priority: High

Preconditions:

* User is logged into the system.

* A task exists with attached files of different types.

Test Data: No test data needed

Test Steps:

1. Navigate to the task with existing attachments.

2. Click on the file name or the designated 'view' icon for each attachment.

3. Verify that the file opens correctly in a new tab or within the platform's viewer.

4. Click the download icon for each attachment.

5. Verify that the file is downloaded successfully to the local system.

Postconditions: No specific postconditions.

Expected Result: Users should be able to view the content of attached files and download them without errors.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TA-FA-004

Test Case Title: Verify that the system limits the size of attachments.

Test Case Description: This test case aims to validate that the system enforces file size limits for attachments. It tests the system's response to attempts to upload files exceeding the defined limit.

Test Suite: File Attachments

Test Priority: High

Preconditions:

* User is logged into the system.

* User has created a task.

Test Data:

* A file exceeding the defined attachment size limit.

Test Steps:

1. Navigate to an existing task.

2. Attempt to attach the oversized file.

3. Observe the system response.

Postconditions: No specific postconditions.

Expected Result: The system should prevent the upload of the oversized file and display a clear error message indicating the size limit.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Negative Testing

Test Case Approach: Negative

Test Case ID: TA-FA-005

Test Case Title: Verify that users can remove attachments if needed.

Test Case Description: This test case checks the functionality for users to delete unwanted attachments from a task. It ensures that users can remove incorrect or unnecessary attachments before or after task completion.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

* User is logged into the system.

* A task exists with attached files.

Test Data: No test data needed

Test Steps:

1. Navigate to the task with the attachments.

2. Locate the 'Delete' or 'Remove' icon/button next to each attachment.

3. Click the 'Delete' or 'Remove' element for one of the attachments.

4. Confirm the deletion if prompted.

5. Verify that the selected attachment is removed from the task.

Postconditions: No specific postconditions.

Expected Result: The selected attachment should be successfully removed from the task without impacting other attachments or task details.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 29

Project name -

Task Management Tool

Project description -

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

Feature name -

Recurring Tasks

Feature description -

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

Userstory title -

As a user, I want to set recurring tasks so that regular activities are automated.

Acceptance criteria -

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

Generated Test Cases:

## Test Cases for Recurring Tasks Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
RT-001	Verify that users can set tasks to recur at specified intervals	This test case verifies the functionality of setting a task to repeat daily, weekly, or monthly.	Recurring Tasks	High	- User is logged in. - User is on the "Create Task" page.	Task Name: Test Recurring Task, Description: This is a test for recurring tasks, Due Date: One week from today	1. Enter task details. 2. Select the "Repeat" option. 3. Choose a recurrence interval (daily, weekly, monthly). 4. Set the desired start date. 5. Click "Create Task."	- Verify the recurring task is created and visible in the task list.	The task should be created successfully and scheduled to recur at the chosen interval.	Blocker	High	Functional, Acceptance	Positive
RT-002	Verify that the system generates instances of recurring tasks automatically	This test case checks if the system automatically creates instances of a recurring task based on the defined recurrence pattern.	Recurring Tasks	High	- A recurring task is set up with a future start date.	No test data needed.	1. Wait until the recurring task's start date and time. 2. Check the user's task list.	- No changes needed.	A new instance of the recurring task should appear in the user's task list on the specified date.	Blocker	High	Functional, Regression	Positive
RT-003	Verify that users can modify the recurrence pattern or cancel the recurrence of a task	This test case validates that users can modify the recurrence settings of a task, such as changing the interval or canceling the recurrence altogether.	Recurring Tasks	High	- A recurring task is created.	No test data needed.	1. Locate the existing recurring task. 2. Access the task's edit/view mode. 3. Modify the recurrence interval or choose "Cancel Recurrence." 4. Save the changes.	- Verify changes in the task list view and upcoming instances.	The system should successfully update the recurrence settings or cancel the recurrence as per the user's modifications.	Critical	High	Functional, Regression	Positive, Negative
RT-004	Verify that the system notifies users of upcoming instances of recurring tasks.	This test case checks the functionality of the system to notify users about upcoming instances of their recurring tasks.	Recurring Tasks	High	- A recurring task is set up with notifications enabled. - Notification settings are configured in the user profile.	No test data needed.	1. Approach the due date and time of the recurring task						

instance. 2. Monitor the designated notification channels (e.g., in-app, email). | - Review notification logs. | The system should deliver a notification to the user through their chosen notification channels, reminding them of the upcoming task instance. | Major | High | Functional, Integration | Positive |

| RT-005 | Verify that users can view a list of all instances of a recurring task. | This test case verifies that users can access a comprehensive history or list of all generated instances of a recurring task. | Recurring Tasks | Medium | - A recurring task with past instances exists. | No test data needed. | 1. Navigate to the recurring task's details page. 2. Look for an option to view "History," "Instances," or a similar view. 3. Select the option to view all instances. | - No changes needed. | The system should display a list of all past, current, and potentially future instances of the recurring task, allowing users to track its history. | Minor | Medium | Functional, Usability | Positive |

--

Data - 30

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tags and Labels

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Task Tags and Labels Feature

Test Case ID: TC_TTL_001

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

Test Case Description: This test case verifies the functionality of adding both pre-defined and custom tags and labels to a newly created task.

Test Suite: Task Management - Tags & Labels

Test Priority: High

Preconditions:

- User is logged in to the Task Management Tool.
- User has successfully created a new task.

Test Data: No test data needed

Test Steps:

1. Navigate to the newly created task.
2. Locate the "Tags" or "Labels" section.
3. Add a pre-defined tag/label to the task.
4. Add a custom tag/label to the task.
5. Save the task changes.

Postconditions:

- The task should be updated with the selected tags and labels.

- The added tags and labels should be visible in the task details.

Expected Result: The user should be able to successfully add both pre-defined and custom tags and labels to the task.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TTL_002

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

Test Case Description: This test case checks if users can modify existing tags by editing their names or deleting them entirely.

Test Suite: Task Management - Tags & Labels

Test Priority: High

Preconditions:

- User is logged in to the Task Management Tool.

- At least one task exists with a tag added to it.

Test Data: No test data needed

Test Steps:

1. Navigate to the task with the existing tag.

2. Locate the "Tags" or "Labels" section.

3. Edit the name of the existing tag.

4. Save the changes.

5. Attempt to delete the existing tag.

6. Confirm the deletion.

Postconditions:

- The task should be updated with the edited tag name.

- The tag should be successfully deleted from the list of available tags.

Expected Result: The user should be able to customize and edit existing tags by changing their names and deleting them.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TTL_003

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

Test Case Description: This test case ensures the filtering and sorting functionality based on applied tags and labels functions correctly, allowing users to easily find specific tasks.

Test Suite: Task Management - Tags & Labels

Test Priority: High

Preconditions:

- User is logged in to the Task Management Tool.
- Multiple tasks exist with different tags and labels applied.

Test Data: No test data needed

Test Steps:

1. Navigate to the main task list.
2. Apply a filter using a specific tag.
3. Verify that the displayed tasks match the applied filter.
4. Apply sorting based on a specific label.
5. Verify that tasks are sorted correctly based on the selected label.

Postconditions: None

Expected Result: The system should accurately filter and sort tasks based on the selected tags and labels, displaying only the relevant tasks.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_TTL_004

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

Test Case Description: This test case aims to confirm that tags and labels are clearly visible within the task list view, allowing users to quickly identify task categorization at a glance.

Test Suite: Task Management - Tags & Labels

Test Priority: Medium

Preconditions:

- User is logged in to the Task Management Tool.

- Tasks with various tags and labels are present in the task list.

Test Data: No test data needed

Test Steps:

1. Navigate to the main task list.

2. Observe the display of tasks with applied tags and labels.

3. Verify that tags and labels are visually distinct and easy to identify.

4. Check for consistent placement and styling of tags and labels across different tasks.

Postconditions: None

Expected Result: Tags and labels should be displayed prominently and consistently for each task in the task list, enabling easy visual identification and categorization.

Severity: Minor

Priority: Medium

Type of Testing: UI Testing

Test Case Approach: Positive

Test Case ID: TC_TTL_005

Test Case Title: Verify that users can remove tags and labels from tasks.

Test Case Description: This test case confirms the functionality of removing previously added tags and labels from a task, ensuring users have full control over task categorization.

Test Suite: Task Management - Tags & Labels

Test Priority: Medium

Preconditions:

- User is logged in to the Task Management Tool.

- At least one task exists with a tag and/or label applied to it.

Test Data: No test data needed

Test Steps:

1. Navigate to a task with an existing tag/label.

2. Locate the "Tags" or "Labels" section.

3. Attempt to remove the existing tag/label from the task.

4. Save the task changes.

Postconditions:

- The task should be updated with the tag/label removed.

Expected Result: The user should be able to successfully remove existing tags and labels from the task.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Negative

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 Cases for Task Status Tracking Feature:

Test Case ID: TST-001

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

Test Case Description: This test case verifies that a user can successfully change the status of a task from "To Do" to other available statuses like "In Progress" and "Completed".

Test Suite: Task Management - Status Tracking

Test Priority: High

Preconditions:

- User is logged into the system.

- A task has been created.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.

2. Select a task in "To Do" status.

3. Click on the task status dropdown menu.

4. Select "In Progress" from the available options.

5. Verify the task status is updated to "In Progress".

6. Repeat steps 3-5 for "Completed" status.

Postconditions:

- The task status is updated in the database.

- Status change is reflected in the task list view.

Expected Result: The task status should be successfully updated, and the change should be reflected in the task list view.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TST-002

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

Test Case Description: This test case verifies that the status of each task is clearly visible and easily identifiable within the task list view.

Test Suite: Task Management - Status Tracking

Test Priority: High

Preconditions:

- User is logged into the system.

- The task list is populated with tasks in various statuses.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.

2. Observe the display of tasks within the list.

3. Verify that each task displays its status clearly (e.g., using a color code, label, or dedicated column).

4. Ensure the status display is consistent across all tasks in the list.

Postconditions: None

Expected Result: The status of each task should be prominently displayed and easily identifiable within the task list.

Severity: Major

Priority: High

Type of Testing: UI Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TST-003

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

Test Case Description: This test case ensures users can view the completion percentage of a task, reflecting its progress.

Test Suite: Task Management - Status Tracking

Test Priority: Medium

Preconditions:

- User is logged into the system.

- A task is in "In Progress" status.

Test Data:

- Task with subtasks or a progress bar implementation.

Test Steps:

1. Navigate to the task list.

2. Open a task in "In Progress" status.

3. Verify that the task view displays a progress percentage.

4. If applicable, confirm the progress percentage is calculated based on completed subtasks or progress input.

Postconditions: None

Expected Result: The task view should clearly display the progress percentage, accurately reflecting the task's completion status.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TST-004

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

Test Case Description: This test case checks if the system generates and delivers appropriate notifications to users when the status of a task is changed.

Test Suite: Task Management - Status Tracking

Test Priority: Medium

Preconditions:

- User is logged into the system.

- User has enabled notifications for task status changes.

Test Data: No test data needed

Test Steps:

1. Navigate to user notification settings.

2. Confirm that notifications for "Task Status Changes" are enabled.

3. Have another user or admin change the status of a task assigned to the logged-in user.

4. Verify that the logged-in user receives a notification about the status change.

5. Check if the notification includes relevant details like the task name, old status, new status, and timestamp.

Postconditions: None

Expected Result: The system should promptly send a notification to the user informing them about the change in task status, including relevant details.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TST-005

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

Test Case Description: This test case ensures that users can effectively filter and sort the task list based on the status of tasks.

Test Suite: Task Management - Status Tracking

Test Priority: Medium

Preconditions:

- User is logged into the system.

- The task list is populated with tasks in various statuses.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.

2. Locate the filter/sort options for task status.

3. Apply a filter to display tasks with a specific status (e.g., "In Progress").

4. Verify that the task list is filtered correctly, showing only tasks matching the selected status.

5. Apply sorting based on task status (e.g., ascending or descending order).

6. Verify that the tasks are sorted correctly in the list based on their status.

Postconditions: None

Expected Result: The system should allow users to filter and sort tasks based on their status, displaying the results accurately according to the applied filter and sort options.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 32

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Filtering and Sorting

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Task Filtering and Sorting

Test Case ID: TFS-001

Test Case Title: Verify that users can filter tasks based on due date.

Test Case Description: This test case verifies the functionality of filtering tasks based on their due dates. The user should be able to filter tasks to display only those due today, tomorrow, this week, next week, or within a custom date range.

Test Suite: Task Filtering

Test Priority: High

Preconditions:

* The user is logged in to the application.

* The user has created at least three tasks with different due dates.

Test Data:

* Due date for Task 1: Today

* Due date for Task 2: Tomorrow

* Due date for Task 3: Next week

Test Steps:

1. Navigate to the task list view.

2. Select the "Due Date" filter option.

3. Choose "Today" from the dropdown menu.

4. Verify that only tasks due today are displayed.

5. Repeat steps 2-4 for "Tomorrow", "This Week", "Next Week" and a custom date range.

Postconditions:

* None

Expected Result: The task list is filtered correctly based on the selected due date filter, displaying only tasks matching the criteria.

Severity: Major

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TFS-002

Test Case Title: Verify that tasks can be sorted by priority.

Test Case Description: This test case verifies if users can sort tasks based on their assigned priority levels (e.g., High, Medium, Low).

Test Suite: Task Sorting

Test Priority: High

Preconditions:

* The user is logged in to the application.

* The user has created at least three tasks with different priority levels assigned.

Test Data:

* Task 1: High priority

* Task 2: Low priority

* Task 3: Medium priority

Test Steps:

1. Navigate to the task list view.

2. Select the "Priority" sort option.

3. Verify that the tasks are sorted in descending order of priority (High to Low).

4. Select the "Priority" sort option again.

5. Verify that the tasks are now sorted in ascending order of priority (Low to High).

Postconditions:

* None

Expected Result: The task list is sorted correctly based on the selected priority level in both ascending and descending orders.

Severity: Major

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TFS-003

Test Case Title: Verify that the system provides multiple filter and sort options in the task list.

Test Case Description: This test case verifies the availability of different filtering and sorting options within the task list.

Test Suite: Task List UI

Test Priority: Medium

Preconditions:

* The user is logged in to the application.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list view.

2. Locate the filter and sort controls within the interface.

3. Verify that the following filter options are available:

    * Due date

    * Priority

    * Assignee

    * Project

4. Verify that the following sort options are available:

    * Due date (Ascending/Descending)

    * Priority (Ascending/Descending)

    * Assignee (Alphabetical/Reverse Alphabetical)

    * Project (Alphabetical/Reverse Alphabetical)

Postconditions:

* None

Expected Result: All required filter and sort options are present and accessible within the task list view.

Severity: Minor

Priority: Medium

Type of Testing: Functional, UI Testing

Test Case Approach: Positive

Test Case ID: TFS-004

Test Case Title: Verify that users can save and apply custom filter and sort settings.

Test Case Description: This test case ensures that users can save their preferred filter and sort settings for future use.

Test Suite: User Preferences

Test Priority: Medium

Preconditions:

* The user is logged in to the application.

* The user has created at least three tasks with different attributes.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list view.

2. Apply a combination of filters and sorting options.

3. Locate the "Save Filter" or "Save View" option (depending on the application's design).

4. Name and save the custom filter and sort settings.

5. Navigate away from the task list view and return.

6. Locate the saved custom filter/view and apply it.

Postconditions:

* Delete the saved filter/view after the test.

Expected Result: The saved filter and sort settings are successfully applied, displaying the task list according to the saved configuration.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: TFS-005

Test Case Title: Verify that filters and sorts do not affect the actual task data.

Test Case Description: This test case confirms that applying filters and sorting only affects the displayed view of tasks and does not alter the underlying task data.

Test Suite: Data Integrity

Test Priority: High

Preconditions:

* The user is logged in to the application.

* The user has created at least three tasks with different attributes.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list view.

2. Note down the details (title, due date, priority, etc.) of several tasks.

3. Apply a combination of filters and sorting options.

4. Verify that the displayed tasks have changed according to the applied filters and sorts.

5. Refresh the task list view or log out and log back in.

6. Verify that the tasks are displayed with their original details and order, confirming no data alteration.

Postconditions:

* None

Expected Result: The filtering and sorting actions only affect the task list view and do not modify the actual task data, ensuring data integrity.

Severity: Critical

Priority: High

Type of Testing: Functional, Non-functional (Data Integrity)

Test Case Approach: Negative

Data - 33

Project name -

Task Management Tool

Project description -

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

Feature name -

Time Tracking

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Task Management Tool - Time Tracking Feature

Test Case ID: TT-001

Test Case Title: Verify that users can start and stop a timer for each task.

Test Case Description: This test case verifies the core functionality of the time tracking feature, ensuring users can accurately track time spent on individual tasks.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

* User is logged in to the application.

* A task has been created.

* The user has accessed the task details.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task details page.

2. Locate the timer control associated with the task.

3. Click the "Start" button on the timer.

4. Verify the timer starts and displays the elapsed time.

5. Click the "Stop" button on the timer.

6. Verify the timer stops and displays the total time elapsed.

Postconditions:

* The timer should display the accurate time spent on the task.

* The recorded time should be saved and associated with the specific task.

Expected Result: Users should be able to start, stop, and view the elapsed time for a specific task.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TT-002

Test Case Title: Verify that the system records and displays the total time spent on tasks.

Test Case Description: This test case ensures the system accurately records and presents the total time spent on a task, including multiple start/stop cycles.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

* User is logged in.

* A task exists with at least one completed time tracking session.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task details page.

2. Locate the section displaying time tracking information.

3. Verify the total time spent on the task is displayed.

4. Start and stop the timer multiple times.

5. Verify the total time displayed is updated accurately after each stop.

Postconditions:

* The displayed total time should reflect all recorded time entries for the task.

Expected Result: The system should accurately calculate and display the total time spent on the task, aggregating time from multiple sessions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TT-003

Test Case Title: Verify that users can manually enter or adjust time entries.

Test Case Description: This test case checks the flexibility of the time tracking system, allowing users to manually manage time entries for accuracy.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* User is logged in.

* A task exists.

Test Data:

* Manually entered time: 1 hour 30 minutes.

Test Steps:

1. Navigate to the task details page.

2. Locate the manual time entry option (e.g., "Add Time" button).

3. Input the desired time spent (e.g., 1 hour 30 minutes).

4. Save the manually entered time.

5. Verify the total time for the task reflects the manual entry.

6. Locate the option to edit existing time entries.

7. Adjust the previously entered time.

8. Verify the total time updates accordingly.

Postconditions: The task's total time should accurately reflect both manually entered and adjusted times.

Expected Result: Users should be able to add, edit, and delete manual time entries, and the system should reflect these changes in the total time spent on a task.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TT-004

Test Case Title: Verify that time tracking data is stored and can be exported.

Test Case Description: This test case ensures the system reliably stores time tracking data and offers export functionality for further analysis or reporting.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* User is logged in.

* Tasks exist with recorded time entries.

Test Data: No test data needed.

Test Steps:

1. Navigate to the reports or data export section.

2. Select the option to export time tracking data.

3. Choose the desired format (e.g., CSV, Excel).

4. Initiate the export process.

5. Verify the downloaded file contains accurate time tracking data for all tasks.

Postconditions: The exported data should accurately represent the time tracking information stored by the system.

Expected Result: The system should allow users to export time tracking data in a commonly used format, ensuring data accessibility and portability.

Severity: Minor

Priority: Medium

Type of Testing: Data Validation Testing, Export Functionality Testing

Test Case Approach: Positive

Test Case ID: TT-005

Test Case Title: Verify that the system provides reports on time spent per task or project.

Test Case Description: This test case validates the system's ability to generate meaningful reports, providing insights into time allocation across tasks and projects.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

* User is logged in.

* Tasks exist with recorded time entries across different projects.

Test Data: No test data needed.

Test Steps:

1. Navigate to the reporting section.

2. Select the option to generate time tracking reports.

3. Choose the desired report type (e.g., time spent per task, time spent per project).

4. Specify any filtering criteria (e.g., date range, specific project).

5. Generate the report.

6. Verify the report accurately displays time spent on tasks within the selected scope.

Postconditions: The generated reports should provide a clear and accurate overview of time allocation based on the selected criteria.

Expected Result: Users should have access to reports that present time tracking data in a clear and informative manner, enabling analysis and decision-making.

Severity: Minor

Priority: Medium

Type of Testing: Reporting Functionality Testing

Test Case Approach: Positive

Data - 34

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Templates

Feature description -

Users can create and use templates for common task types or project workflows, streamlining task creation and standardizing processes across projects. Task templates can include predefined task structures, descriptions, assignees, and due dates, saving time and ensuring consistency in task execution.

Userstory title -

As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

Acceptance criteria -

['Users can create and save task templates.', 'Templates can include predefined task structures, descriptions, assignees, and due dates.', 'Users can apply templates to new tasks or projects.', 'Templates can be edited or deleted as needed.', 'The system displays a list of available templates for selection.']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Templates Feature

Test Case ID: TT-001

Test Case Title: Verify that users can create and save task templates.

Test Case Description: This test case verifies the functionality for users to create new task templates and save them within the system. It will cover the creation of templates with various predefined fields like task structure, descriptions, assignees, and due dates.

Test Suite: Task Templates

Test Priority: High

Preconditions:

* User is logged into the application.

* User has permission to create task templates.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Task Templates' section.

2. Click on the 'Create New Template' button.

3. Enter a template name.

4. Fill in the desired predefined fields (task structure, description, assignee, due date).

5. Click on the 'Save Template' button.

Postconditions:

* The newly created template should be listed in the 'Task Templates' section.

* The saved template should retain all the predefined information.

Expected Result: The system should successfully create and save the new task template with all entered details.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TT-002

Test Case Title: Verify that templates can include predefined task structures, descriptions, assignees, and due dates.

Test Case Description: This test case verifies that when creating a task template, users can define various elements like task structure, description, default assignee, and due date.

Test Suite: Task Templates

Test Priority: High

Preconditions:

* User is logged into the application.

* User is on the 'Create New Template' page.

Test Data:

* Task structure: Checklist

* Description: This is a test template.

* Assignee: John Doe

* Due date: +5 business days

Test Steps:

1. Enter a template name.

2. Choose 'Checklist' as the task structure.

3. Enter the provided description.

4. Select 'John Doe' as the default assignee.

5. Set the due date to '+5 business days'.

6. Save the template.

Postconditions: The saved template should contain the predefined task structure, description, assignee, and due date as specified.

Expected Result: The system should successfully save the template with all the predefined elements.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TT-003

Test Case Title: Verify that users can apply templates to new tasks or projects.

Test Case Description: This test case checks if users can apply existing task templates while creating new tasks or projects. It ensures that the predefined information from the template is populated in the new task.

Test Suite: Task Templates

Test Priority: High

Preconditions:

* User is logged into the application.

* At least one task template is created and saved.

Test Data: Use the previously created template "Test Template".

Test Steps:

1. Navigate to the 'Tasks' or 'Projects' section.

2. Initiate the creation of a new task or project.

3. Choose the option to 'Apply Template'.

4. Select the "Test Template" from the list of available templates.

Postconditions: The new task or project should be populated with the predefined information from the selected template.

Expected Result: The system should successfully apply the chosen template to the new task or project, including all predefined data.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Integration Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TT-004

Test Case Title: Verify that templates can be edited or deleted as needed.

Test Case Description: This test case verifies that existing task templates can be modified or deleted by the user, ensuring flexibility in managing templates.

Test Suite: Task Templates

Test Priority: High

Preconditions:

* User is logged into the application.

* At least one task template is created and saved.

Test Data: Use the previously created template "Test Template".

Test Steps:

1. Navigate to the 'Task Templates' section.

2. Locate the "Test Template".

3. Click on the 'Edit' button for the selected template.

4. Make necessary modifications to the template (e.g., change description, assignee, or due date).

5. Save the changes.

6. Locate the "Test Template" again.

7. Click on the 'Delete' button.

8. Confirm the deletion.

Postconditions:

* The edited template should reflect the changes made.

* The deleted template should no longer be available in the list of templates.

Expected Result: The system should successfully allow editing and deleting of existing task templates.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive, Destructive

Test Case ID: TT-005

Test Case Title: Verify that the system displays a list of available templates for selection.

Test Case Description: This test case ensures that when creating a new task or project, the system presents a list of all saved task templates for the user to choose from.

Test Suite: Task Templates

Test Priority: High

Preconditions:

* User is logged into the application.

* At least one task template is created and saved.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Tasks' or 'Projects' section.

2. Initiate the creation of a new task or project.

3. Click on the 'Apply Template' button or option.

Postconditions: A list containing all saved task templates should be displayed.

Expected Result: The system should successfully display a comprehensive list of available task templates for the user to select from.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing, Acceptance Testing

Test Case Approach: Positive

Data - 35

Project name -

Task Management Tool

Project description -

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

Feature name -

Calendar Integration

Feature description -

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

Userstory title -

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

Acceptance criteria -

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

Generated Test Cases:

## Test Cases for Calendar Integration Feature

Test Case ID: TC_CI_001

Test Case Title: Verify that users can sync tasks with external calendars (Google Calendar).

Test Case Description: This test case verifies the successful synchronization of tasks with a user's Google Calendar. It ensures that the integration allows users to connect their Task Management Tool account with their Google Calendar and sync tasks seamlessly.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- The user has an active account with the Task Management Tool.
- The user has an active Google Calendar account.
- The user has granted the necessary permissions for the Task Management Tool to access their Google Calendar.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the "Calendar Integration" settings.

3. Select "Google Calendar" as the calendar to sync with.

4. Follow the authentication process to connect the accounts.

5. Create a new task in the Task Management Tool.

6. Verify that the task appears in the user's Google Calendar.

Postconditions:

- The task should be visible in the Google Calendar.

- The task details (name, deadline, description) should be accurately reflected.

Expected Result: The task created in the Task Management Tool should successfully synchronize with the user's Google Calendar, appearing with accurate details and deadlines.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CI_002

Test Case Title: Verify that the system displays task deadlines in the calendar view.

Test Case Description: This test case verifies that the integrated calendar view accurately displays task deadlines alongside other calendar events.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- The user has an active account and is logged in.

- The user has synced their calendar with the Task Management Tool.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the calendar view.

3. Verify that tasks with deadlines are displayed on the calendar.

4. Verify that the task deadlines are accurately positioned on the calendar grid.

5. Verify that the task deadlines are visually distinct from other calendar events.

Postconditions: None

Expected Result: The calendar view should accurately display all tasks with deadlines alongside other calendar events, providing a clear visual representation of the user's schedule.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TC_CI_003

Test Case Title: Verify that users can set reminders for tasks via the calendar.

Test Case Description: This test case ensures that users can set reminders for their tasks directly from the integrated calendar view, leveraging the familiar reminder functionality of their chosen calendar application.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- The user has an active account and is logged in.

- The user has synced their calendar with the Task Management Tool.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the calendar view.

3. Select a task displayed on the calendar.

4. Access the task details or options.

5. Set a reminder for the selected task, choosing the desired time and method (e.g., email, pop-up notification).

6. Verify that the reminder is successfully set and reflects in the user?s external calendar.

Postconditions: None

Expected Result: The user should be able to seamlessly set reminders for their tasks via the calendar view, with the reminders reflecting in both the Task Management Tool and their synced external calendar.

Severity: Major

Priority: Medium

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CI_004

Test Case Title: Verify that calendar events update automatically when tasks are changed.

Test Case Description: This test case assesses the real-time synchronization between the Task Management Tool and the external calendar, ensuring that any modifications made to tasks are reflected immediately in the calendar events.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- The user has an active account and is logged in.

- The user has synced their calendar with the Task Management Tool.

- A task with a set deadline exists and is synced with the calendar.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool.

2. Locate and select a task that is synced with the calendar.

3. Modify the task details (e.g., change the deadline, update the description).

4. Save the changes made to the task.

5. Navigate to the calendar view within the Task Management Tool.

6. Verify that the corresponding calendar event reflects the updated task details.

7. Check the synced external calendar to confirm the changes are reflected there as well.

Postconditions: None

Expected Result: Any changes made to a task should automatically update the corresponding calendar event in both the Task Management Tool and the synced external calendar in real-time.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_CI_005

Test Case Title: Verify that users can manage calendar sync settings within the app.

Test Case Description: This test case validates the availability and functionality of the calendar sync settings, allowing users to control how and when their tasks are synchronized with their chosen calendar application.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- The user has an active account and is logged in.

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool.

2. Navigate to the application's settings or preferences.

3. Locate and access the "Calendar Sync" settings.

4. Verify that users can choose to enable or disable calendar sync.

5. Verify that users can select their preferred calendar application for syncing (e.g., Google Calendar, Outlook).

6. Verify that users can customize synchronization options, such as choosing specific calendars to sync or setting synchronization frequency.

7. Make changes to the sync settings and save them.

8. Confirm that the changes are saved and reflected in the application's behavior.

Postconditions: None

Expected Result: The application should provide users with a dedicated section within the settings to manage their calendar sync preferences, allowing them to control the synchronization process according to their needs.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Data - 36

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Duplication

Feature description -

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

Userstory title -

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

Acceptance criteria -

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

Generated Test Cases:

## Test Cases for Task Duplication Feature:

Test Case ID: TMD-001

Test Case Title: Verify that users can duplicate existing tasks.

Test Case Description: This test case verifies the functionality of duplicating an existing task within the task management tool. It ensures that a user can create a copy of a task and that the system handles the duplication process correctly.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

*   The user is logged into the application.

*   A task exists within the system.

*   The user has permission to duplicate tasks.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.

2. Select a task to duplicate.

3. Click the "Duplicate Task" button (or equivalent action).

4. Confirm the duplication if prompted.

Postconditions:

* The duplicated task appears in the task list.

* A system message confirms the successful duplication.

Expected Result: The selected task is successfully duplicated, and a copy of the task, including all its details, appears in the task list.

Severity: Blocker

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TMD-002

Test Case Title: Verify that duplicated tasks retain all original details.

Test Case Description: This test case ensures that when a user duplicates a task, all the original task details are copied over to the new duplicated task without any loss or corruption of information.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

* The user is logged into the application.

* A task with details (description, assignee, deadline, etc.) exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list and select a task with pre-filled details (description, assignee, deadline, etc.).

2. Duplicate the selected task.

3. Open the duplicated task.

4. Verify that all details (description, assignee, deadline, etc.) in the duplicated task match the original task.

Postconditions: None

Expected Result: The duplicated task should accurately reflect all details from the original task, ensuring data integrity during duplication.

Severity: Critical

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TMD-003

Test Case Title: Verify that users can modify duplicated tasks.

Test Case Description: This test case aims to confirm that users have the functionality to modify details of a duplicated task without affecting the original task. This ensures the system treats duplicated tasks as independent entities after duplication.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

* The user is logged into the application.

* A duplicated task exists.

Test Data: No test data needed.

Test Steps:

1. Open the duplicated task.

2. Modify various fields such as the description, assignee, and deadline.

3. Save the changes made to the duplicated task.

4. Open the original task.

Postconditions: None

Expected Result: The modifications should be saved successfully to the duplicated task, while the original task should remain unchanged, confirming the independence of the two tasks.

Severity: Major

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TMD-004

Test Case Title: Verify that the system indicates duplicated tasks.

Test Case Description: This test case verifies that the task management system visually differentiates duplicated tasks from original tasks. This could be through visual cues or by adding a tag to the task name.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

* The user is logged into the application.

* A task has been duplicated.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list where the duplicated task is located.

2. Inspect the visual presentation of the duplicated task.

Postconditions: None

Expected Result: The duplicated task is visually distinct from other tasks (e.g., by a "(Copy)" tag in the task name or a visual indicator), preventing user confusion.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: TMD-005

Test Case Title: Verify that duplicated tasks are saved and displayed in the task list.

Test Case Description: This test case ensures that duplicated tasks are not temporary and are saved within the system. It also validates that they are correctly displayed within the task list for user access.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

* A user is logged into the application.

* A task has been duplicated.

Test Data: No test data needed

Test Steps:

1. Refresh or reload the task list.

2. Search for the duplicated task using its name or other identifying information.

Postconditions: None

Expected Result: The duplicated task should persist in the task list after the refresh, indicating it has been saved correctly and is readily accessible for future use.

Severity: Critical

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Data - 37

Project name -

Task Management Tool

Project description -

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

Feature name -

Task History

Feature description -

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

Userstory title -

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

Acceptance criteria -

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

Generated Test Cases:

## Test Cases for Task History Feature

Test Case ID: TH-001

Test Case Title: Verify that users can view a detailed history of changes to tasks.

Test Case Description: This test case verifies that the task history accurately reflects all modifications made to a task, including changes to its title, description, assignee, due date, priority, status, and any other editable fields.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in.

- A task exists with at least one modification made to it.

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page of the modified task.

2. Locate the "Task History" section.

3. Verify that the history displays a chronological log of all changes made to the task.

4. Verify that each entry in the history includes information on:

    - The type of change (e.g., title updated, description changed, assignee modified).

    - The previous value and the new value for each change.

Postconditions: None

Expected Result: The task history should accurately display all modifications made to the task, including the type of change, old value, new value, and timestamp for each change.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TH-002

Test Case Title: Verify that the task history includes modifications, comments, and other relevant activity.

Test Case Description: This test verifies that the task history captures not only field modifications but also other relevant activities like comments added, files attached, or status changes made to the task.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in.

- A task exists with at least one comment, attachment, or status change.

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page of the task.

2. Locate the "Task History" section.

3. Verify that the history displays entries for:

    - Comments added to the task.

    - Files attached to the task.

    - Status changes of the task (e.g., To Do, In Progress, Completed).

4. Verify that each entry includes relevant details, such as the comment text, file name, or the new status.

Postconditions: None

Expected Result: The task history should accurately capture and display all modifications, comments, attachments, and status changes made to the task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TH-003

Test Case Title: Verify that users can filter the history by type of change or date.

Test Case Description: This test case verifies the functionality of filtering the task history, allowing users to narrow down their view based on the type of change or the date range.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in.

- A task exists with a history containing various types of changes and spanning different dates.

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page of the task.

2. Locate the "Task History" section.

3. Verify the presence of filter options, such as:

    - Filter by type of change (e.g., comments, attachments, status changes, field updates).

    - Filter by date range (e.g., last 24 hours, last week, custom date range).

4. Apply different filter combinations.

5. Verify that the displayed history is correctly filtered based on the selected criteria.

Postconditions: None

Expected Result: The filtering options should function correctly, allowing users to view a subset of the task history based on their chosen filters.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TH-004

Test Case Title: Verify that the system provides timestamps for all changes in the task history.

Test Case Description: This test case ensures that each entry in the task history is accompanied by a timestamp, allowing users to track the exact time of each modification or activity.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in.

- A task exists with a history of changes.

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page of the task.

2. Locate the "Task History" section.

3. For each entry in the history, verify that a timestamp is displayed.

4. Verify that the timestamp format is consistent and readable.

Postconditions: None

Expected Result: Each entry in the task history should clearly display a timestamp indicating when the change or activity occurred.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TH-005

Test Case Title: Verify that task history is accessible from the task details page.

Test Case Description: This test case verifies that users can easily access the task history from the task details page, ensuring a smooth user experience.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in.

- A task exists.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.

2. Click on a task to access its details page.

3. Verify that a clearly labeled section for "Task History" or a similar designation is present on the page.

Postconditions: None

Expected Result: The task history section should be easily accessible from the task details page, ideally without requiring any additional clicks or navigation.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Usability 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 Cases for Task Management Tool - Task Notifications Feature

Test Case ID: TN-001

Test Case Title: Verify that users receive notifications for task updates.

Test Case Description: This test case verifies that users are notified via their preferred notification channels (email, mobile push, in-app) when a task they are assigned to is updated.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged into the application.
- User has enabled notifications for task updates.
- User is assigned to at least one task.

Test Data: None

Test Steps:

1. As another user, navigate to a task assigned to the user under test.
2. Update the task details (e.g., change due date, mark as complete).
3. Verify that the user receives a notification via their preferred channel.
4. Check the notification details to confirm accurate information is displayed (task name, update type, update author).

Postconditions: None

Expected Result: The user should receive a timely notification about the task update with accurate information.

Severity: Critical

Priority: High

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case ID: TN-002

Test Case Title: Verify that notifications are delivered via email, mobile push notifications, or in-app alerts.

Test Case Description: This test case ensures users can receive task notifications through different delivery channels based on their preferences.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged into the application.
- User has configured their notification preferences for each channel.

Test Data: None

Test Steps:

1. Trigger a task update that would generate a notification (e.g., add a comment).
2. Check for the notification delivery through each enabled channel (email, mobile push, in-app alert).
3. Verify the notification content is consistent across all channels.

Postconditions: None

Expected Result: Notifications should be delivered successfully through all enabled channels with consistent content.

Severity: Major

Priority: High

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case ID: TN-003

Test Case Title: Verify that users can customize notification settings based on preferences.

Test Case Description: This test case ensures users can personalize their notification settings, selecting their preferred delivery channels and notification types.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged into the application.

Test Data: None

Test Steps:

1. Navigate to the user's notification settings.

2. Verify that options are available to enable/disable notifications for specific actions (task updates, comments, mentions).

3. Verify that users can choose their preferred delivery channels for each notification type.

4. Save the changes and trigger an action that would generate a notification.

5. Confirm that notifications are delivered according to the updated preferences.

Postconditions: None

Expected Result: Users should be able to customize their notification settings, and notifications should be delivered according to these preferences.

Severity: Major

Priority: Medium

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: TN-004

Test Case Title: Verify that notifications provide detailed information about the activity.

Test Case Description: This test case ensures that notifications provide comprehensive information about the triggering event, including the task, the user who initiated the action, and the specific changes made.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged into the application.

- User has enabled notifications for relevant activities.

Test Data: None

Test Steps:

1. Trigger different types of task activities (update description, add comment, mention a user).

2. For each activity, check the received notifications on all channels.

3. Verify that notifications include relevant details: task name, activity type, user who initiated the action, timestamp, and specific changes made.

Postconditions: None

Expected Result: Each notification should contain clear and detailed information about the corresponding task activity.

Severity: Minor

Priority: Medium

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TN-005

Test Case Title: Verify that the system logs notification history for future reference.

Test Case Description: This test case ensures the system maintains a log of all sent notifications, allowing users to review past activity and track the communication history of tasks.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged into the application.

- User has received several notifications.

Test Data: None

Test Steps:

1. Navigate to the user's notification history section.

2. Verify that a log of past notifications is displayed.

3. Check if the log includes essential information: notification type, date and time, task name, and a link to access the task.

4. Verify that users can filter or sort the notification history based on different criteria.

Postconditions: None

Expected Result: The system should accurately record and display the user's notification history, including relevant details and functionalities for easy review.

Severity: Minor

Priority: Low

Type of Testing: Functional

Test Case Approach: Positive

Data - 39

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Archiving

Feature description -

Completed or inactive tasks can be archived or moved to a separate section, decluttering the task list and maintaining focus on active tasks. Task archiving helps users keep their task lists organized and prioritized, without losing access to important historical data.

Userstory title -

As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks.

Acceptance criteria -

['Users can archive completed or inactive tasks.', 'Archived tasks are moved to a separate section for easy access.', 'Users can search and view archived tasks.', 'The system provides an option to restore archived tasks to the active list.', 'Archived tasks retain all original details and history.']

Generated Test Cases:

## Test Cases for Task Archiving Feature

Test Case ID: TA-001

Test Case Title: Verify that users can archive completed tasks.

Test Case Description: This test case verifies the functionality of archiving a completed task. It ensures users can select and archive a completed task from their task list.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in.

- User has at least one completed task.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'My Tasks' section.

2. Select a task that is marked as 'Completed'.

3. Click on the 'Archive Task' button (or equivalent action).

4. Confirm the archiving action if prompted.

Postconditions:

- The selected completed task should be removed from the 'My Tasks' list.

Expected Result: The completed task is successfully archived and removed from the main task list.

Severity: Blocker

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TA-002

Test Case Title: Verify that archived tasks are moved to a separate section for easy access.

Test Case Description: This test case verifies that once a task is archived, it is moved to a dedicated 'Archived Tasks' section and is accessible to the user.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in.

- User has archived at least one task.

Test Data: No test data needed

Test Steps:

1. Navigate to the task management tool.

2. Locate and select the 'Archived Tasks' section (or equivalent).

Postconditions: None

Expected Result: The system displays a list of all archived tasks belonging to the user.

Severity: Blocker

Priority: High

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: TA-003

Test Case Title: Verify that users can search and view archived tasks.

Test Case Description: This test case ensures the search functionality within the 'Archived Tasks' section is working correctly and users can find specific archived tasks.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in.

- The 'Archived Tasks' section contains multiple entries.

Test Data:

- Search keyword: Use a keyword from an existing archived task title or description.

Test Steps:

1. Navigate to the 'Archived Tasks' section.

2. Enter the search keyword into the search bar.

3. Initiate the search.

Postconditions: None

Expected Result: The system displays only the archived tasks matching the entered keyword.

Severity: Critical

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TA-004

Test Case Title: Verify that the system provides an option to restore archived tasks to the active list.

Test Case Description: This test case ensures users can restore previously archived tasks back to their active task list.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in.

- User has at least one archived task.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Archived Tasks' section.

2. Select an archived task.

3. Click on the 'Restore Task' button (or equivalent action).

4. Confirm the restoration if prompted.

Postconditions:

- The selected task should be removed from the 'Archived Tasks' section.

Expected Result: The selected archived task is restored to the user's active task list ('My Tasks').

Severity: Critical

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TA-005

Test Case Title: Verify that archived tasks retain all original details and history.

Test Case Description: This test case ensures that when a task is archived and later restored, all its original details (description, attachments, comments, deadlines, etc.) are preserved.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in.

- User has at least one archived task with details and history (attachments, comments, etc.).

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Archived Tasks' section.

2. Select an archived task with details and history.

3. View the details and history of the selected archived task.

4. Compare the displayed details and history with the original information before archiving.

Postconditions: None

Expected Result: All original details, including attachments, comments, deadlines, and any other task history, should be present and accurately reflect the information before archiving.

Severity: Critical

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Data - 40

Project name -

Task Management Tool

Project description -

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

Feature name -

Guest Access

Feature description -

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

Userstory title -

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

Acceptance criteria -

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

Generated Test Cases:

## Test Cases for Guest Access Feature in Task Management Tool

Test Case ID: TA-001

Test Case Title: Verify that users can invite external collaborators as guests with limited access.

Test Case Description: This test case aims to validate the guest invitation process. It verifies that a registered user can successfully invite an external collaborator as a guest with limited access to the platform.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged into the task management tool.
- User has an active project created.

Test Data:

- Guest Email: guestuser@example.com

Test Steps:

1. Navigate to the project settings or a dedicated "Guest Access" section.
2. Select the option to invite a guest.
3. Enter the email address of the guest user (guestuser@example.com).
4. Select the specific task or project for which access is being granted.
5. Send the invitation.

Postconditions:

- A guest invitation should be sent to the specified email address.
- The invited guest should appear in the project's guest list.

Expected Result: The guest user should receive an email invitation to join the platform with limited access to the specified task or project.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TA-002

Test Case Title: Verify that guests can view and collaborate on specific tasks or projects.

Test Case Description: This test verifies that guest users have the appropriate level of access to view and collaborate on the specific tasks or projects they have been granted access to.

Test Suite: Guest Access

Test Priority: High

Preconditions:
- Guest user has been invited and has accepted the invitation.
- Guest user is logged into the task management tool.

Test Data: No test data needed

Test Steps:

1. Log in as the guest user.

2. Navigate to the task or project that access was granted for.

3. Verify the guest user can view task details, including descriptions, attachments, and comments.

4. Attempt to add a comment or update task details as the guest user.

Postconditions: No specific postconditions.

Expected Result: The guest user should be able to view all details of the task or project and participate in collaboration features like adding comments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TA-003

Test Case Title: Verify that the system provides clear distinctions between guest and full access roles.

Test Case Description: This test ensures that the system visually and functionally differentiates between users with full access and those with guest access.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- Have both a regular user account and a guest user account set up.

Test Data: No test data needed

Test Steps:

1. Log in as a regular user with full access.
2. Observe the user interface, menus, and available features.
3. Log out and then log in as the guest user.
4. Observe the user interface, menus, and available features for the guest user.
5. Compare the experiences to identify any discrepancies in accessible features or visual cues.

Postconditions: No specific postconditions.

Expected Result: The guest user interface should clearly indicate limited access, potentially with visual cues and restricted access to features not available to guests.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TA-004

Test Case Title: Verify that users can manage and revoke guest access at any time.

Test Case Description: This test confirms that users who granted guest access can manage those permissions, including revoking access when necessary.

Test Suite: Guest Access

Test Priority: High

Preconditions:
- User has previously granted guest access to one or more users.

Test Data: No test data needed

Test Steps:
1. Navigate to the project settings or "Guest Access" section.
2. Locate the list of guests with access to the project.
3. Select the option to revoke access for a specific guest user.
4. Confirm the revocation.

Postconditions:
- The guest user should be removed from the project's guest list.

Expected Result: The guest user should no longer be able to access the task or project.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TA-005

Test Case Title: Verify that guests cannot access sensitive or restricted areas of the platform.

Test Case Description: This test case aims to validate the access restrictions imposed on guest users, ensuring they are unable to access sensitive data, settings, or features not explicitly granted to them.

Test Suite: Guest Access

Test Priority: High

Preconditions:
- Guest user is logged into the platform.

Test Data: No test data needed

Test Steps:

1. Attempt to access sensitive areas such as:
    - Project settings (for projects the guest doesn't have access to)
    - User management sections
    - Billing or account information
    - Administrative features
2. Observe the system's response to each access attempt.

Postconditions: No specific postconditions.

Expected Result: The system should deny access to all sensitive or restricted areas, preventing guest users from viewing or modifying information outside their permitted scope. Error messages or access restriction notifications should be displayed.

Severity: Critical

Priority: High

Type of Testing: Security Testing, Functional Testing, Negative Testing

---

Data - 41

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Import/Export

Feature description -

Users can import tasks from external sources or export tasks to other task management tools or file formats. Import/export capabilities facilitate data migration and integration with other systems, ensuring seamless collaboration and workflow management across platforms.

Userstory title -

As a user, I want to import and export tasks so that I can migrate data between different task management tools.

Acceptance criteria -

['Users can import tasks from external sources in various formats (e.g., CSV, Excel).', 'Users can export tasks to other task management tools or file formats.', 'The system provides clear mapping of fields during import/export.', 'Users can preview imported data before finalizing the process.', 'The system logs import/export activities for future reference.']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Import/Export Feature

Test Case ID: TMTI-001

Test Case Title: Verify that users can import tasks from a CSV file.

Test Case Description: This test case aims to verify the functionality of importing tasks from a CSV file, ensuring correct data mapping and successful import.

Test Suite: Task Import

Test Priority: High

Preconditions:

* The user is logged into the Task Management Tool.

* The user has access to the task import feature.

* A valid CSV file containing task data is available.

Test Data: Sample CSV file with task data (Title, Description, Due Date, Assignee, etc.)

Test Steps:

1. Navigate to the task import section of the application.

2. Select "CSV" as the import file format.

3. Upload the prepared CSV file.

4. Map the CSV columns to corresponding fields in the tool (e.g., Title to Task Name, Due Date to Deadline).

5. Initiate the import process.

Postconditions:

* Verify that all tasks from the CSV file are imported successfully.

* Imported tasks should appear in the user's task list with the correct data.

* Verify that the import activity is logged in the system.

Expected Result: The tool should successfully import all tasks from the CSV file, accurately mapping data fields, and displaying them in the user's task list.

Severity: Critical

Priority: High

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: TMTI-002

Test Case Title: Verify that users can export tasks to an Excel spreadsheet.

Test Case Description: This test case verifies the functionality of exporting tasks to an Excel spreadsheet, ensuring accurate data formatting and completeness.

Test Suite: Task Export

Test Priority: High

Preconditions:
* The user is logged into the Task Management Tool.
* The user has tasks present in their task list.
* The user has access to the task export feature.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.

2. Select the tasks to be exported.

3. Choose "Export" and select "Excel" as the export file format.

4. Initiate the export process.

Postconditions:

* Verify that the system generates an Excel file containing the selected tasks.

* Open the downloaded Excel file.

* Verify that the exported data accurately reflects the task information, including all relevant fields (Title, Description, Due Date, Status, etc.).

Expected Result: The tool should successfully export the selected tasks into a well-formatted Excel spreadsheet, maintaining data integrity and structure.

Severity: Critical

Priority: High

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: TMTI-003

Test Case Title: Verify that the system provides clear and intuitive field mapping during import.

Test Case Description: This test case ensures that the import process offers a user-friendly interface for mapping data fields from the source file to the application's task fields.

Test Suite: Task Import

Test Priority: High

Preconditions:
* The user is logged into the Task Management Tool.
* The user is in the process of importing tasks from a file (CSV or Excel).

Test Data: Sample CSV or Excel file with task data.

Test Steps:
1. Begin the task import process and upload a sample file.
2. Observe the field mapping interface.
3. Verify that the interface clearly displays source file columns and corresponding application fields.
4. Check if the tool provides drag-and-drop functionality or clear selection mechanisms for mapping.
5. Attempt to map different source columns to target fields.

Postconditions: None

Expected Result: The import interface should be intuitive and easy to understand, allowing users to clearly see and map data fields between the source file and the Task Management Tool.

Severity: Major

Priority: High

Type of Testing: Usability, Functional

Test Case Approach: Positive

Test Case ID: TMTI-004

Test Case Title: Verify that users can preview imported data before finalizing the import process.

Test Case Description: This test case ensures that users can preview the imported data and identify any mapping errors before the data is permanently added to the system.

Test Suite: Task Import

Test Priority: High

Preconditions:
* The user is logged into the Task Management Tool.
* The user is in the process of importing tasks and has mapped data fields.

Test Data: Sample CSV or Excel file with task data.

Test Steps:
1. During the import process, after mapping fields, look for a "Preview" or "Validate" option.

2. Click on the "Preview" button.

Postconditions: None

Expected Result: The system should display a preview of how the imported data will be structured within the application, allowing users to review and identify any potential mapping issues before final import.

Severity: Major

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTI-005

Test Case Title: Verify that the system logs import/export activities for future reference.

Test Case Description: This test case checks whether the system correctly logs all import and export actions, including timestamps, user details, and file information.

Test Suite: Task Import/Export, System Logs

Test Priority: Medium

Preconditions:

* The user is logged into the Task Management Tool.

* The user has performed import or export actions.

Test Data: No test data needed.

Test Steps:

1. Navigate to the system logs section (or audit trail) within the application.

2. Look for logs related to import and export activities.

Postconditions: None

Expected Result: The system logs should accurately record all import/export activities, including the timestamp of the action, the user who performed it, the file name (if applicable), and the status (success or failure) of the operation.

Severity: Minor

Priority: Medium

Type of Testing: Security, Non-Functional

Test Case Approach: Positive

Data - 42

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.

Userstory title -

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

Acceptance criteria -

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.', 'Priority changes are logged and tracked.']

Generated Test Cases:

## Test Cases for Task Prioritization Feature

Test Case ID: TP-001

Test Case Title: Verify that users can set and adjust task priorities.

Test Case Description: This test case verifies the functionality of setting and adjusting priorities for newly created and existing tasks using the task management tool.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

* The user is logged into the task management tool.

* The user has at least one existing task.

Test Data: No test data needed.

Test Steps:

1. Create a new task.

2. Set the task priority to "High".

3. Save the task.

4. Locate the previously created task.

5. Adjust the task priority to "Low".

6. Save the changes.

Postconditions:

* The newly created task should display the set "High" priority.

* The existing task should reflect the adjusted "Low" priority.

Expected Result: The user should be able to successfully set and adjust task priorities.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TP-002

Test Case Title: Verify that tasks can be sorted and filtered based on priority levels.

Test Case Description: This test case ensures users can effectively organize their task list by sorting and filtering tasks based on assigned priority levels.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

* The user is logged into the task management tool.

* The user has created multiple tasks with different priority levels (e.g., High, Medium, Low).

Test Data: No test data needed.

Test Steps:

1. Navigate to the main task list view.

2. Locate the sorting/filtering options for task priority.

3. Sort the task list by "High" priority.

4. Verify that tasks are displayed in descending order of priority, with "High" priority tasks at the top.

5. Filter the task list to show only "Medium" priority tasks.

6. Verify that only tasks with "Medium" priority are displayed.

Postconditions: None

Expected Result: The task list should accurately reflect the selected sorting and filtering criteria based on task priority.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TP-003

Test Case Title: Verify that the system provides visual indicators of task priority.

Test Case Description: This test case checks the presence and clarity of visual cues that represent different task priority levels, enhancing user recognition and task management.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

* The user is logged into the task management tool.

* The user has created tasks with different priority levels.

Test Data: No test data needed.

Test Steps:

1. Navigate to the main task list view.

2. Observe the visual representation of tasks with different priority levels.

3. Verify that visual indicators (e.g., color coding, icons) clearly differentiate between "High," "Medium," and "Low" priority tasks.

4. Verify that the chosen visual indicators are intuitive and easily understandable.

Postconditions: None

Expected Result: The system should provide clear and distinct visual indicators for each task priority level.

Severity: Minor

Priority: Medium

Type of Testing: User Interface Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TP-004

Test Case Title: Verify that users receive notifications for high-priority tasks.

Test Case Description: This test case verifies the functionality of the notification system to alert users about upcoming deadlines or updates related to high-priority tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

* The user is logged into the task management tool.

* The user has enabled notifications for high-priority tasks.

* A high-priority task with an approaching deadline exists.

Test Data: No test data needed.

Test Steps:

1. Set a task to "High" priority and set a near-future deadline.

2. Wait for the notification trigger time.

3. Verify that the user receives a notification regarding the approaching deadline of the high-priority task.

4. Check the content of the notification to ensure it contains relevant task information (e.g., task name, deadline).

Postconditions: None

Expected Result: The user should receive a timely and informative notification about the high-priority task.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Test Case ID: TP-005

Test Case Title: Verify that priority changes are logged and tracked.

Test Case Description: This test case confirms that any changes made to a task's priority are recorded and accessible for auditing or review purposes, ensuring transparency and accountability.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

* The user is logged into the task management tool.

* A task exists with an assigned priority.

Test Data: No test data needed.

Test Steps:

1. Locate an existing task and note its current priority level.

2. Change the task's priority to a different level.

3. Access the task's activity log or history.

4. Verify that the log records the change in priority, including the previous and new priority levels, timestamp, and the user who made the change.

Postconditions: None

Expected Result: All changes to task priorities should be accurately logged and readily accessible within the system.

Severity: Minor

Priority: Medium

Type of Testing: Auditing & Logging, Security 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 Cases for Task Management Tool - Task Collaboration Feature

Test Case ID: TC_TC_001

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

Test Case Description: This test case verifies the functionality of assigning a new task to a team member within the Task Management Tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged into the application.

* The user has an existing team with at least one other member.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Create Task' page.

2. Fill in the required task details (title, description, etc.).

3. In the 'Assignee' field, select a team member from the dropdown list.

4. Save the task.

Postconditions:

* The assigned team member should appear in the 'Assignee' field of the created task.

* The assigned team member should receive a notification about the assigned task.

Expected Result: The task should be successfully assigned to the selected team member.

Severity: Blocker

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_TC_002

Test Case Title: Verify that users can share updates and communicate through comments or messages.

Test Case Description: This test case verifies the functionality of adding comments and updating task details to facilitate communication between team members.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged into the application.

* The user has an existing task assigned to them or another team member.

Test Data:

* Comment text: "This is a test comment."

Test Steps:

1. Navigate to the details page of the assigned task.

2. Locate the 'Comments' or 'Updates' section.

3. Enter the test comment text.

4. Click the 'Add Comment' or 'Post Update' button.

Postconditions:

* The added comment should appear in the task's 'Comments' or 'Updates' section.

* The assigned team members should receive a notification about the new comment.

Expected Result: The comment should be successfully added to the task and visible to all assigned team members.

Severity: Critical

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_TC_003

Test Case Title: Verify that task updates and comments are visible to assigned team members.

Test Case Description: This test case checks the visibility of added comments and task updates for all assigned team members on the task details page.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged in as an assigned team member.

* A task exists with comments or updates added by another team member.

Test Data: No test data needed.

Test Steps:

1. Navigate to the details page of the assigned task.

2. Locate the 'Comments' or 'Updates' section.

Postconditions: None

Expected Result: All comments and updates added by any team member should be displayed chronologically in the 'Comments' or 'Updates' section.

Severity: Critical

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_TC_004

Test Case Title: Verify that users receive notifications for task updates and comments.

Test Case Description: This test case ensures that users receive appropriate notifications for new comments, task updates, and assignments within the application and via email (if configured).

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged into the application.

* The user has enabled notifications for task updates and comments.

Test Data:

* Comment text: "This is a test comment for notification."

Test Steps:

1. Have another team member add a comment to a task assigned to the user.

2. Observe the notification area within the application.

3. Check the user's registered email inbox for any notification emails.

Postconditions: None

Expected Result: The user should receive a notification within the application and via email (if configured) about the new comment on the assigned task.

Severity: Major

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_TC_005

Test Case Title: Verify that the system provides a clear view of task collaboration history.

Test Case Description: This test case checks the presentation and completeness of task collaboration history, including all actions taken by team members on the task.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

* The user is logged into the application.

* The user has an existing task with a history of collaborations (comments, updates, assignments).

Test Data: No test data needed.

Test Steps:

1. Navigate to the details page of the task with collaboration history.

2. Locate the 'History' or 'Activity Log' section for the task.

Postconditions: None

Expected Result: The 'History' or 'Activity Log' section should display a chronological record of all actions related to the task, including:

   * Comments added

   * Task updates made

   * Task assignments and reassignments

   * Timestamps for each action

   * The user who performed each action

Severity: Major

Priority: Medium

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Data - 44

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tracking

Feature description -

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

Userstory title -

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

Acceptance criteria -

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.', 'System provides visual indicators for task progress.', '']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Tracking Feature

Test Case ID: TT-001

Test Case Title: Verify that users can view task statuses.

Test Case Description: This test case verifies that users can view the status of their assigned tasks. This includes verifying that different task statuses (e.g., "To Do," "In Progress," "Completed") are clearly displayed and that users can easily identify the status of each task.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

* The user is logged into the application.

* The user has at least one task assigned to them.

Test Data: No test data needed

Test Steps:

1. Navigate to the "My Tasks" view.

2. Observe the displayed list of tasks.

3. Verify that each task displays a clear and distinct status label.

Postconditions: None

Expected Result: The status of each task (e.g., "To Do," "In Progress," "Completed") is clearly visible and easily identifiable for each assigned task.

Severity: Major

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TT-002

Test Case Title: Verify that users can track task progress and deadlines.

Test Case Description: This test case verifies that users can track the progress of a task toward completion and that the task deadline is clearly displayed.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

* The user is logged into the application.

* The user has at least one task assigned to them.

Test Data: No test data needed

Test Steps:

1. Navigate to the details view of a specific task.

2. Verify a progress indicator (e.g., progress bar) is displayed for the task.

3. Verify the task deadline is clearly visible.

Postconditions: None

Expected Result: A visual progress indicator accurately reflects the task's completion status. The task deadline is clearly displayed within the task details.

Severity: Major

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TT-003

Test Case Title: Verify that tasks are updated in real-time.

Test Case Description: This test case verifies that any changes made to a task's status, progress, or other relevant details are immediately reflected in the user interface for all users associated with the task.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

* The user is logged into the application.

* The user has at least one task assigned to them.

Test Data:

* Updated Task Status: "In Progress"

Test Steps:

1. Open the details view of a specific task.

2. Have another user (or use a separate session) simultaneously update the task status to "In Progress."

3. Observe the task details view in the first session.

Postconditions: Revert the task status to its original state.

Expected Result: The task status in the first session should update to "In Progress" in real-time without requiring a manual refresh.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Real-time update Testing

Test Case Approach: Positive

Test Case ID: TT-004

Test Case Title: Verify that the system provides visual indicators for task progress.

Test Case Description: This test case ensures the application utilizes clear and intuitive visual cues to represent the progress of a task. This can include progress bars, percentage indicators, or color-coded systems.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

* The user is logged into the application.

* The user has at least one task assigned to them.

Test Data: No test data needed

Test Steps:

1. Navigate to the "My Tasks" view.

2. Observe the display of tasks.

3. Verify that each task utilizes a visual indicator to represent its progress.

4. Check if the visual indicators are clear, intuitive, and easy to understand.

Postconditions: None

Expected Result: Tasks should display clear visual progress indicators like progress bars, percentage completion, or color-coding to represent their status.

Severity: Minor

Priority: Medium

Type of Testing: UI Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TT-005

Test Case Title: Verify that overdue tasks are visually highlighted.

Test Case Description: This test case ensures that the system provides a clear visual indicator for tasks that have passed their deadlines. This could be achieved through color-coding, specific icons, or other visual cues.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

* The user is logged into the application.

* The user has at least one overdue task assigned to them.

Test Data: No test data needed

Test Steps:

1. Navigate to the "My Tasks" view.

2. Locate a task that has passed its deadline.

3. Observe the visual representation of the overdue task.

Postconditions: None

Expected Result: The overdue task is visually distinct from other tasks, utilizing a clear visual indicator like color-coding (e.g., red), an icon (e.g., exclamation mark), or a similar method.

Severity: Major

Priority: High

Type of Testing: UI Testing, Functional Testing

Test Case Approach: Positive

Data - 45

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

Userstory title -

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

Acceptance criteria -

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.', 'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', '']

Generated Test Cases:

## Test Case Creation

Test Case ID: TC_NTF_001

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

Test Case Description: This test case verifies that users receive a notification when a new task is assigned to them. This includes checking the notification delivery mechanism (e.g., in-app, email) and the content of the notification message.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User A and User B are registered in the task management tool.

- User A has the permission to create and assign tasks.

Test Data: No test data needed

Test Steps:

1. Log in as User A.

2. Create a new task.

3. Assign the task to User B.

4. Save the task.

5. Log in as User B.

Postconditions: None

Expected Result: User B receives a notification about the new task, including the task name, assigned by, and due date (if any). The notification should appear in their preferred notification channel (in-app or email).

Severity: Major

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_NTF_002

Test Case Title: Verify that users receive notifications for task updates and comments.

Test Case Description: This test case aims to verify that users receive appropriate notifications when a task they are involved in is updated or receives new comments. This includes changes to the task description, due date, priority, or status, as well as new comments added by other users.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- A task exists and is assigned to User B.

- User A is a collaborator on the task.

Test Data: No test data needed

Test Steps:

1. Log in as User A.

2. Open the existing task.

3. Make an update to the task (e.g., change the due date, add a comment).

4. Save the changes.

5. Log in as User B.

Postconditions: None

Expected Result: User B receives a notification about the task update or comment. The notification should specify the type of update (comment, due date change, etc.) and who made the change.

Severity: Major

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_NTF_003

Test Case Title: Verify that users receive reminders for upcoming deadlines.

Test Case Description: This test case ensures the system effectively sends timely reminders to users about approaching task deadlines. This includes checking the timing of reminders (e.g., one day before, one hour before) and the accuracy of the information provided in the reminder notification.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- A task is assigned to User B with a future due date.

Test Data:

- Due date: Current date + 1 day

Test Steps:

1. Configure system time to be one day before the task due date.

2. Log in as User B.

3. Wait for the system to generate and send the reminder notification.

Postconditions: Revert system time to default.

Expected Result: User B receives a notification reminding them about the upcoming task deadline. The notification should include the task name, due date, and any relevant details.

Severity: Major

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_NTF_004

Test Case Title: Verify that notification settings can be customized by users.

Test Case Description: This test validates the functionality allowing users to customize their notification preferences. This involves checking if users can choose their preferred notification methods (in-app, email), adjust the frequency of notifications, and select specific event types they want to be notified about.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User B is logged in.

Test Data: No test data needed

Test Steps:

1. Navigate to the user settings.

2. Locate the notification settings section.

3. Verify the available options for customizing notification preferences (notification method, frequency, event types).

4. Modify different notification settings.

5. Save the changes.

Postconditions: None

Expected Result: The system successfully saves the user's chosen notification preferences. Users should only receive notifications according to their customized settings.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_NTF_005

Test Case Title: Verify that users do not receive notifications for irrelevant task events.

Test Case Description: This test ensures that users only receive notifications for events directly related to their tasks and involvement. It verifies that the system correctly filters out notifications for tasks the user is not assigned to, not collaborating on, and not participating in any relevant activities.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User A and User B are registered.

- User A creates a task but does not assign it to User B.

Test Data: No test data needed

Test Steps:

1. Log in as User B.

2. Monitor the notification center or email inbox.

Postconditions: None

Expected Result: User B should not receive any notifications about the task created by User A, as they are not involved in that specific task.

Severity: Minor

Priority: Medium

Type of Testing: Functional 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 Cases for Task Categorization Feature:

Test Case ID: TC_001

Test Case Title: Verify that users can create and assign tasks to different categories.

Test Case Description: This test case verifies that a user can successfully create new categories and assign newly created or existing tasks to these categories.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* The user has access to create new tasks.

Test Data:

* Task Name: "Prepare Presentation"

* Category Name: "Work Projects"

Test Steps:

1. Navigate to the 'Tasks' section.

2. Click on 'Create New Task'.

3. Fill in the task details (e.g., name, description).

4. Click on the "Category" dropdown menu.

5. Select "Create New Category".

6. Enter the new category name ("Work Projects").

7. Save the new category.

8. Select the newly created category for the task.

9. Save the task.

Postconditions:

* Verify that the new category appears in the category list.

* Verify that the new task is assigned to the "Work Projects" category.

Expected Result: The task "Prepare Presentation" is successfully created and assigned to the "Work Projects" category.

Severity: Critical

Priority: High

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case ID: TC_002

Test Case Title: Verify that users can view tasks grouped by categories.

Test Case Description: This test case verifies that the application correctly displays tasks grouped under their assigned categories, allowing users to easily filter and view tasks.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* Several tasks are created and assigned to different categories.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Tasks' section.

2. Click on the "Category" filter or dropdown menu.

3. Select a specific category from the available options.

Postconditions: None

Expected Result: The task list displays only the tasks belonging to the selected category.

Severity: Critical

Priority: High

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case ID: TC_003

Test Case Title: Verify that tasks can be moved between categories.

Test Case Description: This test case ensures that users have the flexibility to re-categorize tasks by moving them from one category to another.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* At least two categories exist with tasks assigned to them.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Tasks' section.

2. Select a task that needs to be moved.

3. Click on the "Edit Task" or "Move" option (depending on the UI).

4. From the "Category" dropdown, choose the new desired category.

5. Save the changes.

Postconditions:

* Verify the task is no longer listed under its original category.

* Verify the task now appears under the new category.

Expected Result: The selected task is successfully moved from its original category to the newly chosen category.

Severity: Major

Priority: High

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case ID: TC_004

Test Case Title: Verify that category information is accurately reflected in task views.

Test Case Description: This test case ensures that when viewing individual tasks, the displayed category information is consistent with the category assigned to that specific task.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

* User is logged into the Task Management Tool.

* A task is created and assigned to a specific category.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Tasks' section.

2. Open a specific task to view its details.

Postconditions: None

Expected Result: The task details view accurately displays the name of the category to which the task is assigned.

Severity: Major

Priority: Medium

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case ID: TC_005

Test Case Title: Verify that deleting a category removes the category from all tasks.

Test Case Description: This test case verifies that when a user deletes a category, all tasks previously associated with that category are updated correctly, either by removing the category association or by handling it based on predefined system behavior.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

* User is logged into the Task Management Tool.

* At least one category exists with tasks assigned to it.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Categories' or 'Projects' section.

2. Choose the category to be deleted.

3. Initiate the delete action (e.g., click 'Delete', use a context menu).

4. Confirm the deletion if prompted.

5. Navigate to the 'Tasks' section.

6. Check tasks previously assigned to the deleted category.

Postconditions:

* Verify that the deleted category is not present in the category list.

Expected Result: The deleted category should no longer be associated with any tasks. The tasks might either have their category removed, be assigned a default category, or be handled according to the system's design for handling category deletion.

Severity: Critical

Priority: High

Type of Testing: Functional, System Testing

Test Case Approach: Destructive

Data - 47

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Search

Feature description -

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

Userstory title -

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Search Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
TMT-TS-001	Verify that users can search for tasks by name.	This test case verifies the functionality of searching for tasks using the task name as the search criteria.	Task Search	High	- User is logged in. - User has created at least one task.	Task Name: "Prepare Presentation"	1. Navigate to the task list view. 2. Enter "Prepare Presentation" in the search bar. 3. Press Enter or click the search icon.	None	The system should display all tasks with the name "Prepare Presentation" in the search results. No other tasks should be displayed.	Critical	High	Functional, Regression	Positive
TMT-TS-002	Verify that users can search for tasks by description.	This test case verifies the functionality of searching for tasks using keywords from the task description.	Task Search	High	- User is logged in. - User has created at least one task with a description.	Task Description: "Prepare slides for the upcoming product launch presentation." Search Keyword: "product launch"	1. Navigate to the task list view. 2. Enter "product launch" in the search bar. 3. Press Enter or click the search icon.	None	The system should display all tasks that contain "product launch" in their description within the search results.	Critical	High	Functional, Regression	Positive
TMT-TS-003	Verify that users can filter search results by category, priority, or due date.	This test case verifies the ability to refine search results using filters like category, priority, and due date.	Task Search	High	- User is logged in. - User has created at least three tasks with varying categories, priorities, and due dates.	- Category: "Marketing" - Priority: "High" - Due Date: Within one week	1. Navigate to the task list view. 2. Perform a general search (optional). 3. Apply the filter for "Marketing" category. 4. Apply the filter for "High" priority. 5. Apply the filter for tasks due within one week.	None	The system should display only the tasks that match all three applied filters: "Marketing" category, "High" priority, and due within one week.	Major	High	Functional, Regression	Positive
TMT-TS-004	Verify that search results are accurate and relevant.	This test case aims to validate the accuracy and relevancy of the displayed search results.	Task Search	High	- User is logged in. - The task								

database contains a variety of tasks. | Various search terms related to existing tasks: - Task Names - Description keywords - Category names | 1. Navigate to the task list view. 2. Perform multiple searches using various task names, description keywords, and category names. 3. For each search, observe the displayed results. | None | The search results should accurately reflect the entered search criteria. Only tasks containing the search term in the name, description, or corresponding category should appear. Irrelevant results should be excluded. | Critical | High | Functional, Regression | Positive |
| TMT-TS-005 | Verify the system behavior when no matching tasks are found. | This test case checks the system's response when a search yields no matching tasks. | Task Search | High | - User is logged in. | Search Term: "Non-existent Task" | 1. Navigate to the task list view. 2. Enter "Non-existent Task" (or any term not present in any task details) in the search bar. 3. Press Enter or click the search icon. | None | The system should display a clear message indicating that no tasks match the entered search criteria. The message could be something like, "No tasks found matching your search." | Minor | Medium | Functional, Negative | 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 Cases for Task Management Tool - Task Collaboration Feature

Test Case ID: TC_001

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

Test Case Description: This test case verifies that a user can successfully add a comment to an existing task.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

*   The user is logged into the Task Management Tool.

*   A task has been created.

Test Data: No test data needed.

Test Steps:

1. Navigate to an existing task.
2. Locate the comment section.
3. Enter a comment in the comment box.
4. Click the "Add Comment" button.

Postconditions:

* The comment is displayed below the task.

* The comment is saved and persists after refreshing the page.

Expected Result: The comment should be successfully added to the task.

Severity: Major

Priority: High

Type of Testing: Functional, System, Acceptance

Test Case Approach: Positive

Test Case ID: TC_002

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

Test Case Description: This test case verifies that a user can successfully attach different types of files (document, image) to an existing task.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

*   The user is logged into the Task Management Tool.

*   A task has been created.

Test Data:

*   Sample files: Document (e.g., .docx, .pdf), Image (e.g., .jpg, .png)

Test Steps:

1. Navigate to an existing task.
2. Locate the file attachment section.
3. Click on the "Choose File" or "Browse" button.
4. Select a file from the local system.
5. Click the "Attach" or "Upload" button.

Postconditions:

* The attached file appears in the task?s file list.

* The attached file is downloadable and viewable.

Expected Result: The selected files should be successfully attached to the task.

Severity: Major

Priority: High

Type of Testing: Functional, System, Acceptance

Test Case Approach: Positive

Test Case ID: TC_003

Test Case Title: Verify that users can share tasks with other users.

Test Case Description: This test case checks if a user can share a task with another registered user and if the shared task appears in the recipient's task list.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* The user is logged into the Task Management Tool.

* A task has been created.

* Another registered user exists.

Test Data:

* The email address or username of the user to share the task with.

Test Steps:

1. Navigate to an existing task.

2. Locate the "Share" or "Collaborators" option.

3. Enter the email address or username of the other user.

4. Click the "Share" or "Add" button.

Postconditions:

* The other user receives a notification about the shared task.

* The shared task appears in the recipient user's task list.

Expected Result: The task should be successfully shared with the specified user.

Severity: Major

Priority: High

Type of Testing: Functional, Integration, System, Acceptance

Test Case Approach: Positive

Test Case ID: TC_004

Test Case Title: Verify that adding a comment to a task sends a notification to other collaborators.

Test Case Description: This test case verifies that the task collaboration feature enhances team communication by sending real-time notifications when a comment is added to a shared task.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* Two users are logged into the Task Management Tool.
* A task has been created and shared between the two users.

Test Data: No test data needed.

Test Steps:

1. As User 1, navigate to the shared task.
2. Add a new comment to the task.
3. As User 2, check for a notification indicating a new comment on the shared task.

Postconditions: None

Expected Result: User 2 should receive a notification about the new comment in real time.

Severity: Major

Priority: High

Type of Testing: Functional, Integration, System, Acceptance

Test Case Approach: Positive

Test Case ID: TC_005

Test Case Title: Verify that attempting to attach a file exceeding the size limit displays an error message.

Test Case Description: This test case aims to check the application's error handling when a user attempts to upload a file that exceeds the allowed size limit.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

* The user is logged into the Task Management Tool.
* A task has been created.

Test Data: A file exceeding the defined size limit for attachments (e.g., a 20MB file if the limit is 10MB).

Test Steps:

1. Navigate to an existing task.
2. Locate the file attachment section.
3. Attempt to attach a file that exceeds the defined size limit.

Postconditions: None

Expected Result:

* The system should prevent the file from being uploaded.
* A clear and user-friendly error message should be displayed, indicating the size limit and potentially suggesting compressing the file or contacting an administrator if needed.

Severity: Minor

Priority: Medium

Type of Testing: Negative, Functional, System

Test Case Approach: Negative

Data - 49

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Course Creation Tools

Feature description -

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to structure content effectively, ensuring an engaging and informative learning experience for students.

Userstory title -

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

Acceptance criteria -

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Course Creation Tools

Test Case ID: TC_CC_001

Test Case Title: Verify that instructors can create courses using various multimedia elements.

Test Case Description: This test case verifies that the course creation tools allow instructors to incorporate videos, presentations, quizzes, and assignments into their courses.

Test Suite: Course Creation

Test Priority: High

Preconditions:

* Instructor account is created and logged in.

* Sample video, presentation, quiz, and assignment files are available.

Test Data: Sample course content including video, presentation, quiz, and assignment files.

Test Steps:

1. Navigate to the course creation page.

2. Create a new course and access the course editor.

3. Upload a video file and configure its settings.

4. Insert a presentation file and adjust its display options.

5. Create a quiz with multiple-choice questions.

6. Set up an assignment with submission instructions and grading criteria.

7. Save the course.

Postconditions:

* The course should be saved with all added elements.

* The added elements should function correctly when accessed.

Expected Result: The course creation tools should successfully allow the integration of video, presentation, quiz, and assignment elements within the course.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_CC_002

Test Case Title: Verify that the course creation tools are user-friendly and require minimal technical skills.

Test Case Description: This test case aims to evaluate the user-friendliness of the course creation tools by observing an instructor with minimal technical expertise attempting to use them.

Test Suite: Course Creation

Test Priority: High

Preconditions:

* Instructor account with minimal technical skills is created and logged in.

Test Data: No test data needed.

Test Steps:

1. Provide the instructor with basic instructions on using the course creation tools.

2. Observe the instructor as they attempt to:

    * Create a new course.

    * Add a text module and format the content.

    * Upload an image and resize it.

    * Create a simple quiz.

    * Preview the created content.

3. Note any difficulties or points of confusion encountered by the instructor.

Postconditions: Collect feedback from the instructor regarding the usability of the tools.

Expected Result: The instructor with minimal technical skills should be able to navigate the interface, understand the basic functionalities, and successfully create a simple course.

Severity: Major

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CC_003

Test Case Title: Verify that courses can be previewed before publishing.

Test Case Description: This test case ensures that instructors have the functionality to preview their courses before making them live for students.

Test Suite: Course Creation

Test Priority: High

Preconditions:

* Instructor account is created and logged in.

* A course is created with some content added.

Test Data: Partially or fully created course.

Test Steps:

1. Navigate to the course editor of the created course.

2. Locate and select the "Preview" option.

3. Review the course content in preview mode.

4. Check if all added elements are displayed and functioning correctly.

5. Close the preview mode.

Postconditions:

* Ensure that exiting preview mode does not affect the course content or settings.

Expected Result: The instructor should be able to preview the entire course as a student would experience it, verifying layout, functionality, and content accuracy before publishing.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CC_004

Test Case Title: Verify that instructors can save courses as drafts.

Test Case Description: This test case ensures that instructors can save their work in progress while creating courses without publishing them.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

* Instructor account is created and logged in.

* A new course is created, and some content is added.

Test Data: Partially created course.

Test Steps:

1. Navigate to the course editor of the partially created course.

2. Locate and select the "Save as Draft" option.

3. Verify that a confirmation message appears indicating the course has been saved as a draft.

4. Close the course editor.

5. Navigate to the course dashboard.

6. Verify that the partially created course is listed as a draft.

Postconditions:

* Ensure that the saved draft retains all previously added content.

Expected Result: The course should be saved as a draft, and the instructor should be able to access and edit the draft version later.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CC_005

Test Case Title: Verify that courses can be edited post-publishing.

Test Case Description: This test case ensures instructors can modify course content even after it has been published for student access.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

* Instructor account is created and logged in.

* A course is created, published, and accessible to students.

Test Data: Published course with content.

Test Steps:

1. Navigate to the published course.

2. Make edits to the existing content (e.g., modify text, replace an image, add a new quiz question).

3. Save the changes made to the published course.

4. Verify that the changes are reflected in the live student-facing version of the course.

Postconditions:

* Ensure that the edited content is immediately visible to students enrolled in the course.

Expected Result: Instructors should be able to edit published courses, and the changes should be reflected in the live course for student access.

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 Cases for Interactive Lessons

Test Case ID: IL-001

Test Case Title: Verify that lessons include quizzes, polls, simulations, and discussions.

Test Case Description: This test case verifies the presence and functionality of all required interactive elements within a lesson.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User has accessed a lesson.

Test Data: No test data needed

Test Steps:

1. Navigate to a lesson.

2. Check for the presence of a quiz within the lesson content.

3. Check for the presence of a poll within the lesson content.

4. Check for the presence of a simulation within the lesson content.

5. Check for the presence of a discussion forum or chat feature linked to the lesson.

Postconditions: None

Expected Result: All four interactive elements (quiz, poll, simulation, discussion) are present and accessible within the lesson.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: IL-002

Test Case Title: Verify that interactive elements are integrated seamlessly into the lesson content.

Test Case Description: This test case evaluates the integration of interactive elements within the lesson flow. It ensures the elements fit naturally and contribute to the learning experience rather than disrupting it.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User has accessed a lesson.

Test Data: No test data needed

Test Steps:

1. Navigate through different sections of a lesson.

2. Observe the placement and context of each interactive element.

3. Assess if the elements are introduced logically and contribute to the topic being discussed.

4. Check for smooth transitions between lesson content and interactive components.

Postconditions: None

Expected Result: All interactive elements are meaningfully integrated into the lesson flow, enhancing the learning experience.

Severity: Major

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: IL-003

Test Case Title: Verify that learners can track their progress through interactive activities.

Test Case Description: This test case ensures that learners can monitor their performance and completion status for all interactive components within a lesson.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User has accessed a lesson and interacted with at least one interactive element.

Test Data: No test data needed

Test Steps:

1. Complete a quiz, participate in a poll, engage with a simulation, and contribute to the discussion forum.

2. Check for progress indicators within the lesson interface (e.g., completion bars, checkmarks).

3. Navigate to the user's profile or dashboard.

4. Verify if the completed interactive activities are reflected in the user's overall course progress.

Postconditions: None

Expected Result: Learners can clearly track their progress through interactive activities both within the lesson and on their profile/dashboard.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: IL-004

Test Case Title: Verify that feedback is provided after each interactive component.

Test Case Description: This test case confirms that appropriate feedback is delivered to the learner after interacting with each interactive element, contributing to their understanding and learning experience.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in to the platform.
- User has accessed a lesson.

Test Data: No test data needed

Test Steps:

1. Complete a quiz question.
2. Verify if feedback is provided (correct/incorrect, explanation).
3. Participate in a poll.
4. Check if results are displayed after submission.
5. Interact with a simulation and complete a task.
6. Verify if feedback is given based on the interaction.
7. Contribute to a discussion forum.

8. Check for feedback mechanisms like replies from peers or instructors.

Postconditions: None

Expected Result: Relevant and timely feedback is provided after interacting with each interactive element.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: IL-005

Test Case Title: Verify that interactive lessons can be accessed on various devices.

Test Case Description: This test case ensures the responsiveness and functionality of interactive lessons across different devices and screen sizes.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User has accessed a lesson.

Test Data:

- Devices: Desktop (Windows, macOS), Mobile (Android, iOS), Tablet (iPad, Android tablet)

- Browsers: Chrome, Firefox, Safari, Edge

Test Steps:

1. Access a lesson on a desktop computer using different browsers.

2. Access the same lesson on a mobile phone (Android and iOS) using different browsers.

3. Access the same lesson on a tablet device (iPad and Android tablet) using different browsers.

4. Interact with all interactive elements on each device and browser combination.

Postconditions: None

Expected Result: Interactive lessons are accessible and function correctly on all tested devices and browsers, maintaining layout and usability.

Severity: Major

Priority: High

Type of Testing: Compatibility Testing, Cross-Browser Testing

Test Case Approach: Positive

Data - 51

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Personalized Learning Paths

Feature description -

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

Userstory title -

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

Acceptance criteria -

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

Generated Test Cases:

## Test Cases for Personalized Learning Paths

Test Case ID: PLP-001

Test Case Title: Verify that learners can choose courses and modules based on their interests.

Test Case Description: This test case aims to confirm that the platform allows users to browse and select courses/modules aligned with their personal interests from the available selection.

Test Suite: Personalized Learning Path

Test Priority: High

Preconditions:

- User is logged in.
- User has completed their profile with areas of interest.

Test Data: No specific test data needed.

Test Steps:

1. Navigate to the 'Explore Courses' or 'Find New Modules' section.
2. Observe if the displayed courses/modules align with the user's specified interests.
3. Select and add a course/module aligned with the user's interest to their learning path.
4. Navigate to the 'My Learning Path' section.
5. Confirm the selected course/module is present in the user's learning path.

Postconditions: None

Expected Result: The platform should display courses/modules matching the user's interests, and these selections should be reflected in their personalized learning path.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: PLP-002

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress.

Test Case Description: This test case evaluates the platform's ability to dynamically adapt the user's learning path based on their course/module completion and progress.

Test Suite: Personalized Learning Path

Test Priority: High

Preconditions:

- User is logged in.
- User has an active learning path with at least one course/module in progress.

Test Data: None

Test Steps:

1. Complete a significant portion (e.g., 50%) of an active course/module in the user's learning path.
2. Observe if the learning path reflects the updated progress status of the course/module.
3. Check if the platform suggests new courses/modules based on the completed portion and user's learning path.

Postconditions: None

Expected Result: The learning path should dynamically update to display the user's progress accurately. The system should recommend relevant content based on the updated progress.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: PLP-003

Test Case Title: Verify that learners receive recommendations for new courses and modules.

Test Case Description: This test validates that the platform provides relevant course/module recommendations to the learner based on their personalized learning path, interests, and potentially, their learning goals.

Test Suite: Personalized Learning Path

Test Priority: Medium

Preconditions:

- User is logged in.
- User has an active learning path with at least one completed course/module.

Test Data: No specific test data needed.

Test Steps:

1. Navigate to the user's personalized learning path or dashboard.
2. Observe if the platform displays a section for "Recommended Courses/Modules."
3. Verify that the recommendations align with the user's learning path, interests, and previous selections.

Postconditions: None

Expected Result: The platform should provide a list of personalized course/module recommendations that align with the learner's interests and existing learning path.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: PLP-004

Test Case Title: Verify that progress tracking is personalized to each learner's path.

Test Case Description: This test case ensures that the platform accurately tracks and displays the individual learner's progress within their selected courses, modules, and overall learning path.

Test Suite: Personalized Learning Path

Test Priority: High

Preconditions:

- User is logged in.
- User has an active learning path with courses/modules in progress.

Test Data: No specific test data needed.

Test Steps:

1. Complete various activities within a course/module (e.g., watch lectures, take quizzes).
2. Navigate to the user's learning path or progress dashboard.
3. Verify that the displayed progress accurately reflects the completed activities and overall progress within each course/module and the entire learning path.

Postconditions: None

Expected Result: The platform should accurately track and visually represent the learner's individual progress, reflecting their completed activities and milestones within their personalized path.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: PLP-005

Test Case Title: Verify that learners can adjust their learning paths at any time.

Test Case Description: This test ensures the platform allows users the flexibility to modify their learning paths by adding or removing courses and modules whenever they choose.

Test Suite: Personalized Learning Path

Test Priority: Medium

Preconditions:

- User is logged in.

- User has an active learning path with at least one course/module.

Test Data: No specific test data needed.

Test Steps:

1. Navigate to the "My Learning Path" section.

2. Attempt to remove an existing course/module from the path.

3. Attempt to add a new course/module to the path.

4. Confirm that the changes are reflected accurately in the learning path overview.

Postconditions: None

Expected Result: Users should have the ability to add or remove courses and modules from their learning path without restrictions, and the changes should immediately reflect in their personalized path view.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive, Destructive

Data - 52

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Progress Tracking

Feature description -

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

Userstory title -

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Acceptance criteria -

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Progress Tracking Feature

Test Case ID: TC_PT_01

Test Case Title: Verify that the progress tracking feature displays completed activities and grades accurately.

Test Case Description: This test case verifies that the platform correctly tracks and displays the learner's progress in terms of completed activities and the grades achieved for those activities.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

* The learner is logged into the platform.

* The learner has enrolled in a course and completed at least one activity with a recorded grade.

Test Data: No test data needed

Test Steps:

1. Navigate to the "My Courses" section.

2. Select the course containing completed activities.

3. View the progress tracking section for the selected course.

4. Verify that completed activities are marked as such.

5. Verify that the displayed grades for completed activities are accurate.

Postconditions: None

Expected Result: The progress tracking section should accurately reflect the completed activities and their corresponding grades.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_PT_02

Test Case Title: Verify that learners can access detailed progress reports for their enrolled courses.

Test Case Description: This test case verifies the functionality of accessing and reviewing detailed reports that provide a comprehensive overview of the learner's progress within a specific course.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

* The learner is logged into the platform.

* The learner is enrolled in a course.

Test Data: No test data needed

Test Steps:

1. Navigate to the "My Courses" section.

2. Select the desired course.

3. Locate and click the option to view the progress report.

4. Verify that the report opens and contains detailed information.

5. Verify the report includes data on completed activities, grades, time spent, and overall course progress.

Postconditions: None

Expected Result: The learner should be able to access a comprehensive progress report for the selected course, providing detailed insights into their performance.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_PT_03

Test Case Title: Verify that achievements are highlighted within the progress tracker.

Test Case Description: This test case ensures that the platform visually highlights any achievements or milestones the learner has reached during their course progression.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

* The learner is logged into the platform.

* The learner has achieved at least one milestone or badge within a course.

Test Data: No test data needed

Test Steps:

1. Navigate to the progress tracking section for a course where an achievement has been earned.

2. Verify that the achieved milestone or badge is visually highlighted.

3. Verify that the highlight is clear and easily noticeable.

Postconditions: None

Expected Result: Achievements should be visually emphasized within the progress tracker to provide positive reinforcement and a clear indication of the learner's accomplishments.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TC_PT_04

Test Case Title: Verify that learners receive timely notifications regarding their progress.

Test Case Description: This test case validates the notification system's functionality, ensuring learners are promptly informed about their progress, including grade updates, course milestones, and other relevant information.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

* The learner is logged into the platform.

* The notification settings are enabled for the learner's account.

Test Data:

* Notification Trigger: Completion of a graded activity

Test Steps:

1. Complete an activity that triggers a grade update within a course.

2. Verify that the learner receives a notification about the new grade.

3. Verify the notification is delivered promptly after the triggering event.

4. Verify the content of the notification accurately reflects the progress update.

Postconditions:

* Dismiss or clear the notification.

Expected Result: Learners should receive timely and accurate notifications regarding their progress, enhancing their engagement and awareness within the platform.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_PT_05

Test Case Title: Verify that progress tracking information can be exported as a report.

Test Case Description: This test case ensures that the platform allows the downloading or exporting of progress data in a common report format for offline viewing or sharing purposes.

Test Suite: Progress Tracking

Test Priority: Low

Preconditions:

* The learner is logged into the platform.

* The learner has progress data available for export.

Test Data: No test data needed

Test Steps:

1. Navigate to the progress tracking section.

2. Locate and select the export functionality (button, link, etc.).

3. Choose the desired report format if applicable (e.g., PDF, CSV).

4. Initiate the export process.

5. Verify that the report downloads successfully in the selected format.

6. Verify that the exported report contains accurate and complete progress information.

Postconditions: None

Expected Result: The learner should be able to export their progress tracking data in a user-friendly format, enabling them to save, print, or share their learning achievements.

Severity: Minor

Priority: Low

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 Cases for "Assessment and Grading" Feature:

Test Case ID: TC_AG_001

Test Case Title: Verify that instructors can create quizzes, tests, and assignments.

Test Case Description: This test case verifies that instructors can successfully create different types of assessments ? quizzes, tests, and assignments ? on the platform.

Test Suite: Assessment Creation

Test Priority: High

Preconditions:

- Instructor account is created and logged in.

- Course is created and accessible by the instructor.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Assessments" section of the course.

2. Select the option to create a new assessment.

3. Choose the assessment type: "Quiz," "Test," or "Assignment."

4. Fill in the required details for the assessment (e.g., title, instructions).

5. Save the assessment.

Postconditions:

- The newly created assessment appears in the list of assessments for the course.

Expected Result: The platform should allow the instructor to create and save each type of assessment successfully.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_AG_002

Test Case Title: Verify that assessments can include various question types.

Test Case Description: This test case ensures that instructors can add different question types, including MCQs, short answers, and essays, to their assessments.

Test Suite: Assessment Creation

Test Priority: High

Preconditions:

- Instructor account is created and logged in.

- An assessment (quiz, test, or assignment) is created.

Test Data: No test data needed.

Test Steps:

1. Open the created assessment in edit mode.

2. Select the option to add a new question.

3. Choose from the available question types: "Multiple Choice," "Short Answer," "Essay."

4. Fill in the necessary details for each question type.

5. Save the assessment.

Postconditions:

- The assessment should contain the added questions of different types.

Expected Result: The platform should support the inclusion of various question types within assessments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_AG_003

Test Case Title: Verify that automatic grading is available for objective questions.

Test Case Description: This test case checks the automatic grading functionality for objective question types like MCQs.

Test Suite: Assessment Grading

Test Priority: High

Preconditions:

- Instructor account is created and logged in.

- An assessment with at least one MCQ question is created.

- A student has submitted the assessment.

Test Data: Student submission with answers to the MCQ question.

Test Steps:

1. Access the submitted assessment.

2. Check if the MCQ question has been automatically graded.

3. Compare the automatic grade with the expected correct answer.

Postconditions: None

Expected Result: The platform should automatically grade the MCQ question accurately.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_AG_004

Test Case Title: Verify that manual grading is available for subjective questions.

Test Case Description: This test case confirms that instructors can manually grade subjective questions like short answers and essays.

Test Suite: Assessment Grading

Test Priority: High

Preconditions:

- Instructor account is created and logged in.

- An assessment with at least one short answer or essay question is created.

- A student has submitted the assessment.

Test Data: Student submission with answers to the subjective question.

Test Steps:

1. Access the submitted assessment.

2. Navigate to the subjective question requiring manual grading.

3. Input the grade and provide feedback for the student's answer.

4. Save the grading.

Postconditions:

- The student's submitted answer should display the assigned grade and feedback.

Expected Result: The platform should allow instructors to manually grade subjective questions and provide feedback.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_AG_005

Test Case Title: Verify that feedback is provided to learners after grading.

Test Case Description: This test case verifies that learners receive feedback on their submitted assessments after the instructor has graded them.

Test Suite: Assessment Feedback

Test Priority: High

Preconditions:

- Instructor account is created and logged in.

- An assessment is created and graded (either automatically or manually).

Test Data: No test data needed.

Test Steps:

1. Log in as a student who submitted the graded assessment.

2. Access the submitted assessment.

3. Check for the presence of feedback provided by the instructor.

Postconditions: None

Expected Result: The platform should display the grade and any feedback provided by the instructor for each question and/or overall assessment.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 54

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Forums

Feature description -

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge sharing, enriching the learning experience through diverse perspectives and active participation.

Userstory title -

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

Acceptance criteria -

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all devices.']

Generated Test Cases:

## Test Cases for Discussion Forums:

Test Case ID: TC_DF_001

Test Case Title: Verify that learners can post questions and responses in forums.

Test Case Description: This test case verifies that registered learners can successfully post new questions and replies to existing threads within a specific course forum.

Test Suite: Discussion Forum Functionality

Test Priority: High

Preconditions:

- User is logged into the platform.

- User is enrolled in a course with an active forum.

Test Data:

- New question: "What are the key takeaways from the last lecture?"

- Reply to an existing thread: "I found this resource helpful: [link]"

Test Steps:

1. Navigate to the desired course page.

2. Access the discussion forum section.

3. Click on the "New Question" button.

4. Input the test question data and submit.

5. Locate an existing thread and click "Reply."

6. Input the test reply data and submit.

Postconditions:

- Verify the new question is displayed correctly in the forum.

- Verify the reply is successfully added to the chosen thread.

Expected Result: The learner should be able to post a new question and reply to an existing thread without encountering any errors.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_DF_002

Test Case Title: Verify that forums support threaded discussions.

Test Case Description: This test case ensures the discussion forum correctly displays replies in a threaded format, maintaining the context and flow of the conversation.

Test Suite: Discussion Forum Functionality

Test Priority: High

Preconditions:

- A forum thread with at least two replies exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to a course forum with existing threads.

2. Select a thread containing multiple replies.

3. Observe the display format of the replies.

Postconditions: None

Expected Result: The replies within the thread should be indented and displayed hierarchically, clearly indicating the conversation flow and the relationship between replies.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_DF_003

Test Case Title: Verify that instructors can moderate discussions.

Test Case Description: This test case confirms that instructors assigned to a course have the necessary permissions to moderate the forum discussions effectively.

Test Suite: Discussion Forum Moderation

Test Priority: High

Preconditions:

- An instructor account is logged in.

- The instructor is assigned to a course with an active forum.

Test Data:

- Inappropriate post content: "This course is a waste of time."

Test Steps:

1. Navigate to the course forum as an instructor.

2. Locate a post with inappropriate content.

3. Attempt to delete the identified post.

4. Attempt to edit the identified post.

Postconditions:

- Verify the deleted post is no longer visible in the forum.

- Verify the edited post reflects the changes made.

Expected Result: The instructor should be able to delete and edit learner posts within their assigned course forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Negative, Destructive

Test Case ID: TC_DF_004

Test Case Title: Verify that learners receive notifications for new posts.

Test Case Description: This test case ensures learners are notified about new posts or replies within threads they are participating in or have subscribed to.

Test Suite: Notifications & Alerts

Test Priority: Medium

Preconditions:

- The learner has enabled notifications for forum activity.
- A new post or reply is made in a thread the learner is participating in.

Test Data:

- New post: "I have a question about the assignment."

Test Steps:

1. As a learner, participate in a forum thread.
2. Log out of the platform and log back in.
3. Check for a notification badge on the forum icon or designated notification area.
4. Click on the notification to view the new post.

Postconditions:

- Verify the notification count accurately reflects unread forum activity.

Expected Result: The learner should receive a timely and clear notification alerting them to the new post within the forum thread.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_DF_005

Test Case Title: Verify that forums are accessible on all devices.

Test Case Description: This test ensures the responsiveness and accessibility of the discussion forums across various devices with different screen sizes and resolutions.

Test Suite: Compatibility & Responsiveness

Test Priority: High

Preconditions:

- Access to different devices (e.g., desktop, laptop, tablet, smartphone).
- Different web browsers on each device.

Test Data: No test data needed.

Test Steps:

1. Access the platform and navigate to a course forum using a desktop computer.
2. Repeat step 1 using a laptop with a different screen size and resolution.
3. Repeat step 1 using a tablet device.
4. Repeat step 1 using a smartphone device.
5. In each case, attempt to post a new question, reply to a thread, and navigate through different forum sections.

Postconditions: None

Expected Result: The forum should be accessible, functional, and display correctly on all tested devices and browsers, ensuring a consistent user experience.

Severity: Major

Priority: High

Type of Testing: Compatibility Testing, Usability Testing, Cross-Browser 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 Cases for Live Classes and Webinars Feature

Test Case ID: TC_LCW_001

Test Case Title: Verify that live classes and webinars are conducted through video conferencing tools.

Test Case Description: This test case verifies that the platform utilizes video conferencing tools for hosting live classes and webinars, ensuring seamless audio-visual communication.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Learner is registered and logged in.

- A live class or webinar is scheduled.

Test Data: No test data needed.

Test Steps:

1. Navigate to the scheduled live class/webinar.

2. Join the session.

3. Observe if the platform launches a video conferencing tool (e.g., Zoom, MS Teams).

4. Check audio and video functionality within the tool.

Postconditions:

- Leave the live session.

Expected Result: The platform should successfully utilize a video conferencing tool for the live session, with functional audio and video.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: TC_LCW_002

Test Case Title: Verify that learners can ask questions in real-time during live sessions.

Test Case Description: This test case validates the real-time question-asking functionality for learners during live sessions.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Learner is registered and logged in.

- Learner is actively participating in a live session.

Test Data: A sample question.

Test Steps:

1. Locate the designated chat or Q&A section within the live session interface.

2. Type the sample question.

3. Submit the question.

4. Observe if the question is visible to the instructor and other participants.

Postconditions:

- Leave the live session.

Expected Result: The learner's question should be successfully submitted and visible in real-time within the session.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_LCW_003

Test Case Title: Verify that live sessions are recorded and available for later viewing.

Test Case Description: This test case ensures that concluded live sessions are recorded and accessible for learners to review at their convenience.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- A live session has been concluded.

Test Data: No test data needed.

Test Steps:

1. Navigate to the completed live session section.

2. Locate the recorded session.

3. Attempt to play the recording.

4. Check if the recording includes both audio and video of the session.

Postconditions: None

Expected Result: The platform should provide access to the recorded session, with both audio and video playback functioning correctly.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_LCW_004

Test Case Title: Verify that instructors can share their screen and resources during live sessions.

Test Case Description: This test case validates the ability of instructors to share their screen or specific resources with learners during live sessions.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- User is logged in as an instructor.

- Instructor is hosting a live session.

Test Data: A sample presentation file.

Test Steps:

1. Locate and utilize the screen sharing functionality within the video conferencing tool.

2. Select to share either the entire screen or a specific application window.

3. If applicable, share the sample presentation file.

4. Observe if the shared content is visible to the participants.

Postconditions:

- End the screen sharing.

- End the live session.

Expected Result: The instructor should successfully share their screen or resources, with the content clearly visible to participants in the live session.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TC_LCW_005

Test Case Title: Verify that learners receive notifications before live sessions start.

Test Case Description: This test case ensures that the platform sends timely notifications to learners before the scheduled start time of live sessions.

Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

- Learner is registered and logged in.

- Learner is enrolled in a course with an upcoming live session.

- Notifications are enabled in the learner's profile settings.

Test Data: No test data needed.

Test Steps:

1. Note the scheduled time of the upcoming live session.

2. Observe if the platform delivers a notification reminding the learner about the session.

3. Check the notification content for accuracy regarding session time and topic.

Postconditions: None

Expected Result: The learner should receive a timely and accurate notification before the live session commences.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

--

Data - 56

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Mobile Accessibility

Feature description -

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

Userstory title -

As a learner, I want mobile access to the platform so that I can learn on the go.

Acceptance criteria -

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Mobile Accessibility

Test Case ID: MA-001

Test Case Title: Verify that the platform is accessible on smartphones.

Test Case Description: This test case verifies the accessibility of the online learning platform on various smartphone devices with different screen sizes and operating systems (Android and iOS).

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- A stable internet connection is available.
- Test devices are fully charged.

Test Data: No test data needed

Test Steps:

1. Access the online learning platform URL on different smartphone models running the latest Android and iOS versions.
2. Observe the platform's loading time and overall responsiveness.
3. Navigate through different sections of the platform, including the course catalog, learning modules, and user profile.
4. Verify that all UI elements are visible and correctly rendered without any distortion or overlapping.
5. Attempt to log in/register an account on the platform.

Postconditions:

- Log out of the platform after each test run.

- Clear browser cache and cookies on the test devices.

Expected Result: The platform should load correctly, and all features should be accessible and functional on all tested smartphone devices without any major usability issues.

Severity: Blocker

Priority: High

Type of Testing: Compatibility Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: MA-002

Test Case Title: Verify that the platform is accessible on tablets.

Test Case Description: This test case verifies the accessibility and responsiveness of the platform on different tablet devices with varying screen sizes and operating systems (Android and iOS).

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- A stable internet connection is available.

- Test devices are fully charged.

Test Data: No test data needed

Test Steps:

1. Access the online learning platform URL on different tablet models running the latest Android and iOS versions.

2. Observe the platform's loading time and responsiveness.

3. Navigate through various platform sections, including the course catalog, learning modules, and user profile.

4. Verify that all UI elements are visible and correctly rendered, adapting to the larger screen size without distortion.

5. Attempt to interact with interactive elements like video players and quizzes.

Postconditions:

- Log out of the platform after each test run.

- Clear browser cache and cookies on the test devices.

Expected Result: The platform should load correctly and adapt to different tablet screen sizes, ensuring all features are accessible and functional without usability issues.

Severity: Blocker

Priority: High

Type of Testing: Compatibility Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: MA-003

Test Case Title: Verify that the mobile UI is optimized for touch navigation.

Test Case Description: This test case checks if the platform's UI is responsive and easy to navigate using touch gestures on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Access the platform on a smartphone or tablet.

Test Data: No test data needed

Test Steps:

1. Navigate through the platform using common touch gestures like tapping, swiping, and pinching.

2. Attempt to select menu items, scroll through content, and interact with buttons and links.

3. Check the responsiveness of touch inputs and if any actions require excessive effort or precision.

4. Verify that interactive elements have a sufficient touch target size to prevent accidental clicks.

Postconditions: None

Expected Result: The platform should respond accurately and promptly to touch gestures, making navigation seamless and user-friendly on mobile devices.

Severity: Critical

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: MA-004

Test Case Title: Verify that course materials are accessible and readable on mobile devices.

Test Case Description: This test case ensures that learners can access and consume course materials, including text, images, videos, and downloadable resources, comfortably on various mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Access a course on the platform using a mobile device.

Test Data: Sample course with various content formats (text, images, video, PDF).

Test Steps:

1. Open and navigate through different types of course materials, including text documents, image galleries, and video lectures.

2. Verify that text is readable without excessive zooming and reflow issues.

3. Check if images are displayed clearly and can be zoomed in/out without distortion.

4. Confirm that video lectures can be played smoothly, and the player controls are accessible.

5. Download a sample resource file (e.g., PDF) and verify that it opens correctly on the device.

Postconditions: None

Expected Result: All course materials should be easily accessible, readable, and viewable on mobile devices without compromising quality or user experience.

Severity: Critical

Priority: High

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: MA-005

Test Case Title: Verify that the platform supports offline access to downloaded materials.

Test Case Description: This test case validates the platform's offline functionality, ensuring that learners can access downloaded course materials without an active internet connection.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Be logged in to the platform on a mobile device.

- Have downloaded course materials saved within the app.

Test Data: Downloaded course material (e.g., PDF, video lecture).

Test Steps:

1. Turn off the internet connection on the mobile device.

2. Open the platform and navigate to the downloaded materials section.

3. Attempt to access and open the previously downloaded content (PDF, video).

4. Verify that the content opens correctly and is accessible without any errors.

5. Try to navigate within the downloaded material, like switching between pages in a PDF or seeking within a

video.

Postconditions:

- Turn on the internet connection on the mobile device.

Expected Result: The downloaded course materials should be accessible offline without any issues, allowing learners to continue their learning even without an active internet connection.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 57

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Resource Library

Feature description -

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

Userstory title -

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

Acceptance criteria -

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

Generated Test Cases:

## Test Cases for Resource Library Feature

Test Case ID: RL-001

Test Case Title: Verify that the resource library contains e-books, articles, and reference guides.

Test Case Description: This test verifies that the resource library contains the specified types of materials, ensuring users have access to diverse content.

Test Suite: Resource Library Functionality

Test Priority: High

Preconditions:

* User is logged into the platform.

* User has navigated to the resource library.

Test Data: No test data needed

Test Steps:

1. Access the resource library.

2. Observe the available resource types.

3. Verify that e-books are present.

4. Verify that articles are present.

5. Verify that reference guides are present.

Postconditions: None

Expected Result: The resource library should display e-books, articles, and reference guides as available resource types.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: RL-002

Test Case Title: Verify that learners can search for specific materials in the resource library.

Test Case Description: This test validates the search functionality within the resource library, ensuring users can efficiently locate desired materials.

Test Suite: Resource Library Functionality

Test Priority: High

Preconditions:

* User is logged in.

* User is in the resource library.

Test Data:

* Search Term: "Data Structures"

Test Steps:

1. Input the search term "Data Structures" into the search bar.

2. Initiate the search.

3. Observe the displayed search results.

Postconditions: None

Expected Result: The search results should display resources related to "Data Structures," including e-books, articles, or reference guides.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: RL-003

Test Case Title: Verify that resources in the library are categorized and tagged for easy navigation.

Test Case Description: This test ensures resources are properly categorized and tagged, allowing users to browse and filter materials efficiently.

Test Suite: Resource Library Usability

Test Priority: High

Preconditions:

* User is logged in.

* User is in the resource library.

Test Data: No test data needed

Test Steps:

1. Observe the resource library interface.

2. Verify the presence of resource categories (e.g., subject, topic).

3. Select a category and verify that resources within the category are relevant.

4. Verify the presence of tags on resources.

5. Select a tag and verify that displayed resources are relevant to the tag.

Postconditions: None

Expected Result: The resource library should display clear categories and relevant tags, facilitating efficient browsing and filtering.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: RL-004

Test Case Title: Verify that learners can download resources for offline access.

Test Case Description: This test confirms the download functionality for offline access, enabling users to access materials without an internet connection.

Test Suite: Resource Library Functionality

Test Priority: High

Preconditions:

* User is logged in.

* User is in the resource library.

* A downloadable resource (e.g., ebook) is available.

Test Data: No test data needed

Test Steps:

1. Select a downloadable resource.

2. Locate and click the "Download" option.

3. Choose a download location on the local device.

4. Verify that the download starts and completes successfully.

5. Locate the downloaded file and verify its integrity.

Postconditions: Delete the downloaded file after verification.

Expected Result: The resource should download successfully, enabling offline access. The downloaded file should be accessible and readable.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RL-005

Test Case Title: Verify that instructors can recommend resources to learners.

Test Case Description: This test confirms instructors can recommend specific resources, enhancing the learning experience through curated suggestions.

Test Suite: Resource Library - Instructor Features

Test Priority: Medium

Preconditions:

* Instructor account is logged in.

* Instructor has access to the resource library.

* A learner is enrolled in the instructor's course.

Test Data:

* Resource to recommend: "Introduction to Python Programming" ebook

Test Steps:

1. Navigate to the resource library as an instructor.

2. Locate the "Introduction to Python Programming" ebook.

3. Use the "Recommend Resource" function.

4. Select the learner(s) to receive the recommendation.

5. Confirm the recommendation.

Postconditions:

* Verify that the learner receives a notification about the recommended resource.

Expected Result: The instructor should be able to successfully recommend the specified resource to the selected learner(s).

Severity: Minor

Priority: Medium

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Data - 58

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Peer Review

Feature description -

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

Userstory title -

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

Acceptance criteria -

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers remain anonymous to ensure unbiased feedback.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Peer Review Feature

Test Case ID: PR-001

Test Case Title: Verify that learners can submit assignments for peer review.

Test Case Description: This test case verifies the functionality of learners submitting assignments for peer review. It ensures that the platform allows assignment submission and correctly associates it with the relevant course and task.

Test Suite: Peer Review Functionality

Test Priority: High

Preconditions:

- The learner is logged in.

- The learner is enrolled in a course with a peer review assignment enabled.

Test Data: A sample assignment file (e.g., .doc, .pdf).

Test Steps:

1. Navigate to the course with the peer review assignment.

2. Access the assignment submission page.

3. Upload the assignment file.

4. Submit the assignment for peer review.

Postconditions:

- The assignment is successfully submitted.

- The learner sees a confirmation message.

- The assignment appears in the pool for peer review.

Expected Result: The learner should be able to submit their assignment for peer review without encountering any errors.

Severity: Critical

Priority: High

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case ID: PR-002

Test Case Title: Verify that peers can provide feedback on submitted assignments.

Test Case Description: This test case checks if the platform allows peers to review submitted assignments and provide feedback through a designated interface.

Test Suite: Peer Review Functionality

Test Priority: High

Preconditions:

- The reviewer is logged in.
- An assignment is available in the pool for review.

Test Data: No test data needed.

Test Steps:

1. Navigate to the peer review section.
2. Select an assignment from the pool.
3. Access the review interface.
4. Provide feedback in the designated fields/sections.
5. Submit the review.

Postconditions:

- Feedback is successfully submitted.
- The reviewer is notified of successful submission.

Expected Result: Reviewers should be able to access, review, and provide feedback on submitted assignments using the provided functionalities.

Severity: Critical

Priority: High

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case ID: PR-003

Test Case Title: Verify that a structured review process guides the feedback.

Test Case Description: This test case ensures that the platform provides a clear and structured framework for providing feedback, guiding reviewers with prompts or criteria.

Test Suite: Peer Review Usability

Test Priority: High

Preconditions:

- The reviewer is logged in and on the assignment review page.

Test Data: No test data needed.

Test Steps:

1. Observe the feedback input section.

2. Identify the presence of guiding prompts or criteria.

3. Attempt to submit feedback without addressing all prompts/criteria.

Postconditions: None

Expected Result:

- The review interface should have clear prompts, questions, or criteria related to the assignment.

- The system should prevent submission if all prompts/criteria are not addressed.

Severity: Major

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive, Negative

Test Case ID: PR-004

Test Case Title: Verify that learners can view feedback on their work.

Test Case Description: This test case ensures that learners can access and view the feedback provided by their peers on their submitted assignments.

Test Suite: Peer Review Feedback

Test Priority: High

Preconditions:

- The learner is logged in.
- At least one peer review has been submitted for their assignment.

Test Data: No test data needed.

Test Steps:

1. Navigate to the submitted assignment.
2. Access the feedback section.
3. Verify that the feedback from peers is displayed.

Postconditions: None

Expected Result: The learner should be able to view the feedback provided by their peers in a clear and organized manner.

Severity: Critical

Priority: High

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case ID: PR-005

Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback.

Test Case Description: This test case checks if the anonymity of reviewers is maintained throughout the feedback process to ensure unbiased reviews.

Test Suite: Peer Review Anonymity

Test Priority: High

Preconditions:

- A learner is logged in and viewing feedback on their assignment.

Test Data: No test data needed.

Test Steps:

1. Access the feedback provided by different reviewers.
2. Attempt to identify any information that could reveal the reviewer's identity.

Postconditions: None

Expected Result: The platform should not display any identifiable information about the reviewer, ensuring their anonymity.

Severity: Blocker

Priority: High

Type of Testing: Security Testing, Functional 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 Cases for Online Learning Platform - Certificates and Badges Feature

Test Case ID: TC_CB_001

Test Case Title: Verify that learners earn certificates upon course completion.

Test Case Description: This test case verifies that the system correctly issues a certificate to a learner upon successfully completing all course requirements, including passing quizzes and final exams.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

* A learner account is created and logged in.

* The learner is enrolled in a course with all modules, quizzes, and exams accessible.

Test Data: No test data needed

Test Steps:

1. Navigate to the enrolled course.

2. Complete all course modules, including watching lectures, accessing resources, and participating in discussions.

3. Successfully pass all quizzes and the final exam with a passing grade.

4. Check the course completion status and verify it reflects as "Completed."

5. Navigate to the learner's profile section dedicated to certificates.

Postconditions:

* Verify a certificate for the completed course is issued and displayed.

* The certificate should include relevant details like the course name, learner's name, date of completion, and platform logo.

Expected Result: A certificate should be generated and awarded to the learner upon successful course completion, viewable in their profile.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, End-to-End Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_CB_002

Test Case Title: Verify that badges are awarded for specific achievements and milestones.

Test Case Description: This test case checks if the system accurately awards badges to learners upon achieving predefined milestones within a course, such as completing a challenging module, achieving a high score on a quiz, or actively participating in the discussion forum.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

* A learner account is created and logged in.

* The learner is enrolled in a course with badges enabled for specific achievements.

Test Data: No test data needed

Test Steps:

1. Access the course with predefined badge criteria.

2. Complete the specific tasks or achieve the milestones required for badge eligibility.

3. For example, achieve a score above 90% on a quiz or contribute meaningfully to a discussion forum thread.

4. Navigate to the learner's profile section dedicated to badges.

Postconditions:

* Verify that the corresponding badge is awarded and displayed in the learner's profile.

* The badge should include a visual representation and a brief description of the achieved milestone.

Expected Result: Relevant badges should be automatically awarded and displayed on the learner's profile upon meeting the predefined criteria.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: TC_CB_003

Test Case Title: Verify that certificates and badges are accessible through learner profiles.

Test Case Description: This test verifies that earned certificates and badges are displayed and readily accessible to learners through their profiles on the online learning platform.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

* A learner account is created and logged in.

* The learner has earned at least one certificate and one badge.

Test Data: No test data needed

Test Steps:

1. Log in to the learner account.

2. Navigate to the learner's profile page.

3. Locate the sections dedicated to displaying earned certificates and badges.

Postconditions: None

Expected Result: The profile should clearly display all earned certificates and badges, with each item easily identifiable and accessible.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CB_004

Test Case Title: Verify that learners can share their certificates and badges on social media.

Test Case Description: This test case ensures that learners can seamlessly share their achieved certificates and badges on various social media platforms directly from the online learning platform.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

* A learner account is created and logged in.

* The learner has earned at least one certificate and one badge.

Test Data: No test data needed

Test Steps:

1. Log in to the learner account.

2. Navigate to the certificates and badges sections within the learner's profile.

3. Locate and click on the "Share" or social media icons associated with each certificate and badge.

4. Verify that sharing options are available for popular social media platforms (e.g., Facebook, LinkedIn, Twitter).

5. Attempt to share a certificate and a badge on a selected social media platform.

Postconditions: None

Expected Result: Sharing options should function correctly, allowing learners to share their achievements on their chosen social media platforms with pre-populated content reflecting the accomplishment.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Integration Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CB_005

Test Case Title: Verify that instructors can issue custom badges for special achievements.

Test Case Description: This test case validates the functionality for instructors to create and award custom badges to learners for special achievements or contributions not covered by the predefined badge criteria.

Test Suite: Certificates and Badges

Test Priority: Low

Preconditions:

* An instructor account is created and logged in.

* The instructor is assigned to a course.

Test Data:

* Badge Name: "Outstanding Contribution"

* Badge Description: "Awarded for exceptional participation and insightful contributions to the course

discussions."

Test Steps:

1. Log in to the instructor account.
2. Navigate to the course settings or a dedicated section for badge management.
3. Select the option to create a new custom badge.
4. Input the badge name, a brief description, and optionally upload a unique visual for the badge.
5. Save the new custom badge.
6. Locate the option to award badges to learners within the course.
7. Select the learner and the newly created custom badge to issue it.

Postconditions:

* Verify that the custom badge appears in the designated learner's profile.

Expected Result: The system should allow instructors to create and issue custom badges to learners, and these badges should be reflected accurately in the recipient's profile.

Severity: Trivial

Priority: Low

Type of Testing: Functional Testing, Security Testing (to ensure only authorized instructors can create badges), Acceptance Testing

Test Case Approach: Positive

Data - 60

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Gamification Elements

Feature description -

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

Userstory title -

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

Acceptance criteria -

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

Generated Test Cases:

## Test Cases for Gamification Elements in Online Learning Platform

Test Case ID: TC_GF_001

Test Case Title: Verify that learners earn points for completing activities.

Test Case Description: This test case verifies that the system correctly awards points to learners upon successful completion of various learning activities like watching a video lecture, passing a quiz, or participating in a forum discussion.

Test Suite: Gamification

Test Priority: High

Preconditions:

* A learner account is created and logged in.

* A course with gamified activities is enrolled in.

Test Data: No test data needed.

Test Steps:

1. Access a course with gamified activities.

2. Complete a learning activity (e.g., watch a full video lecture).

3. Check the points earned for the completed activity.

4. Repeat steps 2-3 for different types of activities (quiz, forum participation).

Postconditions:

* Verify points are added to the learner's total score.

* Ensure the point system aligns with the platform's predefined rules.

Expected Result: The learner should receive the correct number of points for each completed activity as defined by the platform's gamification rules.

Severity: Major

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_GF_002

Test Case Title: Verify that levels are achieved based on accumulated points.

Test Case Description: This test case checks if the platform accurately transitions learners to higher levels based on the total points they accumulate from completing various activities.

Test Suite: Gamification

Test Priority: High

Preconditions:

* A learner account is created and logged in.

* The learner has accumulated enough points to reach the next level.

Test Data: No test data needed.

Test Steps:

1. Reach the required points threshold for the next level.

2. Check if the level automatically updates to the next level.

3. Verify the level displayed on the learner?s profile and dashboard.

Postconditions:

* Ensure level-up notifications are displayed to the learner.

* Confirm any associated rewards or benefits are granted.

Expected Result: Upon reaching the required points, the learner?s level should automatically upgrade, reflecting their progress.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_GF_003

Test Case Title: Verify that leaderboards display top-performing learners.

Test Case Description: This test case examines the leaderboard functionality, ensuring it correctly ranks and displays learners based on their accumulated points or other defined metrics.

Test Suite: Gamification

Test Priority: Medium

Preconditions:

* Multiple learner accounts are active on the platform.

* Learners have earned varying points from completed activities.

Test Data: No test data needed.

Test Steps:

1. Access the leaderboard section of the platform.

2. Verify that learners are ranked in descending order of their points.

3. Check for accurate display of learner names/usernames and points.

4. Ensure the leaderboard updates dynamically as learners earn more points.

Postconditions: None

Expected Result: The leaderboard should accurately reflect the ranking of learners based on their performance, fostering healthy competition.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case ID: TC_GF_004

Test Case Title: Verify that badges are awarded for reaching milestones.

Test Case Description: This test case aims to validate if the platform successfully awards badges to learners upon achieving specific milestones, such as completing a certain number of courses or reaching a high point threshold.

Test Suite: Gamification

Test Priority: Medium

Preconditions:

* A learner account is created and logged in.

* Specific milestones for badge achievements are defined within the platform.

Test Data: No test data needed.

Test Steps:

1. Achieve a predefined milestone (e.g., completing five courses).

2. Check if the corresponding badge is awarded to the learner.

3. Verify the badge is displayed on the learner's profile.

4. Test different badge achievement scenarios.

Postconditions:

* Ensure badge notifications are displayed to the learner.

Expected Result: Upon achieving a predefined milestone, the platform should automatically award the corresponding badge to the learner, motivating further engagement.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_GF_005

Test Case Title: Verify that learners can view their progress in a gamification dashboard.

Test Case Description: This test case ensures that learners can access a dedicated dashboard that provides a comprehensive overview of their gamification progress, including their current level, points earned, badges collected, and leaderboard ranking.

Test Suite: Gamification

Test Priority: High

Preconditions:

* A learner account is created and logged in.

* The learner has engaged in gamified activities.

Test Data: No test data needed.

Test Steps:

1. Access the gamification dashboard from the learner's profile.

2. Verify that the dashboard displays the current level, total points, and badges earned.

3. Check if the leaderboard ranking is visible and accurate.

4. Ensure all information on the dashboard is up-to-date and relevant to the learner's progress.

Postconditions: None

Expected Result: The gamification dashboard should provide a clear and concise overview of the learner?s progress, enhancing their understanding of their achievements and motivating further participation.

Severity: Minor

Priority: High

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Data - 61

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Multilingual Support

Feature description -

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

Userstory title -

As a learner, I want multilingual support so that I can access the platform in my preferred language.

Acceptance criteria -

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated help and support resources.', 'Instructors can provide content in multiple languages.']

Generated Test Cases:

## Test Cases for Multilingual Support on Online Learning Platform

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
LS-001	Verify that the platform interface is available in multiple languages	This test case verifies that the user interface of the platform, including menus, buttons, and navigation, is correctly translated and displayed in the selected languages.	Multilingual Support	High	- User account created<br>- User logged in	No test data needed	1. Navigate to the language selection option.<br>2. Choose a language from the available options (e.g., Spanish, French, Mandarin).<br>3. Observe the platform interface.	- Revert language to default.	The platform interface, including all elements, should be displayed in the selected language.	Blocker	High	Functional, UI Testing	Positive
LS-002	Verify that course materials are available in different languages	This test case verifies that learners can access course content, including lectures, readings, and assignments, in their chosen language.	Multilingual Support	High	- User account created<br>- User logged in<br>- Course enrolled	Course ID with multilingual support	1. Navigate to a course offering multilingual content.<br>2. Select the preferred language for the course materials.<br>3. Access different types of course content (lectures, readings, assignments).	None	The selected course content should be displayed in the chosen language.	Critical	High	Functional, Content Testing	Positive
LS-003	Verify that learners can select their preferred language at any time	This test case verifies the functionality of changing the preferred language during an active session.	Multilingual Support	Medium	- User account created<br>- User logged in	No test data needed	1. Log in to the platform and navigate to the profile settings.<br>2. Select a new preferred language from the available options.<br>3. Save the changes

and observe the platform interface and content. | None | The platform should immediately reflect the new language preference across all sections. | Major | Medium | Functional, Usability Testing | Positive |
| LS-004 | Verify that multilingual support includes translated help and support resources | This test case verifies that help articles, FAQs, and other support materials are accessible in the user's preferred language. | Multilingual Support | Medium | - User account created
- User logged in | No test data needed | 1. Navigate to the help and support section of the platform.
2. Verify that the content is available in the user's preferred language.
3. Attempt to access different help resources. | None | Help and support materials should be accurately translated and readily accessible in the user's preferred language. | Minor | Medium | Functional, Content Testing | Positive |
| LS-005 | Verify that instructors can provide content in multiple languages | This test case checks if instructors have the tools and functionality to upload and manage course materials in various languages. | Multilingual Support | Medium | - Instructor account created
- Instructor logged in | Course material in different languages | 1. Login as an instructor.
2. Navigate to course creation/editing section.
3. Attempt to upload and associate different language versions of the same course material.
4. Verify if the platform correctly saves and displays these different language options. | None | Instructors should be able to seamlessly upload, manage, and associate various language versions of their course materials, ensuring learners can access them in their preferred language. | Major | Medium | Functional, Content Management Testing | Positive |

Data - 62

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Accessibility Features

Feature description -

The platform prioritizes accessibility by incorporating features such as screen reader compatibility, adjustable font sizes, and color contrast options. These accessibility features ensure that learners with disabilities can access and navigate the platform with ease, promoting inclusivity and equitable access to educational opportunities.

Userstory title -

As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

Acceptance criteria -

['The platform is compatible with screen readers.', 'Learners can adjust font sizes for better readability.', 'Color contrast options are available for improved visibility.', 'Accessibility features are easy to activate and customize.', 'Support resources are available to assist learners with disabilities.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Accessibility Features

Test Case ID: TC_AF_001

Test Case Title: Verify that the platform is compatible with screen readers.

Test Case Description: This test case verifies the compatibility of the platform with popular screen reader software, ensuring that users with visual impairments can access and navigate the platform's content and functionality effectively.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- A screen reader software (e.g., NVDA, JAWS) is installed and running.

- The Online Learning Platform is launched in a compatible browser.

Test Data: No test data needed

Test Steps:

1. Navigate through different sections of the platform (homepage, courses, lectures, quizzes) using only keyboard shortcuts.
2. Listen to the screen reader output for each element, ensuring clear and accurate descriptions of page content, links, buttons, and interactive elements.
3. Attempt to complete various tasks (e.g., enrolling in a course, accessing learning materials, submitting a quiz) relying solely on screen reader guidance.

Postconditions: None

Expected Result: The screen reader should accurately announce all page elements and functionality, allowing users to navigate and interact with the platform without visual cues.

Severity: Blocker

Priority: High

Type of Testing: Accessibility Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_AF_002

Test Case Title: Verify that learners can adjust font sizes for better readability.

Test Case Description: This test case ensures that users can easily adjust the font size of the platform's content to enhance readability and accommodate individual preferences or visual needs.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- The Online Learning Platform is launched in a compatible browser.

Test Data: No test data needed

Test Steps:

1. Locate and access the font size adjustment settings within the platform's settings or preferences.

2. Increase the font size to the maximum available setting.

3. Verify that the font size of all text elements (headings, body text, captions) on various pages reflects the chosen setting.

4. Decrease the font size to the minimum available setting and repeat step 3.

Postconditions: None

Expected Result: The font size of the platform's content should be adjustable, with changes reflecting accurately across all elements and pages, enhancing readability for users with varying visual acuity.

Severity: Critical

Priority: High

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_AF_003

Test Case Title: Verify that color contrast options are available for improved visibility.

Test Case Description: This test case evaluates the availability and effectiveness of color contrast options, ensuring that users, particularly those with visual impairments, can perceive and distinguish text and UI elements easily.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- The Online Learning Platform is launched in a compatible browser.

Test Data: No test data needed

Test Steps:

1. Locate and access the color contrast settings or themes within the platform's settings.

2. Select different color contrast options (e.g., high contrast, light theme, dark theme).

3. Observe the changes in the platform's color scheme, ensuring sufficient contrast between text and background colors for improved readability.

4. Verify that the chosen color contrast settings persist across different pages and sessions.

Postconditions: None

Expected Result: The platform should provide various color contrast options that meet accessibility standards, enhancing visual clarity and readability for all users.

Severity: Major

Priority: High

Type of Testing: Accessibility Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_AF_004

Test Case Title: Verify that accessibility features are easy to activate and customize.

Test Case Description: This test case ensures that users can easily locate, activate, and personalize accessibility features according to their individual needs and preferences.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- The Online Learning Platform is launched in a compatible browser.

Test Data: No test data needed

Test Steps:

1. Explore the platform's settings and menus to locate accessibility feature options.

2. Attempt to activate and deactivate different accessibility features (e.g., screen reader mode, font size adjustments, color contrast options).

3. Verify that the activation and customization process is intuitive and clearly explained.

4. Check for any personalized settings retention across different sessions or devices.

Postconditions: None

Expected Result: Users should be able to find and customize accessibility features effortlessly, with clear instructions and options to personalize their experience.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_AF_005

Test Case Title: Verify that support resources are available to assist learners with disabilities.

Test Case Description: This test case confirms the presence and accessibility of comprehensive support resources specifically designed to assist learners with disabilities in using the platform effectively.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- The Online Learning Platform is launched in a compatible browser.

Test Data: No test data needed

Test Steps:

1. Locate the platform's "Help," "Support," or "Accessibility" section.

2. Verify that this section includes specific resources (e.g., FAQs, tutorials, contact information) dedicated to accessibility features and support for learners with disabilities.

3. Access and review the content of these resources, ensuring accuracy, clarity, and relevance to users with disabilities.

4. Test any provided contact methods (e.g., email, phone, live chat) to confirm their functionality and responsiveness.

Postconditions: None

Expected Result: The platform should provide readily accessible and comprehensive support resources that address the specific needs of learners with disabilities, ensuring they can get assistance when needed.

Severity: Minor

Priority: Medium

Type of Testing: Documentation 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 Cases for Discussion Moderation Feature

Test Case ID: DM-001

Test Case Title: Verify that instructors can edit posts in discussions.

Test Case Description: This test case verifies the instructor's ability to edit student posts within the discussion forum. This includes correcting misinformation, clarifying points, or improving post formatting for better readability.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- The instructor is logged into the platform.
- A discussion forum with student posts exists.

Test Data: No test data needed

Test Steps:

1. Navigate to the desired course and discussion forum.
2. Locate a student post within the discussion.
3. Click on the "Edit" option associated with the post.

4. Make the desired edits to the post content.

5. Click the "Save" or "Submit" button to apply changes.

Postconditions:

- The edited post is displayed correctly in the discussion forum.

- The edit history is logged (if applicable).

Expected Result: The instructor should successfully edit the selected post, and the changes should reflect in the forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: DM-002

Test Case Title: Verify that instructors can delete inappropriate posts in discussions.

Test Case Description: This test confirms the instructor's ability to remove posts that violate community guidelines, contain offensive content, or are irrelevant to the discussion topic.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- The instructor is logged into the platform.

- A discussion forum with at least one inappropriate student post exists.

Test Data: An example of an inappropriate post (e.g., containing offensive language or spam).

Test Steps:

1. Navigate to the relevant course and discussion forum.

2. Locate the inappropriate student post.

3. Click on the "Delete" or "Remove" option associated with the post.

4. Confirm the deletion action in the system prompt (if applicable).

Postconditions:

- The inappropriate post is removed from the discussion forum.

- The deletion action is logged and reported.

Expected Result: The inappropriate post should be permanently removed from the discussion forum.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Negative

Test Case ID: DM-003

Test Case Title: Verify that instructors can mute or ban disruptive users from discussions.

Test Case Description: This test ensures that instructors have the ability to temporarily mute or permanently ban users who exhibit disruptive behavior in discussion forums, such as spamming, harassment, or personal attacks.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- The instructor is logged into the platform.

- A discussion forum with an identified disruptive user exists.

Test Data: A test user account identified as "disruptive."

Test Steps:

1. Navigate to the course discussion forum with the disruptive user.

2. Locate a post or comment made by the disruptive user.

3. Access the user moderation options (e.g., from a dropdown menu).

4. Select the "Mute" option to temporarily silence the user.

5. Verify that the muted user can no longer post in the forum.

6. Select the "Ban" option to permanently remove user access.

7. Verify that the banned user is no longer visible in the forum.

Postconditions:

- The muted user is temporarily prevented from posting.

- The banned user is permanently removed from the forum.

- The mute/ban actions are logged and reported.

Expected Result: The instructor should be able to mute or ban disruptive users, effectively preventing further disruption within the forum.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Negative, Destructive

Test Case ID: DM-004

Test Case Title: Verify that community guidelines are visible to all participants in discussions.

Test Case Description: This test ensures that community guidelines, outlining acceptable behavior and participation rules, are easily accessible to all users within the discussion forum.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- A user is logged into the platform.

- A discussion forum exists.

Test Data: No test data needed

Test Steps:

1. Navigate to any discussion forum.

2. Look for a dedicated section or link displaying "Community Guidelines."

3. Click on the link or section to access the guidelines.

Postconditions:

- A clear and comprehensive document outlining community guidelines is displayed.

Expected Result: The community guidelines should be easily accessible from the discussion forum, ensuring visibility and awareness for all participants.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: DM-005

Test Case Title: Verify that notifications are sent to instructors and users for moderated actions.

Test Case Description: This test ensures that appropriate notifications are generated and delivered to both instructors and users when moderation actions, such as post edits, deletions, or user mutes/bans, are taken within a discussion forum.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- The instructor is logged in and performs a moderation action (edit, delete, mute, or ban).

Test Data: No specific test data needed.

Test Steps:

1. As an instructor, perform a moderation action within a discussion forum.

2. Check the instructor's notification center for a corresponding notification.

3. Log in as the affected user (if applicable).

4. Check the user's notification center for a corresponding notification.

Postconditions:

- Notifications should be clearly worded and provide relevant context about the moderated action.

Expected Result: Both the instructor and the affected user (if applicable) should receive timely and informative notifications regarding the moderation action taken within the discussion forum.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Data - 64

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Analytics

Feature description -

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement, performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning experience to meet learners' needs effectively.

Userstory title -

As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

Acceptance criteria -

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.', 'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.', 'Instructors can export analytics reports for further analysis.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Learning Analytics Feature

Test Case ID: LA-001

Test Case Title: Verify that learning analytics dashboards display engagement metrics.

Test Case Description: This test case verifies that instructors can access and view various engagement metrics on the learning analytics dashboards, such as course views, resource downloads, forum activity, and time spent on different learning materials.

Test Suite: Learning Analytics Dashboard

Test Priority: High

Preconditions:

- The instructor is logged into the platform.
- The instructor has at least one active course with enrolled learners.

Test Data: No test data needed.

Test Steps:

1. Navigate to the learning analytics dashboard.
2. Verify that the dashboard displays a section dedicated to engagement metrics.
3. Verify that the following engagement metrics are present:
    - Course views
    - Resource downloads
    - Forum posts and replies
    - Time spent on lectures, quizzes, etc.
4. Verify that each metric displays accurate data for the selected course and time period.

Postconditions: None

Expected Result: The learning analytics dashboard successfully displays all relevant engagement metrics with accurate data.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: LA-002

Test Case Title: Verify that performance trends are visualized over time.

Test Case Description: This test case ensures that the learning analytics dashboard provides visual representations of learner performance trends over different timeframes.

Test Suite: Learning Analytics Dashboard

Test Priority: High

Preconditions:

- The instructor is logged into the platform.
- The instructor has at least one active course with learner performance data available (e.g., quiz scores, assignment submissions).

Test Data: No test data needed.

Test Steps:

1. Navigate to the learning analytics dashboard.
2. Locate the performance trends visualization section.
3. Verify that the dashboard can display trends for different timeframes (e.g., weekly, monthly, entire course duration).
4. Select different timeframes and verify that the visualizations update accordingly.
5. Verify that the visualizations accurately represent learner performance data (e.g., average quiz scores, assignment completion rates).

Postconditions: None

Expected Result: The dashboard effectively visualizes performance trends for selected timeframes, providing instructors with insights into learner progress.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: LA-003

Test Case Title: Verify that instructors can access detailed reports on learner progress.

Test Case Description: This test case checks if instructors can drill down from the dashboard overview to access individual learner progress reports containing comprehensive performance details.

Test Suite: Learning Analytics Dashboard, Learner Reports

Test Priority: High

Preconditions:

- The instructor is logged into the platform.
- The instructor has at least one active course with detailed learner performance data available.

Test Data: No test data needed.

Test Steps:

1. Navigate to the learning analytics dashboard.
2. Select a specific course if necessary.
3. Locate and click on a learner's name or an option to view individual reports.
4. Verify that the system navigates to a dedicated learner progress report.
5. Verify that the report displays detailed information on:
    - Quiz scores and attempts

    - Assignment submissions and grades
    - Forum participation
    - Resource access and downloads
    - Time spent on various learning activities

Postconditions: None

Expected Result: The platform allows instructors to access detailed and informative individual learner progress reports.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Navigation Testing

Test Case Approach: Positive

Test Case ID: LA-004

Test Case Title: Verify that analytics data is updated in real-time.

Test Case Description: This test case confirms that the learning analytics dashboard reflects learner interactions and performance updates in real-time, ensuring instructors have access to the most current data.

Test Suite: Learning Analytics Dashboard, Real-Time Updates

Test Priority: High

Preconditions:

- The instructor is logged into the platform.
- An active learning session is ongoing with learner interactions occurring (e.g., students taking a quiz, submitting assignments).

Test Data: N/A

Test Steps:

1. Navigate to the learning analytics dashboard for a course with ongoing learner activity.

2. Monitor the displayed engagement and performance metrics.

3. Have learners engage in various activities (e.g., answer quiz questions, submit assignments).

4. Observe the dashboard for real-time updates reflecting these actions.

5. Verify that the updates are accurately reflected without any significant delay.

Postconditions: None

Expected Result: The learning analytics dashboard accurately displays real-time updates of learner activity and performance data without noticeable delays.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Real-time Data Testing

Test Case Approach: Positive

Test Case ID: LA-005

Test Case Title: Verify that instructors can export analytics reports for further analysis.

Test Case Description: This test case confirms if instructors have the functionality to export learning analytics data in various formats for further analysis or reporting purposes.

Test Suite: Learning Analytics Dashboard, Data Export

Test Priority: Medium

Preconditions:

- The instructor is logged into the platform.

- The instructor has access to the learning analytics dashboard with data available.

Test Data: N/A

Test Steps:

1. Navigate to the learning analytics dashboard.

2. Locate and select the export functionality.

3. Verify that the system offers options to export data in different formats (e.g., CSV, Excel, PDF).

4. Choose a format and initiate the export process.

5. Verify that the downloaded file contains accurate and complete data from the selected analytics sections.

Postconditions: None

Expected Result: The platform allows instructors to export chosen learning analytics data in various formats, maintaining data integrity and structure.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Data Export/Import Testing

Test Case Approach: Positive

Data - 65

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Learning Features

Feature description -

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

Userstory title -

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Acceptance criteria -

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

Generated Test Cases:

## Test Cases for Social Learning Features

Test Case ID: SLF-001

Test Case Title: Verify that learners can create and join groups for projects.

Test Case Description: This test case verifies the functionality for learners to create new project groups and join existing ones. It ensures that the process of group creation and joining is intuitive and functions as expected.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

* The learner is logged into the online learning platform.

* The learner has access to courses with group project features enabled.

Test Data: No test data needed.

Test Steps:

1. Navigate to a course with group project functionality.

2. Locate and select the option to create a new group.

3. Provide the required details for the group (e.g., group name, description).

4. Submit the group creation request.

5. Verify that the group is successfully created and visible.

6. Locate and select the option to join an existing group.

7. Choose a group from the list of available groups.

8. Submit the join group request.

9. Verify that the learner is successfully added to the selected group.

Postconditions:

* The newly created group appears in the list of available groups.

* The learner is listed as a member of the group they joined.

Expected Result: Learners should be able to create and join project groups without encountering any errors.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: SLF-002

Test Case Title: Verify that collaboration tools are available for group communication.

Test Case Description: This test case checks the availability and functionality of various collaboration tools within project groups, ensuring that learners can communicate effectively.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

* The learner is logged into the online learning platform.

* The learner is a member of a project group.

Test Data: No test data needed.

Test Steps:

1. Navigate to a project group.

2. Verify that a group chat or messaging feature is available.

3. Send a message within the group chat.

4. Verify that other group members can receive and read the message.

5. Check for the availability of a file-sharing feature within the group.

6. Upload a file using the file-sharing feature.

7. Verify that other group members can access and download the uploaded file.

Postconditions: None.

Expected Result: All integrated collaboration tools should function correctly, enabling seamless communication and file sharing within project groups.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: SLF-003

Test Case Title: Verify that learners can share resources and feedback within groups.

Test Case Description: This test case examines the resource and feedback sharing mechanism within project groups, ensuring a smooth and effective process.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

* The learner is logged into the online learning platform.

* The learner is a member of a project group.

* Test resources (e.g., documents, links) are prepared.

Test Data: Sample resources for sharing (documents, links).

Test Steps:

1. Navigate to a project group.

2. Locate the resource sharing section within the group.

3. Upload a test resource (document, link).

4. Provide a description or feedback related to the uploaded resource.

5. Share the resource and feedback with the group.

6. Verify that other group members can view, access, and download the shared resource.

7. Check that the accompanying feedback is visible to the group members.

Postconditions: None.

Expected Result: Learners should be able to share resources and provide constructive feedback within their respective project groups without any issues.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SLF-004

Test Case Title: Verify that instructors can assign group projects and monitor progress.

Test Case Description: This test case confirms that instructors have the necessary tools and permissions to create, assign, and monitor the progress of group projects.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

* An instructor account is logged into the online learning platform.

* The course is set up to allow group projects.

Test Data:

* Group project details (title, description, deadlines).

Test Steps:

1. Navigate to the course management section.

2. Create a new group project and define its details.

3. Assign the group project to specific groups or allow learners to form their own groups.

4. Set deadlines for project submissions.

5. Verify that the group project is visible to the assigned learners.

6. Check the functionality for instructors to monitor group progress (e.g., submission status, group activity).

Postconditions: None.

Expected Result: Instructors should have the capabilities to seamlessly create, assign, and monitor the progress of group projects, ensuring a structured learning experience.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Role Testing

Test Case Approach: Positive

Test Case ID: SLF-005

Test Case Title: Verify that social sharing features allow learners to share achievements.

Test Case Description: This test case validates the functionality of social sharing features, allowing learners to share their achievements and course progress on external platforms.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

* The learner is logged into the online learning platform.

* The learner has achieved a shareable milestone (e.g., completed a course, earned a badge).

Test Data: No test data needed.

Test Steps:

1. Complete a course or achieve a shareable milestone within the platform.

2. Locate the option to share the achievement on social media platforms.

3. Select a social media platform (e.g., Facebook, Twitter, LinkedIn) for sharing.

4. Verify that the platform's sharing interface opens correctly.

5. Customize the shared message (optional).

6. Submit the post to share the achievement.

7. Verify that the achievement is successfully shared on the selected social media platform.

Postconditions: None.

Expected Result: Learners should be able to effortlessly share their achievements and milestones on various social media platforms, promoting engagement and recognition.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Data - 66

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Authoring Tools

Feature description -

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

Userstory title -

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

Acceptance criteria -

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Content Authoring Tools

Test Case ID: TC_CAT_001

Test Case Title: Verify that instructors can create multimedia presentations using the content authoring tools.

Test Case Description: This test case verifies the functionality of the content authoring tools to create multimedia presentations. The instructor should be able to add different types of media (images, audio, video) and integrate them seamlessly into a cohesive presentation.

Test Suite: Content Authoring

Test Priority: High

Preconditions:

- Instructor account is created and logged in.

- Course is created and accessible.

Test Data: Sample presentation content including images, audio files, and video clips.

Test Steps:

1. Navigate to the content creation section for the chosen course.

2. Select the option to create a new multimedia presentation.

3. Upload and insert images, audio files, and video clips into the presentation.

4. Arrange and format the media content within the presentation.

5. Save and preview the created multimedia presentation.

Postconditions:

- The multimedia presentation is saved successfully.

- The presentation can be accessed and played back within the course.

Expected Result: The instructor should be able to create a multimedia presentation with integrated images, audio, and video content that plays back correctly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TC_CAT_002

Test Case Title: Verify that interactive modules can be developed and integrated into courses using the content authoring tools.

Test Case Description: This test case aims to confirm the ability for instructors to develop and integrate interactive modules within their courses, such as quizzes, drag-and-drop exercises, or interactive simulations.

Test Suite: Content Authoring

Test Priority: High

Preconditions:

- Instructor account is logged in.

- A course is selected for editing.

Test Data: Sample quiz questions, drag-and-drop elements, or simulation parameters.

Test Steps:

1. Access the content authoring tools for the selected course.

2. Choose the option to create a new interactive module.

3. Select the desired interactive module type (e.g., quiz, drag-and-drop).

4. Input content and define the functionality of the interactive elements.

5. Integrate the created module into the relevant section of the course.

6. Preview the course to confirm the module's functionality and integration.

Postconditions:

- Interactive module is saved and linked to the course.

- Students can access and interact with the module within the course.

Expected Result: The interactive module should be successfully created, integrated into the course, and function as intended, providing an engaging learning experience.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case ID: TC_CAT_003

Test Case Title: Verify that the content authoring tools support the integration of rich media resources, such as videos and animations.

Test Case Description: This test case verifies the ability for instructors to seamlessly upload, embed, or link to external rich media resources like videos and animations, enhancing the learning material.

Test Suite: Content Authoring

Test Priority: High

Preconditions:

- Instructor account is logged in.

- A course is open for editing.

Test Data: URLs of sample videos and animations hosted online, or local video/animation files.

Test Steps:

1. Navigate to the desired section of the course where the media will be added.

2. Select the option to embed a video or animation.

3. Provide the URL of the hosted media or upload the local file.

4. Adjust display settings (size, autoplay, etc.) for the media resource.

5. Save the changes and preview the course material to ensure proper integration.

Postconditions:

- The video or animation is successfully embedded/linked within the course material.

- The media content plays back correctly within the course platform.

Expected Result: The content authoring tools should support the seamless integration and playback of external rich media resources, enriching the course content.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CAT_004

Test Case Title: Verify that the content authoring tools are user-friendly and intuitive for instructors.

Test Case Description: This test case focuses on evaluating the usability and intuitiveness of the content authoring tools from an instructor's perspective, ensuring ease of use and navigation.

Test Suite: Content Authoring

Test Priority: High

Preconditions:

- Instructor account is created and logged in.

Test Data: N/A

Test Steps:

1. Access the content authoring tools section of the platform.

2. Explore the layout, menus, and options available.

3. Attempt to create different content types (presentations, modules, etc.).

4. Observe the clarity of instructions, tooltips, and error messages.

5. Evaluate the ease of finding necessary features and completing tasks.

Postconditions: N/A

Expected Result: The content authoring tools should have a clear and intuitive design, making it easy for instructors to navigate, create content, and find the required features without encountering unnecessary complexity.

Severity: Major

Priority: High

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CAT_005

Test Case Title: Verify that instructors can preview and edit content before publishing it to their courses.

Test Case Description: This test case ensures that instructors have the ability to preview and edit course content before it becomes live for students, ensuring quality control.

Test Suite: Content Authoring

Test Priority: High

Preconditions:

- Instructor account is logged in.

- Course content is in the process of creation or editing.

Test Data: N/A

Test Steps:

1. Create or navigate to existing course content within the authoring tool.

2. Locate and select the "Preview" option for the specific content.

3. Review the content in preview mode to ensure accuracy and desired presentation.

4. Make necessary edits to the content within the authoring tool.

5. Preview the edited content again to confirm changes are reflected correctly.

Postconditions:

- Changes made during editing should be saved correctly.

- Content should not be published until the instructor chooses to do so.

Expected Result: Instructors should be able to preview content in a student-facing view, make necessary edits, and only publish the content when they are satisfied with its final form.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

These test cases cover the given acceptance criteria and aim to thoroughly test the functionality, usability, and reliability of the content authoring tools within the online learning platform.

Data - 67

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Paths and Curriculum

Feature description -

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence and continuity in learners' educational journeys.

Userstory title -

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

Acceptance criteria -

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Learning Paths and Curriculum

Test Case ID: LP-001

Test Case Title: Verify that learning paths include sequential courses and modules.

Test Case Description: This test case verifies that a learning path consists of courses and modules arranged in a specific order, ensuring a logical progression of learning.

Test Suite: Learning Path Functionality

Test Priority: High

Preconditions:

- User is logged in as a learner.

- A learning path related to a specific subject is created.

Test Data: No test data needed.

Test Steps:

1. Navigate to the learning paths section.

2. Select a learning path.

3. Observe the displayed list of courses and modules.

4. Verify that the courses and modules within the learning path are arranged sequentially.

Postconditions: None

Expected Result: The learning path should display a series of courses and modules in a predefined, logical order.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: LP-002

Test Case Title: Verify that curricula are aligned with specific learning objectives.

Test Case Description: This test case checks if each learning path and its associated curriculum have clearly defined learning objectives, ensuring relevance to the learner's goals.

Test Suite: Curriculum Design and Alignment

Test Priority: High

Preconditions:

- User is logged in as a learner.

- A learning path with an associated curriculum is available.

Test Data: No test data needed.

Test Steps:

1. Navigate to a specific learning path.

2. Access the curriculum details for the selected learning path.

3. Identify the stated learning objectives for the curriculum.

4. Verify that the learning objectives are clear, measurable, achievable, relevant, and time-bound (SMART).

5. Confirm that the curriculum content, including courses and modules, aligns with the stated learning objectives.

Postconditions: None

Expected Result: The curriculum should have well-defined, SMART learning objectives that are directly addressed by the learning path content.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Content Review

Test Case Approach: Positive

Test Case ID: LP-003

Test Case Title: Verify that learners can track their progress within a learning path.

Test Case Description: This test case ensures learners can monitor their advancement through a learning path, providing feedback and motivation.

Test Suite: Progress Tracking and Reporting

Test Priority: High

Preconditions:

- User is logged in as a learner.

- The learner has enrolled in a learning path and started at least one course or module.

Test Data: No test data needed.

Test Steps:

1. Navigate to an active learning path.

2. Locate the progress tracking feature (e.g., progress bar, percentage completion).

3. Verify that the progress display accurately reflects the completed courses or modules within the learning path.

4. Complete a module or course within the learning path.

5. Observe the progress tracking feature to confirm it updates accordingly.

Postconditions: None

Expected Result: The platform should accurately display the learner's progress within the learning path, reflecting completed elements and overall progression.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: LP-004

Test Case Title: Verify that instructors can customize learning paths for different needs.

Test Case Description: This test case checks the instructor's ability to modify learning paths to cater to diverse learning styles, paces, or specific learning requirements.

Test Suite: Instructor Tools and Customization

Test Priority: Medium

Preconditions:

- User is logged in as an instructor.

- An existing learning path is accessible.

Test Data:

- Modified course order.

- Additional resource links.

- Adjusted learning objectives.

Test Steps:

1. Access the instructor dashboard.

2. Navigate to the learning path management section.

3. Select an existing learning path for editing.

4. Attempt to customize the learning path by:

    - Reordering courses or modules.

    - Adding supplementary resources.

    - Modifying learning objectives.

5. Save the changes made to the learning path.

Postconditions: Revert any customization to the original learning path after the test.

Expected Result: The platform should allow instructors to make necessary adjustments to the learning path content and structure, providing flexibility in catering to diverse learner needs.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: LP-005

Test Case Title: Verify that notifications are sent for upcoming courses or milestones.

Test Case Description: This test case validates the notification system, ensuring learners receive timely reminders and updates regarding their learning paths.

Test Suite: Notifications and Communication

Test Priority: Medium

Preconditions:

- User is logged in as a learner.

- The learner has enabled notifications for learning path activities.

Test Data:

- Scheduled start date of a course within the learning path.

- Upcoming milestone within the learning path.

Test Steps:

1. Enroll in a learning path with a future course start date or approaching milestone.

2. Verify that the platform sends a notification:

    - Before the start date of a new course within the learning path.

    - When an important milestone within the learning path is approaching.

3. Check for notifications through available channels (e.g., email, platform dashboard).

Postconditions: None

Expected Result: The platform should successfully deliver timely notifications to learners, informing them of upcoming courses or milestones within their chosen learning paths.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

--

Data - 68

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Attendance Tracking

Feature description -

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

Userstory title -

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Acceptance criteria -

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Attendance Tracking Feature

Test Case ID: TC_ATT_01

Test Case Title: Verify that learner attendance is automatically recorded during live sessions.

Test Case Description: This test case verifies the system's ability to automatically track and record learner attendance throughout a live session.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- A live session is scheduled and active.

- The learner has successfully joined the live session.

- The learner's device maintains a stable internet connection.

Test Data: No test data needed

Test Steps:

1. Log in as a learner.

2. Join the scheduled live session.

3. Remain in the session for a specific duration.

4. End the session.

5. Navigate to the learner's attendance record.

Postconditions:

- Attendance record should reflect the session attendance.

Expected Result: The learner's attendance record should accurately display the date, start time, end time, and duration of the live session attended.

Severity: Critical

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_ATT_02

Test Case Title: Verify that instructors can view detailed attendance reports for each live session.

Test Case Description: This test case validates the functionality for instructors to access and view comprehensive attendance reports for each live session they conduct.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- An instructor account exists with assigned live sessions.

- At least one live session has been conducted with recorded learner attendance.

Test Data: No test data needed

Test Steps:

1. Log in as the instructor.

2. Navigate to the live session management section.

3. Select the specific live session for which to view the attendance report.

4. Access the attendance report.

Postconditions:

- The report should be available for download or export.

Expected Result: The attendance report should accurately display the list of learners, their attendance status (present, absent, late), session duration, and potentially additional participation metrics.

Severity: Major

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_ATT_03

Test Case Title: Verify that learners can access and view their individual attendance records.

Test Case Description: This test case ensures that learners can independently access and review their attendance history for all attended live sessions within the platform.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- A learner account exists with recorded attendance data from past live sessions.

Test Data: No test data needed

Test Steps:

1. Log in as a learner.

2. Navigate to the learner profile or dashboard.

3. Access the attendance records section.

Postconditions: None

Expected Result: A comprehensive log of the learner's attendance should be displayed, including details such as session name, date, time, duration, and attendance status (present, absent).

Severity: Major

Priority: High

Type of Testing: System Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_ATT_04

Test Case Title: Verify that attendance data is integrated with participation metrics in instructor reports.

Test Case Description: This test case checks the integration of attendance data with other participation metrics in the instructor's report, ensuring a comprehensive view of learner engagement.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- An instructor account exists with assigned live sessions that include recorded participation metrics (e.g., chat activity, poll responses).
- At least one live session has been conducted with recorded learner attendance and participation data.

Test Data: No test data needed

Test Steps:

1. Log in as the instructor.
2. Navigate to the live session reports.
3. Select a session report that contains both attendance and participation data.
4. Verify that attendance information (present, absent) is displayed alongside other participation metrics for each learner.

Postconditions: None

Expected Result: The report should showcase a correlated view of attendance and participation, allowing instructors to identify potential trends, such as learners who attend but show low engagement.

Severity: Minor

Priority: Medium

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: TC_ATT_05

Test Case Title: Verify that learners receive notifications for missed live sessions.

Test Case Description: This test case validates the system's functionality to identify missed sessions and promptly send corresponding notifications to absent learners.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- A learner account exists with notification preferences enabled for missed sessions.
- A live session is scheduled, and the learner does not attend.

Test Data: No test data needed

Test Steps:

1. Log in as a learner.
2. Do not join the scheduled live session.
3. Check the learner's designated notification channel (e.g., email, platform inbox) for a missed session notification.

Postconditions: None

Expected Result: The learner should receive a timely notification clearly indicating the missed session's title, date, and time.

Severity: Minor

Priority: Medium

Type of Testing: System Testing, Functional Testing

Test Case Approach: Negative

Data - 69

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Licensing and Rights Management

Feature description -

The platform implements robust content licensing and rights management features to protect intellectual property and ensure compliance with copyright laws. By managing content licenses, usage rights, and permissions, the platform safeguards the interests of content creators and providers, enabling secure and lawful distribution of educational materials.

Userstory title -

As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

Acceptance criteria -

['Content licenses are assigned to each resource.', 'Usage rights and permissions are clearly defined.', 'Content creators can set restrictions on usage and distribution.', 'Instructors and learners are informed of content licensing terms.', 'Notifications are sent for license expiration or renewal.']

Generated Test Cases:

## Test Cases for Content Licensing and Rights Management

Test Case ID: TC_CLRM_01

Test Case Title: Verify that content licenses are assigned to each resource.

Test Case Description: This test case verifies that each uploaded resource has a content license assigned to it.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* A content creator account is created and logged in.

* The content creator has uploaded at least one resource (e.g., document, video).

Test Data: No test data needed.

Test Steps:

1. Navigate to the content library or resource management section.

2. Select a previously uploaded resource.

3. Verify that the content license information is displayed alongside the resource details.

Postconditions: None.

Expected Result: A content license should be visibly assigned to the selected resource.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_CLRM_02

Test Case Title: Verify that usage rights and permissions are clearly defined for each content license.

Test Case Description: This test case checks if the platform provides clear and comprehensive information about usage rights and permissions associated with each available content license.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* A content creator account is created and logged in.

* The platform offers different content license options.

Test Data: No test data needed.

Test Steps:

1. Navigate to the content licensing section.

2. Select a specific content license type from the available options.

3. Verify that the details for the chosen license are displayed, including permitted usage rights (e.g., viewing, downloading, modifying) and any limitations or restrictions.

Postconditions: None

Expected Result: Each content license option should have a clear definition of allowable usage rights and permissions, easily understandable by the user.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CLRM_03

Test Case Title: Verify that content creators can set restrictions on the usage and distribution of their resources.

Test Case Description: This test case ensures that content creators can apply specific restrictions on how their resources are used and distributed based on the chosen content license.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* A content creator account is created and logged in.

* The platform offers options for setting usage restrictions.

Test Data:

* Sample resource for uploading.

* Different content license options with varying restriction levels.

Test Steps:

1. Upload a new resource to the platform.

2. During the upload or resource editing process, choose a content license from the available options.

3. If the chosen license allows for custom restrictions, attempt to define limitations like download restrictions, print disablement, or limited view counts.

4. Save the resource with the applied restrictions.

5. Verify that the selected restrictions are enforced when accessing the resource.

Postconditions: Remove the uploaded test resource.

Expected Result: Content creators should be able to successfully apply and enforce usage and distribution restrictions according to the chosen content license.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Positive, Negative

Test Case ID: TC_CLRM_04

Test Case Title: Verify that instructors and learners are informed of content licensing terms.

Test Case Description: This test case checks that both instructors and learners can easily access and understand the licensing terms associated with the content they are interacting with on the platform.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

* A course is created with resources that have different content licenses applied.

* Both instructor and learner accounts are available and enrolled in the course.

Test Data: No test data needed.

Test Steps:

1. Log in as an instructor and navigate to the course content.

2. Verify that the content license information is displayed alongside each resource within the course.

3. Log in as a learner enrolled in the same course.

4. Navigate to the course content and verify that the content license information is clearly visible for each resource.

Postconditions: None.

Expected Result: Content licensing terms should be clearly presented and easily accessible to both instructors and learners within the course context.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CLRM_05

Test Case Title: Verify that notifications are sent for license expiration or renewal.

Test Case Description: This test case ensures that appropriate notifications are sent to content creators regarding upcoming license expirations or required renewals.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

* The platform supports content licenses with expiration dates.

* Email notification functionality is implemented.

Test Data:

* Content resource with a soon-to-expire license.

Test Steps:

1. As a content creator, upload a resource and assign a content license with a defined expiration date (set to a near future date).

2. Monitor the content creator's associated email account.

3. Verify that timely notifications are received prior to the license expiration date, reminding the creator about the upcoming expiry and providing renewal options.

Postconditions: Remove the uploaded test resource. Adjust the content license expiration date to its original value if modified for testing.

Expected Result: The content creator should receive timely and informative email notifications about content license expirations and renewal procedures.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Integration 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 Cases for Content Versioning Feature:

Test Case ID: TC_CV_001

Test Case Title: Verify that instructors can create and save multiple versions of content.

Test Case Description: This test case verifies that instructors can successfully create and save multiple versions of their course content without any issues.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* The instructor is logged into the platform.

* The instructor has an existing course.

Test Data: No test data needed

Test Steps:

1. Navigate to the course content editing section.

2. Make changes to existing content or add new content.

3. Select the "Save New Version" option.

4. Verify that a new version of the content is created and saved.

5. Repeat steps 2-4 multiple times to create multiple versions.

Postconditions:

* Multiple versions of the content are saved.

* Each version is accessible.

Expected Result: The system should successfully create and save each new version of the content, allowing for multiple revisions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_CV_002

Test Case Title: Verify that version history is maintained and accessible.

Test Case Description: This test case ensures the platform accurately records and displays the version history for any piece of course content.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* The instructor is logged in.

* The course has content with multiple saved versions.

Test Data: No test data needed

Test Steps:

1. Navigate to a piece of content with multiple versions.

2. Locate and access the "Version History" option.

3. Verify that a list of all saved versions is displayed.

4. Verify the list includes version numbers, timestamps, and author information.

Postconditions: None

Expected Result: The system should display a complete and accurate version history, including all relevant details for each version.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CV_003

Test Case Title: Verify that instructors can compare different versions of content.

Test Case Description: This test case checks the functionality for instructors to compare any two versions of their content to see the changes made.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* The instructor is logged in.

* The chosen content has at least two saved versions.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Version History" of a piece of content.

2. Select two different versions to compare.

3. Activate the "Compare" functionality.

4. Verify that differences between the selected versions are clearly highlighted.

Postconditions: None

Expected Result: The system should provide a clear and understandable comparison view highlighting all changes between the selected versions.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: TC_CV_004

Test Case Title: Verify that previous versions can be restored if needed.

Test Case Description: This test case validates the functionality for instructors to revert to a previous version of their course content.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

* The instructor is logged in.

* The content has at least two saved versions.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Version History" of a piece of content.

2. Select a previous version to restore.

3. Confirm the restore action.

4. Verify that the chosen version is restored as the current active version.

Postconditions:

* The selected version is the active version of the content.

* A new version reflecting the restoration might be created (depending on the system's design).

Expected Result: The system should allow for seamless restoration of previous versions, making the selected version the active one.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_CV_005

Test Case Title: Verify that notifications are sent for significant content updates.

Test Case Description: This test case ensures the system correctly triggers and sends notifications to instructors upon making significant updates to the course content.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

* The instructor is logged in.

* Notification settings for content updates are enabled.

Test Data:

* Significant content update (e.g., adding a new module, deleting a section).

Test Steps:

1. Make a significant update to the course content.

2. Save the changes as a new version.

3. Verify that a notification is received.

4. Check the notification content for accuracy and clarity.

Postconditions: None

Expected Result: The system should send a timely and informative notification to the instructor regarding the significant content update.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Data - 71

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

API Integration

Feature description -

The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

Userstory title -

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

Acceptance criteria -

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

Generated Test Cases:

## Test Cases for Online Learning Platform - API Integration

Here are 5 test cases designed to cover the acceptance criteria for API Integration:

Test Case ID: API-001

Test Case Title: Verify that APIs are available for key platform functionalities.

Test Case Description: This test verifies the availability of APIs for core platform functionalities like user authentication, course retrieval, and content access. The test will attempt to access these APIs and validate their responses.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API documentation is available.

- API endpoints are deployed and accessible.

- Valid API credentials are obtained.

Test Data: No test data needed.

Test Steps:

1. Access the API documentation.

2. Identify API endpoints for key functionalities: user login, course listing, retrieving course content.

3. Send API requests to each identified endpoint using valid credentials.

4. Verify the response status code (e.g., 200 OK).

5. Inspect the response body to ensure it contains expected data and format.

Postconditions: None.

Expected Result: All tested API endpoints for key functionalities should respond with a success status code and provide the expected data in the correct format.

Severity: Blocker

Priority: High

Type of Testing: API Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: API-002

Test Case Title: Verify that comprehensive API documentation is provided.

Test Case Description: This test ensures the provided API documentation is clear, complete, and easy to understand for developers. It checks for the presence of essential information like authentication methods, endpoint descriptions, request/response examples, and error handling.

Test Suite: API Integration

Test Priority: High

Preconditions: API documentation is accessible.

Test Data: No test data needed.

Test Steps:

1. Review the API documentation structure and organization.

2. Check for the presence of authentication instructions and examples.

3. Verify each API endpoint has a clear description, expected input parameters, and sample request/response bodies.

4. Ensure error codes and messages are documented with clear explanations.

5. Attempt to execute an API call solely based on the documentation instructions.

Postconditions: Any identified gaps or inconsistencies in the documentation are reported for correction.

Expected Result: API documentation should be comprehensive, well-structured, and provide all necessary information for successful API integration.

Severity: Critical

Priority: High

Type of Testing: Documentation Testing

Test Case Approach: Positive

Test Case ID: API-003

Test Case Title: Verify that secure access protocols are implemented.

Test Case Description: This test validates the implementation of secure access protocols for API authentication and data transmission. It checks for the use of HTTPS, API keys, and token-based authentication to prevent unauthorized access.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API endpoints are deployed and accessible.
- Knowledge of implemented security protocols.

Test Data:

- Invalid API credentials.
- Expired API token.

Test Steps:

1. Verify that all API communication uses HTTPS protocol.

2. Attempt to access an API endpoint without authentication credentials.

3. Attempt to access an API endpoint with invalid credentials.

4. Attempt to access an API endpoint with an expired token.

5. Analyze server logs for any unauthorized access attempts.

Postconditions: None.

Expected Result: All attempts to access APIs without proper authentication or using insecure methods should be rejected with appropriate error messages.

Severity: Blocker

Priority: High

Type of Testing: Security Testing, API Testing

Test Case Approach: Negative

Test Case ID: API-004

Test Case Title: Verify that third-party applications can authenticate and access APIs.

Test Case Description: This test simulates the integration process of a third-party application with the platform APIs. It verifies the authentication flow, data retrieval, and any limitations imposed on third-party applications.

Test Suite: API Integration

Test Priority: High

Preconditions:

- A test third-party application is set up.

- API credentials are provided for the test application.

Test Data:

- Test application credentials.

Test Steps:

1. Using the test application, initiate the authentication process with the platform API.

2. Verify the successful retrieval of an access token.

3. Use the access token to access various API endpoints.

4. Perform basic operations like fetching data and sending requests.

5. Observe any restrictions or limitations imposed on the third-party application.

Postconditions: Revoke the access token of the test application.

Expected Result: The test application should successfully authenticate, access APIs, and perform allowed actions within the defined limitations.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, API Testing

Test Case Approach: Positive

Test Case ID: API-005

Test Case Title: Verify that rate limiting and monitoring features are in place.

Test Case Description: This test ensures the platform has implemented rate limiting to prevent abuse and monitors API usage for potential issues. It attempts to exceed request limits and checks for appropriate responses and logging.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

- Knowledge of implemented rate limits.

- Access to server logs for monitoring.

Test Data: No test data needed.

Test Steps:

1. Identify the rate limits for various API endpoints (requests per minute, requests per hour, etc.).

2. Send API requests exceeding the defined limits for a specific endpoint.

3. Observe the server response and check for "429 Too Many Requests" error.

4. Analyze the server logs for details of the rate limit exceeded event.

5. Verify the monitoring system captures API usage data accurately.

Postconditions: None.

Expected Result: The platform should enforce rate limiting by throttling excessive requests and logging all related events for monitoring and analysis.

Severity: Major

Priority: Medium

Type of Testing: Performance Testing, Security Testing, API Testing

Test Case Approach: Negative, Destructive

Data - 72

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Data Analytics and Insights

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
DA-001	Verify that data visualization tools are available for analysis	This test case verifies that the platform provides various data visualization tools for analyzing learner performance data.	Data Analytics	High	- Administrator account logged in. - Learner data available in the system. - Access to the Data Analytics dashboard.	No test data needed	1. Navigate to the Data Analytics dashboard. 2. Check for the availability of different visualization tools (e.g., charts, graphs, heatmaps). 3. Attempt to interact with each tool and verify functionality.	None	The platform should offer a variety of functional data visualization tools.	Blocker	High	Functional, Usability	Positive
DA-002	Verify that reports can be generated for learner performance metrics	This test case verifies that administrators can generate reports on various learner performance metrics.	Data Analytics	High	- Administrator account logged in. - Learner data available in the system. - Access to the reporting functionality.	No test data needed	1. Navigate to the reporting section. 2. Select desired learner performance metrics (e.g., course completion rate, quiz scores). 3. Choose the report format (e.g., PDF, CSV). 4. Generate the report. 5. Verify the report content for accuracy and completeness.	1. Delete or archive the generated report if necessary.	The platform should generate accurate and comprehensive reports based on the selected metrics and format.	Critical	High	Functional	Positive

| DA-003 | Verify that real-time analytics provide insights into engagement and outcomes | This test case

verifies that the platform provides real-time insights into learner engagement and outcomes. | Data Analytics | High | - Administrator account logged in. - Ongoing learner activity within the platform. - Access to the real-time analytics dashboard. | No test data needed | 1. Access the real-time analytics dashboard. 2. Observe the displayed data for various engagement metrics (e.g., active users, course progress). 3. Trigger learner activity within the platform (e.g., course completion, quiz submission). 4. Verify that the dashboard updates in real-time to reflect the new data. | None | The platform should display real-time data updates reflecting learner engagement and outcomes. | Major | High | Functional, Performance | Positive |
| DA-004 | Verify that data can be exported for further analysis | This test case verifies that administrators can export data in various formats for further analysis. | Data Analytics | Medium | - Administrator account logged in. - Learner data available in the system. - Access to the data export functionality. | No test data needed | 1. Navigate to the data export section. 2. Select the desired data set for export. 3. Choose the preferred export format (e.g., CSV, XLSX). 4. Initiate the export process. 5. Verify that the exported file is downloaded successfully and contains the expected data. | 1. Delete the downloaded file after verification if necessary. | The platform should allow exporting data in various formats, ensuring data integrity and completeness. | Minor | Medium | Functional | Positive |
| DA-005 | Verify that notifications are sent for significant data trends or anomalies | This test case verifies that the platform sends notifications for significant data trends or anomalies. | Data Analytics | Medium | - Administrator account logged in. - Notification settings configured. - Simulated data trend or anomaly exceeding predefined thresholds. | - Predefined thresholds for notifications. - Simulated data exceeding the thresholds. | 1. Configure notification settings for specific data trends and anomalies. 2. Simulate a data trend or anomaly exceeding the predefined threshold. 3. Monitor the notification channels (e.g., email, platform dashboard). 4. Verify that a notification is received promptly, accurately describing the triggered event. | 1. Reset the simulated data to normal levels. | The platform should send timely and accurate notifications for significant data trends or anomalies based on predefined thresholds and notification settings. | Major | Medium | Functional | Positive |

Data - 73

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Roles and Permissions

Feature description -

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

Userstory title -

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

Acceptance criteria -

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

Generated Test Cases:

## Test Cases for User Roles and Permissions:

Test Case ID: TC_URP_001

Test Case Title: Verify that administrators can create and assign user roles.

Test Case Description: This test case verifies the administrator's ability to create new user roles and assign them to users on the Online Learning Platform.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

* Administrator account is created and logged in.
* Users exist in the system without assigned roles.

Test Data:

* Role Name: "Instructor"
* Role Description: "Responsible for creating and managing course content."

Test Steps:

1. Navigate to the User Management section.
2. Click on the "Roles" tab.
3. Click on the "Create Role" button.
4. Enter the Role Name and Role Description.
5. Save the newly created role.

6. Select a user from the user list.

7. Assign the "Instructor" role to the selected user.

Postconditions:

* A new user role "Instructor" should be created and visible in the role list.

* The selected user should have the "Instructor" role assigned.

Expected Result: The administrator should successfully create a new user role and assign it to a user.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_URP_002

Test Case Title: Verify that permissions are customizable for each role.

Test Case Description: This test case verifies the administrator's ability to customize permissions for each user role on the platform.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

* Administrator account is created and logged in.
* At least one user role exists (e.g., "Student").

Test Data:

* Permission to grant: "Access Discussion Forum"

Test Steps:

1. Navigate to the User Management section.
2. Click on the "Roles" tab.
3. Select the "Student" role.
4. Locate the "Permissions" section.
5. Grant the "Access Discussion Forum" permission to the "Student" role.
6. Save the changes made to the role.

Postconditions:

* The "Student" role should have the "Access Discussion Forum" permission enabled.

Expected Result: The administrator should successfully grant specific permissions to a chosen user role.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_URP_003

Test Case Title: Verify that users are restricted to functionalities based on their roles.

Test Case Description: This test case verifies that users can only access functionalities allowed by their assigned roles on the platform.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

* Two user accounts are created with "Student" and "Instructor" roles assigned respectively.
* "Instructor" role has "Create Course" permission enabled, "Student" role does not.

Test Data:

* No test data needed.

Test Steps:

1. Log in as the user with the "Student" role.
2. Attempt to access the "Create Course" functionality.

Postconditions:

* No changes are made to user roles or permissions.

Expected Result: The user with the "Student" role should be denied access to the "Create Course" functionality and receive an appropriate error message.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Security Testing, System Testing

Test Case Approach: Negative

Test Case ID: TC_URP_004

Test Case Title: Verify that changes to roles and permissions are logged.

Test Case Description: This test case verifies that any changes made to user roles and permissions are logged by the system for audit purposes.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

* Administrator account is created and logged in.

Test Data:

* No test data needed.

Test Steps:

1. Navigate to the User Management section.
2. Click on the "Roles" tab.
3. Select any existing role and modify its permissions.
4. Save the changes made to the role.
5. Access the System Logs.

Postconditions:

* The System Logs should be accessible for review.

Expected Result: The System Logs should display an entry reflecting the changes made to the user role and permissions, including timestamp, user who made the changes and details of modifications.

Severity: Major

Priority: Medium

Type of Testing: Security Testing, System Testing

Test Case Approach: Positive

Test Case ID: TC_URP_005

Test Case Title: Verify that users are notified of their assigned roles and permissions.

Test Case Description: This test case verifies that users receive appropriate notifications about their assigned roles and permissions on the platform.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

* A new user account is created.

Test Data:

* Role to assign: "Student"

Test Steps:

1. As an administrator, assign the "Student" role to the newly created user.
2. Log in to the platform as the newly created user.

Postconditions:

* The new user should be logged in.

Expected Result: Upon login, the user should receive a notification (e.g., welcome message, dashboard prompt) indicating their assigned role ("Student") and a summary of their permissions.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing, System Testing

Test Case Approach: Positive

Data - 74

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Feedback and Rating System

Feature description -

The platform includes a user feedback and rating system, allowing learners to provide feedback on courses, instructors, and learning materials. This feature enables continuous improvement by gathering user insights and ratings, helping instructors refine their content and approach.

Userstory title -

As a learner, I want to provide feedback and ratings so that I can share my experience and help improve the platform.

Acceptance criteria -

['Learners can rate courses and instructors on a scale.', 'Feedback forms are available for detailed comments.', 'Instructors can view and respond to feedback.', 'Aggregated ratings are displayed publicly.', 'Anonymous feedback options are available.']

Generated Test Cases:

## Test Cases for User Feedback and Rating System

Test Case ID: TC_FF_001

Test Case Title: Verify that learners can rate courses and instructors on a scale.

Test Case Description: This test case verifies that the rating system allows learners to provide numerical ratings for both courses and instructors.

Test Suite: User Feedback

Test Priority: High

Preconditions:

- The learner is logged in.
- The learner has accessed a completed course.

Test Data: No test data needed.

Test Steps:

1. Navigate to a completed course page.
2. Locate the rating section for the course.
3. Select a rating from the available scale.
4. Locate the rating section for the instructor.
5. Select a rating from the available scale.

Postconditions:

- Ratings are saved and displayed on the course and instructor profiles.

Expected Result: The system should successfully record and display the selected ratings for both the course and the instructor.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_FF_002

Test Case Title: Verify that feedback forms are available for detailed comments.

Test Case Description: This test case ensures the feedback form allows learners to input detailed textual feedback for courses and instructors.

Test Suite: User Feedback

Test Priority: High

Preconditions:

- The learner is logged in.

- The learner has accessed a completed course.

Test Data: No test data needed.

Test Steps:

1. Navigate to a completed course page.

2. Locate the feedback section.

3. Enter textual feedback into the provided text box.

4. Submit the feedback.

Postconditions:

- The feedback is submitted and stored in the system.

Expected Result: The system should successfully submit and store the entered textual feedback.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_FF_003

Test Case Title: Verify that instructors can view and respond to feedback.

Test Case Description: This test case verifies that instructors can access, view, and provide responses to learner feedback.

Test Suite: User Feedback

Test Priority: High

Preconditions:

- An instructor account is logged in.

- Learner feedback is available for the instructor's course.

Test Data: No test data needed.

Test Steps:

1. Navigate to the instructor dashboard.

2. Access the feedback section for a specific course.

3. View submitted learner feedback.

4. Input a response to the feedback.

5. Submit the response.

Postconditions:

- The instructor?s response is saved and displayed alongside the learner's feedback.

Expected Result: The system should allow the instructor to view, respond to, and save responses to learner feedback.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Security Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_FF_004

Test Case Title: Verify that aggregated ratings are displayed publicly.

Test Case Description: This test case ensures the system calculates and displays aggregated ratings for courses and instructors publicly, allowing potential learners to assess them.

Test Suite: User Feedback

Test Priority: High

Preconditions:

- Multiple ratings exist for a course and instructor.

Test Data: No test data needed.

Test Steps:

1. Navigate to a course page as a guest user (not logged in).

2. Locate the rating section for the course.

3. Verify the displayed rating is an aggregate of all submitted ratings.

4. Locate the rating section for the instructor.

5. Verify the displayed rating is an aggregate of all submitted ratings for that instructor.

Postconditions: No specific postconditions.

Expected Result: The system should correctly calculate and display aggregated ratings for both the course and the instructor, visible even to non-logged-in users.

Severity: Minor

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: TC_FF_005

Test Case Title: Verify that anonymous feedback options are available.

Test Case Description: This test case confirms that learners can choose to provide feedback anonymously, protecting their identity if they desire.

Test Suite: User Feedback

Test Priority: High

Preconditions:

- The learner is logged in.

- The learner is on the feedback submission form.

Test Data: No test data needed.

Test Steps:

1. Navigate to the feedback submission form.

2. Locate the option to submit feedback anonymously.

3. Select the anonymous feedback option.

4. Submit feedback.

Postconditions:

- Feedback is submitted without any identifying information linked to the learner.

Expected Result: The system should successfully submit the feedback without associating it with the learner's identity.

Severity: Minor

Priority: High

Type of Testing: Functional Testing, Security Testing, Regression 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 Cases for Social Media Integration Feature:

Test Case 1

Test Case ID: SM-001

Test Case Title: Verify that learners can connect their social media accounts to the platform.

Test Case Description: This test case verifies the functionality of connecting various social media accounts (Facebook, Twitter, LinkedIn) to the learner's profile on the platform.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* User has an active account on the platform.

* User has active social media accounts (Facebook, Twitter, LinkedIn).

Test Data: Valid social media credentials.

Test Steps:

1. Navigate to the "Settings" or "Profile" section of the platform.

2. Locate the "Social Media Integration" option.

3. Choose a social media platform (Facebook, Twitter, LinkedIn) to connect.

4. Enter valid credentials for the selected social media account.

5. Authorize the platform to access the social media account.

Postconditions:

* Verify that the social media account is successfully linked to the platform.

* Check for appropriate confirmation messages.

Expected Result: The selected social media account should be successfully linked to the learner's profile, allowing for future sharing of achievements and course completion.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: SM-002

Test Case Title: Verify that achievements and course completions can be shared on social media.

Test Case Description: This test case checks the functionality of sharing achievements and course completion badges on connected social media accounts upon successful achievement or completion.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* User has successfully connected at least one social media account to their profile.

* User has an unlocked achievement or completed course available for sharing.

Test Data: None

Test Steps:

1. Navigate to the "Achievements" or "Courses" section, where the completed item is displayed.

2. Locate and click on the "Share" button associated with the achievement/course completion.

3. Choose the connected social media platform to share the achievement/course completion.

4. (Optional) Add a personalized message to the post.

5. Confirm the sharing action.

Postconditions:

* Verify the successful posting of the achievement/completion on the chosen social media platform.

Expected Result: The platform should successfully share the achievement or course completion on the selected social media account, along with any personalized message.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: SM-003

Test Case Title: Verify that privacy settings control the visibility of shared content.

Test Case Description: This test case checks if the platform respects user-defined privacy settings when sharing content on social media.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* User has connected at least one social media account.

* User has the option to configure privacy settings for social media sharing (Public, Friends Only, Private).

Test Data: None

Test Steps:

1. Navigate to the platform's privacy settings related to social media sharing.

2. Configure the desired privacy level for shared content (Public, Friends Only, Private).

3. Share an achievement or course completion while logged into the connected social media account.

4. Check the visibility of the shared content on the social media platform based on the chosen privacy setting.

Postconditions:

* Verify that the shared content's visibility aligns with the configured privacy settings.

Expected Result: The shared content's visibility on the social media platform should adhere to the user's selected privacy setting (Public, Friends Only, Private).

Severity: Major

Priority: High

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Positive, Negative

Test Case 4

Test Case ID: SM-004

Test Case Title: Verify that instructors can promote courses and updates through social media.

Test Case Description: This test case checks the ability of instructors to share course information and updates directly to their connected social media accounts.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

* User has an instructor account on the platform.

* Instructor has successfully connected their social media accounts.

Test Data: Course link, update message

Test Steps:

1. Log in as an instructor and navigate to the course management dashboard.

2. Locate the option to "Share" or "Promote" the course.

3. Choose the desired social media platform for sharing.

4. (Optional) Add a custom message or promotional content alongside the course link.

5. Confirm sharing to the selected social media account.

Postconditions:

* Verify the successful posting of the course link and promotional message on the chosen social media platform.

Expected Result: The platform should successfully share the course information and update on the selected social media account with the provided message and link.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: SM-005

Test Case Title: Verify that analytics track the reach and engagement of shared content.

Test Case Description: This test case verifies the platform's ability to track and display analytics related to social media sharing, including reach, engagement, and click-through rates.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

* Users have shared achievements/course completions on social media.

* The platform has a dedicated section for displaying social media analytics.

Test Data: None

Test Steps:

1. Navigate to the analytics dashboard for social media.

2. Select a specific timeframe to view analytics data.

3. Verify that the platform displays relevant metrics for shared content, including:

    * Reach (number of people who saw the shared post)
    * Engagement (likes, comments, shares)
    * Click-through rate (percentage of users who clicked on the shared link)

Postconditions:

* None

Expected Result: The platform should accurately track and display key metrics related to the reach and engagement of shared content on social media, allowing administrators and instructors to measure the effectiveness of their social media efforts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Data-Driven Testing

Test Case Approach: Positive

Data - 76

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Activity Tracking

Feature description -

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity levels and progress towards fitness goals.

Userstory title -

As a user, I want to track my activities so that I can monitor my daily physical activity levels.

Acceptance criteria -

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']

Generated Test Cases:

## Fitness Tracker App - Activity Tracking Test Cases

Here are 5 test cases designed to test the acceptance criteria of the "Activity Tracking" feature in the Fitness Tracker App:

Test Case 1

Test Case ID: FT-AT-001

Test Case Title: Verify that the app accurately records data on distance, duration, pace, and calories burned during an activity.

Test Case Description: This test case verifies that the app accurately captures and displays the key metrics of an activity.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

* The user is logged in.

* The user has granted necessary permissions (location, health data).

Test Data: No test data needed.

Test Steps:

1. Start a new activity tracking session.

2. Simulate a walk/run/cycle (using a simulator or real device) for a specific distance, duration, and pace.

3. End the activity tracking session.

4. Check that the app displays the correct distance, duration, pace, and estimated calories burned.

Postconditions:

* Save the activity data.

* Clear the activity data for subsequent tests.

Expected Result: The app should accurately record and display the distance, duration, pace, and estimated calories burned for the activity session.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: FT-AT-002

Test Case Title: Verify that users can view activity summaries with relevant data.

Test Case Description: This test case verifies that users can easily access and view summaries of their tracked activities, including date, time, activity type, and key metrics.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

* The user is logged in.

* At least one activity has been recorded.

Test Data: No test data needed.

Test Steps:

1. Navigate to the activity history or summary section.

2. Select a previously recorded activity.

3. Verify that the activity summary screen displays the date, start/end time, activity type (e.g., walk, run, cycle), distance, duration, pace, and calories burned.

Postconditions: None

Expected Result: The app should display a comprehensive summary of the selected activity with all relevant data points.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: FT-AT-003

Test Case Title: Verify that activity data is stored and accessible for historical analysis.

Test Case Description: This test case checks the app's ability to store activity data over time and allow users to access past activity data for analysis.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

* The user is logged in.

* Multiple activities have been recorded over a period of time.

Test Data: No test data needed.

Test Steps:

1. Navigate to the activity history section.

2. Verify that the app displays activities recorded on different days, weeks, or months.

3. Select a past date range to view historical activity data.

4. Check that the app filters and displays activity data correctly for the selected period.

Postconditions: None

Expected Result: The app should store activity data and allow users to access and analyze their historical activity data based on different time periods.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Data Validation

Test Case Approach: Positive

Test Case 4

Test Case ID: FT-AT-004

Test Case Title: Verify that activity tracking can be paused and resumed without data loss.

Test Case Description: This test case ensures that users can pause and resume an activity tracking session without losing any recorded data.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

* The user is logged in.

* The user is currently tracking an activity.

Test Data: No test data needed.

Test Steps:

1. Start an activity tracking session.

2. After a short duration, pause the activity tracking.

3. Wait for a few minutes.

4. Resume the activity tracking session.

5. End the activity tracking session.

6. Check that the app accurately records the total duration, distance, and other metrics, including the paused time.

Postconditions:

* Save the activity data.

* Clear the activity data for subsequent tests.

Expected Result: The app should pause and resume the activity tracking without losing any data and accurately reflect the total activity time.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Experience Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: FT-AT-005

Test Case Title: Verify that users receive notifications for reaching activity milestones.

Test Case Description: This test case validates that the app sends appropriate notifications to the user upon achieving certain activity milestones (e.g., distance goals, calorie goals).

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

* The user is logged in.

* The user has enabled notifications for activity milestones.

* The user has set an activity milestone (e.g., distance goal of 5km).

Test Data: No test data needed.

Test Steps:

1. Start an activity tracking session.

2. Continue the activity until the predefined milestone is reached.

3. Check if the app triggers a notification to the user informing them about the achieved milestone.

Postconditions: None

Expected Result: Upon reaching the predefined activity milestone, the app should send a notification to the user.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Notification 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 Cases for Fitness Tracker App - GPS Tracking Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
GPS-001	Verify that GPS tracking accurately records route data	This test case verifies that the app accurately records the user's route data, including distance, duration, and pace, while using GPS tracking during an outdoor activity.	GPS Tracking	High	- User has granted location permissions to the app. - User has a stable internet connection. - User's device has GPS enabled.	No test data needed	1. Start a new outdoor activity tracking session. 2. Walk or run a pre-determined route with varying speeds and elevations. 3. End the tracking session. 4. View the recorded route details.	- Close the app.	Recorded route data should accurately reflect the actual route taken, including distance, duration, pace, and elevation changes.	Blocker	High	Functional, Acceptance, Regression	Positive

| GPS-002 | Verify that users can view detailed maps and elevation profiles | This test case verifies that users

can access and interact with detailed maps and elevation profiles for their recorded GPS tracked activities. | GPS Tracking | High | - User has completed a GPS tracked activity. - User has a stable internet connection. | No test data needed | 1. Open the app and navigate to the "History" section. 2. Select a completed GPS tracked activity. 3. Verify that the map displays the recorded route. 4. Check for the availability and accuracy of zooming and panning functionalities on the map. 5. Locate and verify the elevation profile displaying elevation changes throughout the route. | - Close the app. | The app should display a detailed map of the recorded route, allowing users to zoom and pan. An accurate elevation profile corresponding to the route should also be accessible. | Critical | High | Functional, Usability, Acceptance | Positive |

| GPS-003 | Verify that performance data includes location-based metrics | This test case verifies that the app displays relevant location-based metrics alongside other performance data for GPS tracked activities. | GPS Tracking | Medium | - User has completed a GPS tracked activity. | No test data needed | 1. Open the app and navigate to the "History" section. 2. Select a completed GPS tracked activity. 3. View the detailed performance data for the selected activity. | - Close the app. | In addition to standard performance data, the app should display location-based metrics like average speed for specific segments of the route, elevation gain/loss, and best pace achieved at different points. | Major | Medium | Functional, Acceptance | Positive |

| GPS-004 | Verify that routes can be saved and revisited | This test case verifies that users can save completed GPS tracked routes and access them later for review or to use as a guide for future activities. | GPS Tracking | Medium | - User has completed a GPS tracked activity. | No test data needed | 1. Open the app and navigate to the "History" section. 2. Select a completed GPS tracked activity. 3. Save the route using the provided functionality. 4. Navigate to the "Saved Routes" or similar section. 5. Verify that the saved route is present and accessible. | - Close the app. | Users should be able to save completed GPS tracked routes and easily access them later within the app. | Minor | Medium | Functional, Usability, Acceptance | Positive |

| GPS-005 | Verify that notifications are provided for deviations from planned routes | This test case verifies that the app sends notifications to the user if they deviate significantly from their planned route during a GPS tracked activity. | GPS Tracking | Medium | - User has created and saved a planned route. - User has started a GPS tracked activity following the saved route. - User's device has notification permissions enabled for the app. | No test data needed | 1. Create a new planned route within the app. 2. Start a GPS tracked activity,

choosing to follow the planned route. 3. Intentionally deviate significantly from the planned route. 4. Observe if the app provides a notification about the deviation. | - Close the app. | The app should promptly notify the user if they deviate significantly from their planned route during a GPS tracked activity. | Major | Medium | Functional, Usability, Acceptance | Negative |

Data - 78

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Heart Rate Monitoring

Feature description -

The app monitors heart rate in real-time using built-in sensors or compatible wearable devices, providing insights into exercise intensity, recovery, and cardiovascular health. Heart rate data helps users optimize their workouts, maintain target heart rate zones, and track improvements in fitness over time.

Userstory title -

As a user, I want to monitor my heart rate so that I can optimize my workouts and track my cardiovascular health.

Acceptance criteria -

['Heart rate data is recorded in real-time.', 'Users can set and monitor target heart rate zones.', 'Historical heart rate data is stored and viewable.', 'Alerts are provided for abnormal heart rate readings.', 'Data is compatible with various wearable devices.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Heart Rate Monitoring Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
HRM-001	Verify that the app records heart rate data in real-time.	This test case verifies that the app accurately captures and displays heart rate data from the sensor in real-time.	Heart Rate Monitoring	High	- User has granted heart rate monitoring permission. <br> - App is connected to a compatible heart rate sensor or device. <br> - User is actively wearing the sensor.	No test data needed.	1. Launch the app and navigate to the heart rate monitoring screen. <br> 2. Initiate a workout or activity that elevates heart rate. <br> 3. Observe the heart rate display on the app screen. <br> 4. Compare the displayed heart rate to the reading on the sensor device (if applicable).	- Close the app.	The heart rate displayed on the app should update dynamically and reflect the user's current heart rate as detected by the sensor, with minimal latency.	Blocker	High	Functional, Real-time, Integration	Positive
HRM-002	Verify that users can set and monitor target heart rate zones.	This test case ensures users can define their target heart rate zones and the app provides visual feedback during workouts.	Heart Rate Monitoring	High	- User is logged into the app.	- Minimum heart rate: 120 bpm <br> - Maximum heart rate: 180 bpm	1. Navigate to the heart rate settings or profile section. <br> 2. Locate the option to set target heart rate zones. <br> 3. Input the desired minimum and maximum heart rate values for the target zone. <br> 4. Start a workout and observe the visual feedback on the heart rate display.	- Save the configured target heart rate zones.	The app should visually represent the target heart rate zone (e.g., with color-coded indicators) and alert the user when they are within or outside of their defined zone during workouts.	Critical	High

Functional, Usability | Positive |

| HRM-003 | Verify that historical heart rate data is stored and viewable. | This test case validates the app's ability to store historical heart rate data and allow users to access and view it. | Heart Rate Monitoring | Medium | - User has previously recorded heart rate data. | No test data needed. | 1. Navigate to the heart rate data history section within the app.
 2. Select a specific date or time period to view.
 3. Observe the displayed historical heart rate data, including timestamps and any associated workout information. | - No specific postconditions. | The app should display historical heart rate data accurately, organized chronologically, and potentially with options to filter or analyze the data. | Major | Medium | Functional, Data Management | Positive |

| HRM-004 | Verify that alerts are provided for abnormal heart rate readings. | This test case ensures the app generates timely alerts for heart rate readings that fall outside a defined safe range. | Heart Rate Monitoring | Critical | - User has enabled heart rate alerts.
 - User has set their resting and maximum heart rate in the app. | - Abnormal heart rate: 200 bpm | 1. Simulate or trigger an abnormally high heart rate reading from the sensor.
 2. Observe the app's response to this reading. | - Dismiss the alert. | The app should immediately generate a clear and noticeable alert to the user, indicating the abnormal heart rate reading and potentially providing recommendations or guidance. | Critical | High | Functional, Safety-critical | Negative |

| HRM-005 | Verify that heart rate data is compatible with various wearable devices. | This test case confirms the app's compatibility with a range of wearable devices for accurate data synchronization. | Heart Rate Monitoring | High | - User has a compatible wearable device. | - List of compatible devices. | 1. Pair the app with a compatible wearable device according to the instructions.
 2. Initiate a workout or activity that elevates heart rate.
 3. Compare the heart rate data displayed in the app with the data recorded by the wearable device. | - Unpair the devices after testing. | The app should successfully connect and synchronize heart rate data with different compatible devices, ensuring accurate and consistent data across platforms. | Blocker | High | Integration, Compatibility | Positive |

Data - 79

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Calorie Tracking

Feature description -

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

Userstory title -

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

Acceptance criteria -

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Calorie Tracking Feature

Test Case Creation Template:

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data |

Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing | Test Case Approach |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CT-001 | Verify that users can log daily food intake and calorie consumption. | This test case verifies that users can successfully input their food intake and the app correctly records the calorie consumption for each entry. | Calorie Tracking | High | - User account created and logged in.
- Food database populated. | - Food Item: Banana
- Quantity: 1
- Calories per serving: 105 | 1. Navigate to the "Calorie Tracking" section.
2. Tap on "Add Food".
3. Search for "Banana" in the food database.
4. Select "Banana" and input quantity "1".
5. Confirm food log entry. | - Verify food log displays "Banana (1 serving, 105 calories)". | Food log entry successfully created and displays correct calorie information. | Critical | High | Functional, Regression | Positive |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CT-002 | Verify that the app calculates and displays total calories consumed and burned. | This test case ensures that the app accurately calculates and displays the user's total daily calorie intake and subtracts calories burned from exercises, reflecting the net calorie balance. | Calorie Tracking | High | - User account created and logged in.
- Food log entries added.
- Exercise log entries added. | - Food Log: 2000 calories
- Exercise Log: 500 calories burned | 1. Navigate to the "Dashboard".
2. Observe the "Calories Consumed" value.
3. Observe the "Calories Burned" value.
4. Observe the "Net Calories" value. | - N/A | "Calories Consumed" displays 2000.
- "Calories Burned" displays 500.
- "Net Calories" displays 1500 (2000 consumed - 500 burned). | Critical | High | Functional, Regression | Positive |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CT-003 | Verify that users can set daily calorie goals and track progress. | This test case validates that users can set personalized daily calorie goals and the app accurately tracks their progress toward these goals. | Calorie Tracking | High | - User account created and logged in. | - Daily Calorie Goal: 1800 calories | 1. Navigate to the "Calorie Tracking" section.
2. Tap on "Set Calorie Goal".
3. Enter "1800" as the daily calorie goal.
4. Confirm goal setting.
5. Log food intake throughout the day. | - Verify the progress bar updates based on calorie intake. | - Calorie goal is saved and displayed.
- Progress bar accurately reflects calorie intake compared to the set goal. | Major | High | Functional, Regression | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| CT-004 | Verify that food items and calorie information are searchable in the database. | This test case ensures the functionality of the food database search, allowing users to easily find and log their consumed food items. | Calorie Tracking | High | - User account created and logged in. | - Search Term: "Apple" | 1. Navigate to the "Calorie Tracking" section.
2. Tap on "Add Food".
3. Enter "Apple" in the search bar. | - N/A | - Search results display relevant food items containing "Apple" (e.g., Apple, Green Apple, Apple Pie).
- Each search result displays corresponding calorie information. | Major | High | Functional, Usability | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| CT-005 | Verify that users receive alerts for exceeding calorie goals. | This test case ensures that the app provides timely notifications to users when they have exceeded their predefined daily calorie limits, promoting awareness and self-monitoring. | Calorie Tracking | High | - User account created and logged in.
- Daily Calorie Goal set.
- Push notifications enabled. | - Daily Calorie Goal: 1800 calories
- Calorie Intake: 2000 calories | 1. Set a daily calorie goal (e.g., 1800 calories).
2. Log food intake exceeding the set calorie goal (e.g., 2000 calories). | - Verify the alert message content. | An alert notification is displayed, stating "You have exceeded your daily calorie goal by 200 calories." | Major | High | Functional, Usability | Negative |

Data - 80

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

## Fitness Tracker App - Sleep Tracking Feature Test Cases

Here are 5 test cases designed to cover the acceptance criteria for the sleep tracking feature:

Test Case ID: ST-001

Test Case Title: Verify that the app records sleep data including duration and quality.

Test Case Description: This test case verifies that the app accurately captures and displays sleep duration and quality after a sleep session.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- The user has granted necessary permissions for sleep tracking.

- The sleep tracking feature is enabled.

- The user has completed a sleep session with the app actively tracking.

Test Data: No test data needed

Test Steps:

1. Initiate sleep tracking before going to sleep.

2. Stop sleep tracking upon waking up.

3. Navigate to the sleep tracking dashboard.

Postconditions:

- Sleep data remains available for future reference.

- The app is ready for the next sleep tracking session.

Expected Result: The dashboard displays the sleep duration and an overall sleep quality score (e.g., poor, fair, good, excellent).

Severity: Blocker

Priority: High

Type of Testing: Functional Testing, Acceptance Testing

Test Case Approach: Positive

Test Case ID: ST-002

Test Case Title: Verify that users can view detailed sleep stages and patterns.

Test Case Description: This test case verifies the app's ability to present detailed information regarding sleep stages (e.g., light, deep, REM) and overall sleep patterns.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- Sleep tracking data from at least one complete sleep session is available.

Test Data: No test data needed

Test Steps:

1. Navigate to the sleep tracking section of the app.

2. Select the desired sleep session to view details.

Postconditions: None

Expected Result: The app displays a visual representation (e.g., graph, chart) of the sleep stages within the chosen session, including time spent in each stage and potential patterns.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: ST-003

Test Case Title: Verify that the app provides insights into sleep disturbances.

Test Case Description: This test case validates that the app identifies and informs users about potential sleep disturbances detected during tracking.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- Sleep tracking data with potential disturbances (e.g., movements, heart rate spikes) is available.

Test Data: Simulated sleep data containing movement and heart rate variations.

Test Steps:

1. Access the sleep details for a session with simulated disturbances.

2. Review the provided sleep analysis and insights.

Postconditions: None

Expected Result: The app highlights periods of potential sleep disturbance, providing possible explanations (e.g., restlessness, external noise) based on the sensor data.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Data-Driven Testing

Test Case Approach: Positive

Test Case ID: ST-004

Test Case Title: Verify that users receive recommendations for improving sleep quality.

Test Case Description: This test case confirms that the app delivers personalized suggestions to users aimed at enhancing sleep quality based on collected sleep data.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- A history of sleep tracking data is available for analysis.

Test Data: Several sleep sessions with varying quality scores and potential patterns.

Test Steps:

1. Navigate to the sleep insights or recommendations section within the app.

Postconditions: None

Expected Result: The app presents personalized recommendations for sleep improvement (e.g., consistent sleep schedule, relaxing bedtime routine, optimized sleep environment) based on the user's sleep history.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ST-005

Test Case Title: Verify that sleep data can be viewed in daily, weekly, and monthly summaries.

Test Case Description: This test case ensures users can access and analyze their sleep data through various summary views.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- Sleep data spanning multiple days, weeks, and a month is available.

Test Data: No specific test data needed, but a range of dates with recorded sleep sessions.

Test Steps:

1. Navigate to the sleep tracking dashboard or history section.
2. Select and view daily, weekly, and monthly summaries of sleep data.

Postconditions: None

Expected Result: The app accurately presents sleep duration, quality trends, and other relevant metrics within the selected timeframe (daily, weekly, or monthly).

Severity: Critical

Priority: High

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 Cases for Fitness Tracker App - Workout Logging Feature

Test Case ID: FT-WL-001

Test Case Title: Verify that users can log various workout types with details.

Test Case Description: This test case verifies that the app allows users to log different workout types (e.g., running, swimming, weightlifting) and input specific details for each workout.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- The user is logged in to the app.

- The user navigates to the "Log Workout" section.

Test Data:

- Workout type: Running

- Duration: 45 minutes

- Intensity: Moderate

- Notes: Ran 5 miles outdoors.

Test Steps:

1. Select "Running" from the list of workout types.

2. Enter the duration as "45" and select "minutes" as the unit.

3. Choose "Moderate" as the intensity level.

4. In the "Notes" field, type "Ran 5 miles outdoors."

5. Save the workout log.

Postconditions:

- The workout log should be successfully saved.

- The log should display the correct workout type, duration, intensity, and notes.

Expected Result: The workout log is created and displayed with the correct details.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: FT-WL-002

Test Case Title: Verify that workout logs include duration, intensity, and notes.

Test Case Description: This test case ensures that the workout logging functionality captures and displays the duration, intensity level, and any user-entered notes for each workout entry.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- The user has logged at least one workout.

Test Data: No test data needed.

Test Steps:

1. Navigate to the workout log history.

2. Select a previously logged workout entry.

3. Verify that the workout details section displays the duration, intensity level, and notes entered during logging.

Postconditions: None

Expected Result: The workout log entry displays the correct duration, intensity, and user-entered notes.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case ID: FT-WL-003

Test Case Title: Verify that users can edit and delete workout entries.

Test Case Description: This test case checks the functionality for users to edit or delete existing workout log entries within the app.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- The user has logged at least one workout.

Test Data: No test data needed.

Test Steps:

1. Navigate to the workout log history.

2. Select a previously logged workout entry.

3. Locate and select the "Edit" option.

4. Modify the duration, intensity, or notes of the workout.

5. Save the changes.

6. Select the same workout entry again.

7. Locate and select the "Delete" option.

8. Confirm the deletion.

Postconditions:

- The selected workout log should reflect the edited information.

- The selected workout log should be removed from the log history.

Expected Result: The user can successfully edit and delete workout log entries.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: FT-WL-004

Test Case Title: Verify that workout data is stored and viewable for tracking progress.

Test Case Description: This test case ensures that all logged workout data is persistently stored by the app and can be accessed by the user for reviewing and tracking their fitness progress.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- The user has logged multiple workouts over different dates.

Test Data: No test data needed.

Test Steps:

1. Log out of the application and then log back in.

2. Navigate to the workout log history.

3. Verify that all previously logged workouts are displayed.

4. Check that the displayed workouts maintain the correct chronological order.

Postconditions: None

Expected Result: All previously logged workout data is accessible and displayed correctly after logging out and back into the application.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Data Persistence Testing

Test Case Approach: Positive

Test Case ID: FT-WL-005

Test Case Title: Verify that users can filter logs by date and workout type.

Test Case Description: This test case confirms that the app provides filtering options for users to easily search and view their workout logs based on specific dates or workout types.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- The user has logged multiple workouts of different types across various dates.

Test Data: No test data needed.

Test Steps:

1. Navigate to the workout log history.

2. Locate and utilize the filter function.

3. Apply a filter to display workouts from a specific date range.

4. Verify that only workouts within the selected date range are displayed.

5. Apply a filter to display workouts of a specific type (e.g., "Swimming").

6. Verify that only workouts of the selected type are displayed.

Postconditions: None

Expected Result: The workout log history accurately filters and displays entries based on the chosen date range and workout type.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 82

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Goal Setting

Feature description -

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

Userstory title -

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

Acceptance criteria -

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

Generated Test Cases:

## Test Case Creation for Fitness Tracker App - Goal Setting Feature

Test Case ID: FT-GS-001

Test Case Title: Verify that users can set goals for various fitness metrics.

Test Case Description: This test case verifies that users can successfully set goals for different fitness metrics offered by the app, such as steps taken, calories burned, distance traveled, and active minutes.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- The app is installed and launched.

- The user is logged in.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Goals' section of the app.

2. Select 'Set a new goal'.

3. Choose a fitness metric from the available options (steps, calories, distance, etc.).

4. Input the desired target value for the chosen metric.

5. Save the goal.

Postconditions:

- The newly created goal should be displayed in the user's list of active goals.

- The goal should reflect the chosen metric and target value.

Expected Result: The app should allow the user to successfully set a goal for the selected fitness metric with the specified target value.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: FT-GS-002

Test Case Title: Verify that progress tracking is available for each set goal.

Test Case Description: This test case checks if the app accurately tracks and displays the user's progress towards each set fitness goal, including visual representations like progress bars or charts.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- A user has set at least one fitness goal within the app.

- The user has recorded some activity data related to the set goal.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Goals' section.

2. Select the specific goal for which progress needs to be viewed.

3. Observe the displayed progress information, including current status, target value, and potentially visual aids.

Postconditions: None

Expected Result: The app should accurately display the user's progress towards the selected goal, visually representing the current status in relation to the target.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: FT-GS-003

Test Case Title: Verify that users receive motivational feedback and reminders related to their set goals.

Test Case Description: This test case ensures the app provides appropriate motivational messages and timely reminders to encourage users towards achieving their fitness goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- The user has set at least one fitness goal.
- Push notifications or in-app alerts are enabled for goal reminders.

Test Data: Goal nearing completion, goal completion.

Test Steps:

1. Simulate scenarios where the user is close to achieving a goal.
2. Observe if the app displays motivational messages encouraging the user to complete the goal.
3. Trigger a scenario where a goal is achieved.
4. Observe if the app provides congratulatory feedback.
5. Monitor the app for timely reminders related to ongoing goals.

Postconditions: None

Expected Result: The app should display appropriate motivational messages when users are close to achieving a goal and send timely reminders to keep them engaged.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: FT-GS-004

Test Case Title: Verify that users can adjust or update their set fitness goals.

Test Case Description: This test case validates the functionality for users to modify or update the target values or timelines of their existing goals within the app.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- A user has set at least one fitness goal within the app.

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Goals' section of the app.
2. Locate and select the specific goal that needs modification.
3. Choose the option to edit the goal details.
4. Modify either the target value, timeline, or other relevant parameters of the goal.
5. Save the changes made to the goal.

Postconditions:

- The updated goal with its modified parameters should be reflected in the user's list of active goals.

Expected Result: The app should successfully update the selected goal with the new parameters and reflect the changes in the user's goal list.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: FT-GS-005

Test Case Title: Verify that the app provides a summary of goal attainment.

Test Case Description: This test case checks if the app generates and presents a comprehensive summary of the user's goal achievement history, including achieved goals, missed goals, and relevant statistics.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- The user has been actively using the app and has a history of setting and engaging with fitness goals.

Test Data: Historical goal data

Test Steps:

1. Navigate to the 'Goals' or 'Progress' section (depending on app structure) where goal summary is accessible.
2. Observe the presented information, which should include a list of achieved goals, potentially missed goals, and other relevant details like dates of achievement.
3. Check for any visual representations of goal attainment history, such as graphs or charts.

Postconditions: None

Expected Result: The app should accurately display a summarized view of the user's goal attainment history, including achieved and potentially missed goals with relevant dates or statistics.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, User Interface 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 Cases for Fitness Tracker App - Workout Plans Feature

Test Case ID: WP-001

Test Case Title: Verify that users can select workout plans based on different fitness levels.

Test Case Description: This test case verifies that the app allows users to filter and select workout plans according to their specified fitness level (e.g., Beginner, Intermediate, Advanced).

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged into the app.

- User has navigated to the "Workout Plans" section.

Test Data: No test data needed.

Test Steps:

1. Open the Fitness Tracker app.

2. Navigate to the "Workout Plans" section.

3. Observe if the app provides filtering options for fitness levels.

4. Select a fitness level (e.g., Beginner).

5. Verify that the displayed workout plans match the chosen fitness level.

Postconditions: None.

Expected Result: The app should display workout plans specifically tailored to the selected fitness level.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: WP-002

Test Case Title: Verify that workout plans include detailed exercise routines and schedules.

Test Case Description: This test case verifies that each workout plan provides users with detailed information about the exercise routines, including sets, reps, rest time, and a clear weekly schedule.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged into the app.

- User is viewing a specific workout plan.

Test Data: No test data needed.

Test Steps:

1. Open the Fitness Tracker app.

2. Select a workout plan from the available options.

3. Observe if the plan displays a list of exercises with details (sets, reps, rest time).

4. Check for a clear weekly schedule indicating which exercises to perform on specific days.

5. Verify that the exercise descriptions are understandable and provide sufficient guidance.

Postconditions: None.

Expected Result: The workout plan should provide detailed exercise information and a clear schedule for users to follow.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: WP-003

Test Case Title: Verify that users can track their adherence and progress within a workout plan.

Test Case Description: This test case ensures the app allows users to mark completed workouts, track their progress towards plan completion, and view relevant metrics like calories burned or workouts completed.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged into the app.

- User has started following a workout plan.

Test Data: Sample completed workout data.

Test Steps:

1. Open the Fitness Tracker app.

2. Navigate to the active workout plan.

3. Mark a workout as completed after finishing it.

4. Check if the app updates the plan progress accordingly (e.g., percentage completed).

5. Verify that the app displays relevant progress metrics related to the plan.

Postconditions: None.

Expected Result: The app should accurately track the user's adherence to the plan and display relevant progress metrics.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: WP-004

Test Case Title: Verify that the app provides notifications and reminders for scheduled workouts.

Test Case Description: This test case checks if the app sends timely notifications and reminders to users about their scheduled workouts based on the chosen plan.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is logged into the app.

- User has enabled notifications for the app.
- User has a workout plan with scheduled workouts.

Test Data: Scheduled workout time.

Test Steps:

1. Open the Fitness Tracker app.
2. Ensure notifications are enabled for the app.
3. Select a workout plan with scheduled workouts.
4. Wait for the scheduled workout time.
5. Observe if the app sends a notification or reminder about the upcoming workout.

Postconditions: Disable notifications if needed.

Expected Result: The app should send timely notifications to remind users about their scheduled workouts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case ID: WP-005

Test Case Title: Verify that users can adjust workout plans based on their progress and preferences.

Test Case Description: This test case ensures users can adjust their chosen workout plans, change the fitness level, or modify individual workout schedules based on their needs.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is logged into the app.
- User has an active workout plan.

Test Data: New fitness level or adjusted workout schedule.

Test Steps:

1. Open the Fitness Tracker app.

2. Navigate to the active workout plan settings.

3. Attempt to change the fitness level of the plan.

4. Try to modify the schedule of individual workouts within the plan.

5. Save the changes and verify if the plan reflects the adjustments made.

Postconditions: None.

Expected Result: The app should allow users to adjust their workout plans, including fitness level and individual workout schedules.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive, Negative

Data - 84

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Community Support

Feature description -

Users can connect with a community of like-minded individuals, share progress updates, participate in challenges, and offer support and encouragement to fellow users. Community support fosters accountability, motivation, and a sense of camaraderie among users striving towards similar fitness goals.

Userstory title -

As a user, I want to connect with a community so that I can share my progress and stay motivated.

Acceptance criteria -

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.', 'The app facilitates participation in community challenges.', 'Users can offer and receive support and encouragement.', 'Community activity and engagement are tracked.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Community Support Feature

Test Case Creation Template:

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

Test Case 1

| FIT-CS-001 | Verify that users can join and participate in community groups. | This test case verifies that users can successfully join existing community groups and engage in group activities. | Community Support | High | - User account created and logged in<br>- Active internet connection<br>- Community groups exist within the app | No test data needed | 1. Navigate to the Community section.<br>2. Select a community group from the available list.<br>3. Tap the "Join Group" button.<br>4. Verify that the user is added to the group

and can access group content. | - User remains a member of the joined group.<br>- Activity within the group is reflected in the user's profile.<br>- App server logs the user's group interaction. | User successfully joins the selected community group and can access group features. | Blocker | High | Functional, Regression, Acceptance | Positive |

Test Case 2

| FIT-CS-002 | Verify that users can share progress updates and achievements. | This test case verifies the functionality allowing users to share their progress updates, achievements, and milestones within a community group. | Community Support | High | - User account created and logged in<br>- User has joined at least one community group<br>- User has recorded fitness data or achievements to share | No test data needed | 1. Navigate to the chosen community group.<br>2. Select the option to share an update.<br>3. Input the desired progress information or achievement details.<br>4. Optionally, attach relevant media (photo, screenshot).<br>5. Submit the update to the group. | - The shared update is visible to other group members.<br>- The update is stored and displayed chronologically within the group's activity feed.<br>- App server logs the update and its associated information. | Users can successfully share their progress updates and achievements with the community group. | Critical | High | Functional, Regression, Acceptance | Positive |

Test Case 3

| FIT-CS-003 | Verify that the app facilitates participation in community challenges. | This test case verifies the app's functionality for users to join, participate in, and track progress within community challenges. | Community Support | High | - User account created and logged in<br>- Active community challenges are available<br>- User has joined at least one community group | No test data needed | 1. Navigate to the Community Challenges section.<br>2. Browse and select a challenge to join.<br>3. Join the challenge and start recording relevant activity data.<br>4. Verify that the app tracks and displays the user's progress within

the challenge. | - The user's progress is updated in real-time as they record activity data.<br>- Challenge leaderboards accurately reflect participants' progress.<br>- App server correctly records and updates challenge data. | Users can successfully join, participate in, and track their progress within community challenges. | Major | High | Functional, Regression, Acceptance | Positive |

Test Case 4

| FIT-CS-004 | Verify that users can offer and receive support and encouragement. | This test case validates the functionality for users to interact with each other by offering and receiving support, encouragement, and feedback within community groups. | Community Support | Medium | - User account created and logged in<br>- User is a member of at least one community group with active members<br>- Community group has posts or updates available for interaction | No test data needed | 1. Navigate to a community group.<br>2. Locate a post or update where a user can offer support or encouragement.<br>3. Leave a comment or utilize reaction features (like, upvote, etc.).<br>4. Verify that the interaction is recorded and displayed correctly within the group. | - Comments and interactions are visible to other group members.<br>- Users receive notifications for interactions directed at their posts or activities.<br>- App server correctly processes and stores user interactions. | Users can successfully offer and receive support and encouragement through comments, reactions, and other interactive features within community groups. | Minor | Medium | Functional, Regression, Acceptance | Positive |

Test Case 5

| FIT-CS-005 | Verify that Community activity and engagement are tracked. | This test case ensures the app accurately tracks and displays various metrics related to community activity and engagement levels within the app. | Community Support | Medium | - User account created and logged in<br>- Users actively participating in community features<br>- Sufficient time has passed to generate trackable data | No test data needed | 1. Navigate to relevant sections displaying community activity (e.g., leaderboards, user profiles, group activity

feeds).<br>2. Verify that the app accurately displays metrics such as: <br> - Number of active users in a group.<br> - Total posts or comments within a specific timeframe.<br> - Participation rates in challenges.<br>3. Compare displayed metrics with backend data to ensure accuracy. | - Displayed metrics align with actual user activity and engagement data.<br>- Tracked data is used to personalize user experience (e.g., recommendations).<br>- App server correctly aggregates and stores community activity data. | The app accurately tracks and displays community activity and engagement metrics to provide insights into user participation and platform health. | Minor | Medium | Functional, Regression, Performance | Positive |

Data - 85

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Wearable Devices

Feature description -

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

Userstory title -

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

Acceptance criteria -

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Wearable Device Integration

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
FIT-WD-001	Verify that the app supports integration with multiple wearable devices.	This test case verifies that the app can successfully connect and communicate with a variety of popular wearable devices from different manufacturers.	Wearable Integration	High	1. Fitness Tracker App installed on the test device. 2. Bluetooth enabled on the test device. 3. A selection of different wearable devices (at least 3) from different manufacturers, compatible with the app.	List of supported wearable devices.	1. Open the Fitness Tracker app. 2. Navigate to the 'Settings' or 'Devices' section. 3. Attempt to pair the app with each of the selected wearable devices.	1. Disconnect the wearable devices.	The app should successfully pair with each wearable device without errors.	Blocker	High	Integration Testing	Positive
FIT-WD-002	Verify that users can sync activity data from their wearables.	This test case ensures that the app can successfully retrieve and display various activity data points from the connected wearable device.	Data Syncing	High	1. Fitness Tracker App installed and paired with a wearable device. 2. Wearable device has recorded activity data.	No test data needed.	1. Record activity data (e.g., steps, heart rate) on the wearable device. 2. Open the Fitness Tracker App. 3. Initiate a manual sync or wait for the automatic sync to occur. 4. Check the app's dashboard or relevant sections for the synced data.	1. None.	The app should display the synced activity data accurately, matching the data recorded on the wearable device.	Critical

High | Integration Testing, Functional Testing | Positive |

| FIT-WD-003 | Verify that the app provides real-time data updates from wearables. | This test case validates the app's ability to receive and display activity data from the connected wearable device in real-time. | Real-time Updates | High | 1. Fitness Tracker App installed and paired with a wearable device. 2. Wearable device is actively tracking data. | No test data needed. | 1. Wear the wearable device and start an activity (e.g., walking, running). 2. Open the Fitness Tracker app and navigate to the real-time tracking screen. 3. Observe the data displayed on the app. 4. Compare the data displayed on the app with the data displayed on the wearable device. | 1. Stop the activity on the wearable device. | The app should display data updates from the wearable device with minimal latency, reflecting changes in activity level in real time. | Major | High | Integration Testing, Functional Testing, Performance Testing | Positive |

| FIT-WD-004 | Verify that wearable data includes steps, heart rate, and calories burned. | This test case ensures that the app correctly receives and displays specific data points - steps taken, heart rate, and calories burned - from the connected wearable. | Data Accuracy | High | 1. Fitness Tracker App installed and paired with a wearable device. 2. Wearable device has recorded activity data including steps, heart rate, and calories burned. | No test data needed. | 1. Record activity data including steps, heart rate, and calories burned on the wearable device. 2. Sync the data with the Fitness Tracker App. 3. Verify that the app displays the correct data for each metric. | 1. None. | The app should accurately display the steps, heart rate, and calories burned data received from the wearable device. | Critical | High | Integration Testing, Functional Testing | Positive |

| FIT-WD-005 | Verify that users can troubleshoot connectivity issues with wearables. | This test case assesses the availability and effectiveness of troubleshooting options within the app to address common connectivity problems with wearable devices. | Connectivity Troubleshooting | Medium | 1. Fitness Tracker App installed. | No test data needed. | 1. Simulate a connectivity issue (e.g., disable Bluetooth, move out of range). 2. Attempt to sync data or access wearable features in the app. 3. Check for in-app guidance or troubleshooting steps provided to the user. 4. Follow the provided steps to resolve the simulated issue. | 1. Re-enable Bluetooth or reconnect the device. | The app should provide clear error messages and helpful troubleshooting steps to guide users in resolving connectivity issues with their wearables. | Major | Medium | Functional Testing, Usability Testing | Negative |

Data - 86

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Water Intake Tracking

Feature description -

Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

Userstory title -

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

Acceptance criteria -

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Water Intake Tracking Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
WIT-001	Verify that the app allows users to log their daily water intake.	This test case verifies the functionality of logging water intake and ensures that the app accurately records the input.	Water Intake Tracking	High	- User account created and logged in. - Water Intake feature accessible.	- Water intake amount: 500 ml	1. Navigate to the Water Intake feature. 2. Select the option to add water intake. 3. Enter "500 ml" as the water intake amount. 4. Save the logged intake.	- Verify the logged intake is displayed in the daily log.	Water intake of 500 ml should be logged and displayed accurately.	Critical	High	Functional, Regression	Positive
WIT-002	Verify that the app calculates total water consumption for the day.	This test case validates the app's ability to sum up all logged water intakes and display the total daily consumption to the user.	Water Intake Tracking	High	- User account created and logged in. - At least two water intake entries logged for the day.	- Water intake 1: 300 ml - Water intake 2: 450 ml	1. Navigate to the Water Intake feature. 2. Observe the total water consumption displayed for the current day.	No specific postconditions.	The app should correctly calculate and display the total water consumption for the day (750 ml).	Critical	High	Functional, Regression	Positive
WIT-003	Verify that users can set daily hydration goals.	This test case ensures users can set personalized daily hydration goals within the app.	Water Intake Tracking	High	- User account created and logged in. - Water Intake feature accessible.	- Hydration goal: 2000 ml	1. Navigate to the Water Intake feature settings. 2. Locate the option to set a daily hydration goal. 3. Input "2000 ml" as the desired goal. 4. Save the entered hydration goal.	- Verify the set goal is displayed within the Water Intake feature.	The app should allow setting a daily hydration goal and reflect it within the user interface.	Major	High	Functional, Regression	Positive
WIT-004	Verify that reminders are provided to encourage regular water intake.	This test case validates the reminder functionality for promoting regular water consumption based on the user's set preferences.	Water

Intake Tracking | High | - User account created and logged in. - Hydration goal set and reminders enabled. - Reminder frequency: Every 2 hours | - No specific test data needed. | 1. Configure the reminder settings with desired frequency (e.g., every 2 hours). 2. Wait for the reminder notification to appear. | No specific postconditions. | The app should deliver timely reminders to encourage water intake based on the configured frequency. | Minor | Medium | Functional, Usability | Positive |
| WIT-005 | Verify that users can view hydration progress and trends. | This test case ensures users can access and understand their hydration progress and trends over time. | Water Intake Tracking | Medium | - User account created and logged in. - Water intake data logged for at least one week. | - No specific test data needed. | 1. Navigate to the Water Intake feature. 2. Locate the section displaying hydration progress and trends. 3. Observe the visual representation of hydration data (e.g., graphs, charts). | No specific postconditions. | The app should display clear and informative visualizations of hydration progress and trends based on the user's logged data. | Minor | Medium | Functional, Usability | Positive |

--

Data - 87

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Nutrition Tracking

Feature description -

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

Userstory title -

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

Acceptance criteria -

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Nutrition Tracking Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
NUT-001	Verify that users can log their daily meals and snacks.	This test case verifies the functionality of logging various meals and snacks throughout the day.	Nutrition Tracking	High	- User account created. - User logged into the app. - User has navigated to the nutrition tracking section.	- Meal type (Breakfast, Lunch, Dinner, Snack). - Food items (e.g., Oatmeal, Salad, Chicken Breast, Apple). - Serving size.	1. Select the appropriate meal type. 2. Search for the consumed food item. 3. Select the food item from the search results. 4. Input the serving size. 5. Save the meal entry.	- Meal is successfully logged and displayed in the food diary. - Nutritional information for the meal is calculated and displayed.	The user should be able to log meals and snacks with details.	Critical	High	Functional, Regression	Positive
NUT-002	Verify that the app calculates macronutrient intake (carbs, fats, proteins).	This test case validates the app's ability to accurately calculate and display macronutrient breakdown for logged meals.	Nutrition

Tracking | High | - User account created. - User logged into the app. - User has logged at least one meal. | - Meal with known macronutrient values. | 1. Navigate to the logged meal details. 2. Observe the displayed macronutrient breakdown (carbs, fats, proteins). | None | The app should accurately calculate and display the macronutrient breakdown for the logged meal, matching the known values. | Critical | High | Functional, Regression | Positive |

| NUT-003 | Verify that users can set nutritional goals and track progress. | This test case checks if users can define daily or weekly nutritional goals and monitor their progress towards those goals. | Nutrition Tracking | High | - User account created. - User logged into the app. - User has navigated to the nutrition goal settings. | - Target daily/weekly calorie intake. - Target macronutrient ratio (e.g., 40% carbs, 30% protein, 30% fat). | 1. Set desired calorie intake goal. 2. Set desired macronutrient ratio. 3. Save the nutritional goals. 4. Log meals and track progress towards set goals. | - Nutritional goals are saved and displayed. - App tracks daily/weekly progress and provides visual feedback (e.g., progress bars, charts). | Users should be able to set personalized nutritional goals and the app should accurately track their progress. | Major | High | Functional, Regression | Positive |

| NUT-004 | Verify that the app provides nutritional information and suggestions. | This test case ensures the app delivers relevant nutritional information and potentially offers personalized suggestions based on logged data. | Nutrition Tracking | Medium | - User account created. - User logged into the app. - User has logged meals for a significant duration (e.g., one week). | None | 1. Navigate to the nutrition insights section. 2. Observe if the app provides information related to logged meals, such as average daily intake, macronutrient distribution, etc. 3. Check for personalized suggestions or recommendations based on eating habits. | None | The app should provide insightful information about the user's nutritional intake and may offer relevant suggestions for improvement. | Minor | Medium | Functional | Positive |

| NUT-005 | Verify that users can view their nutritional history and trends. | This test case verifies the functionality of accessing past nutritional data and identifying potential trends in eating habits. | Nutrition Tracking | Medium | - User account created. - User logged into the app. - User has logged meals consistently for a specific period (e.g., one month). | None | 1. Navigate to the nutrition history section. 2. Select a specific timeframe to view data (e.g., past week, past month). 3. Observe the displayed historical data, including

logged meals, macronutrient intake, and potentially visualized trends. | None | Users should be able to easily access and review their past nutritional data for any chosen timeframe. | Minor | Medium | Functional | Positive |

Data - 88

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Progress Reports

Feature description -

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

Userstory title -

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

Acceptance criteria -

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

Generated Test Cases:

## Fitness Tracker App - Progress Reports Test Cases

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
PR-001	Verify that users receive weekly and monthly progress reports	This test case verifies that the application automatically generates and delivers both weekly and monthly progress reports to the user.	Progress Reports	High	- User account created. <br> - User has logged at least one week of activity data. <br> - Notifications are enabled for the app.	No test data needed	1. Log in to the app as a user with activity data. <br> 2. Navigate to the progress reports section. <br> 3. Verify that a weekly progress report is available for the current week. <br> 4. Wait until the start of the next week and repeat step 3. <br> 5. Verify that a monthly progress report is available at the beginning of the following month.	None	Weekly and Monthly progress reports are generated and displayed correctly.	Critical	High	Functional Testing, Regression Testing	Positive
PR-002	Verify that progress reports include activity levels, workout performance, and goal attainment	This test case verifies that the progress reports accurately display information related to the user's activity levels, workout performance, and progress towards their set goals.	Progress Reports	High	- User account created. <br> - User has logged activity data, including workouts and goals.	No test data needed	1. Log in to the app and navigate to the progress reports section. <br> 2. Open a weekly or monthly progress report. <br> 3. Verify that the report includes sections for activity levels (e.g., steps, calories burned). <br> 4. Verify that the report displays information on workout performance (e.g., duration, average heart rate). <br> 5. Verify that the report shows progress towards any active goals set by the user.	None	Progress reports accurately present information on activity levels, workout performance, and goal attainment.	Critical	High	Functional

Testing, Regression Testing | Positive |

| PR-003 | Verify that users can view detailed breakdowns of achievements and trends | This test case checks if users can access detailed information and visualizations related to their achievements and trends within the progress reports. | Progress Reports | Medium | - User account created.
 - User has several weeks of logged activity data. | No test data needed | 1. Log in to the app and navigate to the progress reports section.
 2. Open a progress report and locate an achievement or trend summary.
 3. Tap/click on the summary to view a detailed breakdown.
 4. Verify that the detailed view provides additional information, such as daily/weekly breakdowns, graphs, or comparisons to previous periods. | None | Users can access detailed information and visualizations for achievements and trends. | Major | Medium | Functional Testing, Usability Testing | Positive |

| PR-004 | Verify that the app provides recommendations based on report findings | This test case validates that the app analyzes the user's progress data and presents relevant and personalized recommendations for improvement. | Progress Reports | Medium | - User account created.
 - User has several weeks of logged activity data, including areas for potential improvement. | No test data needed | 1. Log in to the app and navigate to the progress reports section.
 2. Open a progress report.
 3. Verify that the app presents recommendations based on the report findings (e.g., increase activity on certain days, adjust workout intensity).
 4. Ensure the recommendations are relevant to the user's activity patterns and goals. | None | The app provides personalized and relevant recommendations based on user's progress data. | Minor | Medium | Functional Testing, Regression Testing | Positive |

| PR-005 | Verify that users can customize the frequency and content of reports | This test case ensures users have the ability to customize the frequency (e.g., weekly, bi-weekly, monthly) and content (e.g., choose data points to include) of their progress reports. | Progress Reports | Medium | - User account created. | No test data needed | 1. Log in to the app and navigate to settings.
 2. Locate the "Progress Reports" settings section.
 3. Verify that options exist to modify the report frequency and select specific data points for inclusion.
 4. Adjust the frequency and content settings and save the changes.
 5. Check subsequent progress reports to confirm the customizations have been applied correctly. | None | Users can successfully customize the frequency and content of their progress reports. | Minor | Low | Functional Testing, Usability Testing |

Positive |

Data - 89

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

In-App Challenges

Feature description -

The app offers in-app challenges and competitions, encouraging users to set personal records, compete with friends, and earn rewards for achieving fitness milestones. In-app challenges add an element of fun, motivation, and friendly competition to the fitness tracking experience.

Userstory title -

As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals.

Acceptance criteria -

['Users can join various in-app challenges.', 'Challenges have clear objectives and rules.', 'Users can track their progress within the challenge.', 'Rewards are provided for achieving challenge milestones.', 'Users can compete with friends and view leaderboards.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - In-App Challenges Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
TC_IAC_001	Verify that users can join various in-app challenges	This test case verifies that users can successfully join different types of in-app challenges offered within the app.	In-App Challenges	High	- User is logged into the app. - User navigates to the "Challenges" section.	No test data needed	1. Observe the list of available challenges. 2. Select a challenge from the list. 3. Tap the "Join Challenge" button.	- User is successfully enrolled in the selected challenge. - Challenge progress is reflected in the user's profile.	User should be able to join the selected challenge without any errors.	Critical	High	Functional, User Interface	Positive
TC_IAC_002	Verify that challenges have clear objectives and rules	This test case verifies that each in-app challenge clearly displays its objectives, rules, and any relevant information to the user.	In-App Challenges	High	- User is logged into the app. - User navigates to the "Challenges" section.	No test data needed	1. Select a challenge from the list. 2. Observe the challenge details screen. 3. Verify that the challenge objectives, rules, duration, and other relevant information are clearly presented.	N/A	Challenge details screen should clearly present all necessary information to the user.	Critical	High	Functional, User Interface	Positive
TC_IAC_003	Verify that users can track their progress within the challenge	This test case verifies that users can easily track their progress towards the challenge goals within the app.	In-App Challenges	High	- User is logged into the app. - User has joined an active challenge.	No test data needed	1. Navigate to the "Challenges" section. 2. Select the active challenge. 3. Verify that the current progress towards the challenge goal is clearly displayed. 4. Perform an activity that contributes to the challenge goal (e.g., record a workout). 5. Verify that the progress is updated accordingly.	N/A	The app should accurately and clearly display the user's progress within the challenge.	Major	High	Functional, User Interface	Positive

| TC_IAC_004 | Verify that rewards are provided for achieving challenge milestones | This test case checks if users receive appropriate rewards upon reaching specific milestones within a challenge. | In-App Challenges | High | - User is logged into the app. - User has joined an active challenge. - User has achieved a defined milestone within the challenge. | No test data needed | 1. Navigate to the "Challenges" section. 2. Select the active challenge. 3. Observe if any notification or indication of reward achievement is presented upon reaching the milestone. 4. Check the user's profile for any awarded badges, points, or other rewards. | N/A | The app should properly reward the user upon achieving challenge milestones as defined by the challenge rules. | Major | High | Functional, User Interface | Positive |
| TC_IAC_005 | Verify that users can compete with friends and view leaderboards | This test case verifies that users can participate in challenges with friends and view leaderboards to track their ranking among friends and other participants. | In-App Challenges | High | - User is logged into the app. - User has joined an active challenge that allows friend competition. | No test data needed | 1. Navigate to the "Challenges" section and select the active challenge. 2. Locate the leaderboard section, if available. 3. Verify that the leaderboard displays the user's current ranking. 4. Verify that the leaderboard displays the ranking of other participants, including friends. | N/A | The app should allow users to compete with friends within challenges and provide a clear and accurate leaderboard. | Major | High | Functional, User Interface, Integration | Positive |

Data - 90

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Customizable Dashboards

Feature description -

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

Userstory title -

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Acceptance criteria -

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

Generated Test Cases:

## Test Cases for Customizable Dashboards in Fitness Tracker App

Test Case ID: TC_CD_001

Test Case Title: Verify that users can select which metrics to display on their dashboard.

Test Case Description: This test case verifies the functionality of selecting and deselecting different fitness metrics for display on the user's customizable dashboard.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions:

- The user is logged into the app.
- The user has navigated to the dashboard customization screen.

Test Data: No test data needed.

Test Steps:

1. Observe the list of available fitness metrics for display.

2. Select a metric currently not displayed on the dashboard.

3. Verify that the selected metric is added to the dashboard preview.

4. Deselect a metric currently displayed on the dashboard.

5. Verify that the deselected metric is removed from the dashboard preview.

6. Repeat steps 2-5 for a variety of different metrics.

Postconditions:

- The dashboard preview accurately reflects the selected metrics.

Expected Result: Users should be able to seamlessly add or remove fitness metrics from their dashboard view.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TC_CD_002

Test Case Title: Verify that the app allows customization of dashboard layout and theme.

Test Case Description: This test case checks the user's ability to modify the dashboard's layout by rearranging displayed metrics and applying different visual themes.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions:

- The user is logged into the app.

- The user is on the dashboard customization screen.

Test Data: No test data needed.

Test Steps:

1. Access the layout customization options.

2. Attempt to rearrange the displayed metrics using drag-and-drop or a similar method.

3. Verify that the dashboard preview updates in real-time to reflect the new arrangement.

4. Access the theme customization options.

5. Select different themes from the available choices.

6. Verify that the dashboard preview updates in real-time to reflect the chosen theme.

Postconditions:

- The dashboard preview reflects the user's chosen layout and theme.

Expected Result: The app should provide intuitive controls for modifying the layout and theme of the dashboard, with real-time visual feedback.

Severity: Major

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: TC_CD_003

Test Case Title: Verify that users can save and update their dashboard settings.

Test Case Description: This test case ensures that any changes made to the dashboard customization are saved and persist across different sessions.

Test Suite: Dashboard Customization

Test Priority: High

Preconditions:

- The user is logged into the app.

- The user has made changes to their dashboard settings (e.g., selected metrics, layout, theme).

Test Data: No test data needed.

Test Steps:

1. Save the current dashboard configuration.

2. Navigate away from the dashboard customization screen (e.g., to another section of the app).

3. Return to the dashboard view.

4. Verify that the previously saved dashboard configuration is loaded correctly.

5. Make additional modifications to the dashboard settings.

6. Save the updated configuration.

7. Repeat steps 2-4 to confirm that the latest changes are saved and persistent.

Postconditions:

- The user's customized dashboard configuration is saved and reloaded successfully.

Expected Result: The app should save and load the user's personalized dashboard settings reliably across multiple sessions.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TC_CD_004

Test Case Title: Verify that dashboard displays are dynamic and update in real-time.

Test Case Description: This test case validates that the displayed metrics on the dashboard update dynamically as the user engages in activities tracked by the app.

Test Suite: Dashboard Functionality

Test Priority: High

Preconditions:

- The user is logged into the app.

- The user is actively using the app to track a workout or physical activity.

Test Data: Simulated workout data (e.g., steps taken, calories burned).

Test Steps:

1. Initiate a simulated workout or input workout data manually.

2. Observe the corresponding metrics displayed on the dashboard.
3. Verify that the displayed metrics update in real-time to reflect the ongoing activity data.
4. Continue the simulated workout or input additional data.
5. Confirm that the dashboard display continues to update dynamically and accurately.

Postconditions: None.

Expected Result: The dashboard should dynamically reflect the most up-to-date fitness data, providing real-time feedback to the user during and after activities.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Performance Testing

Test Case Approach: Positive

Test Case ID: TC_CD_005

Test Case Title: Verify that users can reset the dashboard to default settings.

Test Case Description: This test case confirms the functionality for users to revert their dashboard back to its original default configuration.

Test Suite: Dashboard Customization

Test Priority: Medium

Preconditions:

- The user is logged into the app.
- The user has customized their dashboard settings.

Test Data: No test data needed.

Test Steps:

1. Access the dashboard customization settings.
2. Locate the option to "Reset to Default" or similar.
3. Trigger the reset functionality.
4. Confirm with the user if they wish to proceed with resetting the dashboard (confirmation dialog).

5. Verify that the dashboard configuration reverts to the default settings.

Postconditions:

- The dashboard is reset to its original default state.

Expected Result: The app should allow users to easily restore their dashboard to the default configuration if desired.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

--

Data - 91

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Social Sharing

Feature description -

Users can share workout summaries, achievements, and progress updates on social media platforms, inspiring others and fostering accountability. Social sharing enables users to celebrate their successes, seek support from their social networks, and stay motivated on their fitness journey.

Userstory title -

As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

Acceptance criteria -

['Users can share workout summaries and achievements.', 'The app supports sharing on multiple social media platforms.', 'Users can customize the content and format of shared posts.', 'Sharing features are integrated seamlessly with social media.', 'Users can track the engagement of their shared posts.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Social Sharing Feature

Test Case ID: FT-SS-001

Test Case Title: Verify that users can share workout summaries and achievements.

Test Case Description: This test case verifies the functionality of sharing workout summaries and achievements on social media platforms. It ensures that users can successfully share their desired information after completing a workout or achieving a fitness goal.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

* The user is logged in to the Fitness Tracker App.
* The user has completed a workout or achieved a fitness goal.

Test Data: Sample workout summary with achievement data.

Test Steps:

1. Navigate to the workout summary or achievement section.
2. Select the "Share" option.
3. Choose the desired social media platform.
4. Verify that the app redirects to the selected platform with pre-populated content including the workout summary/achievement details.

5. Post the content on the social media platform.

Postconditions:

* Verify that the shared post appears on the user's social media feed.

* Verify that the shared content accurately reflects the selected workout summary or achievement.

Expected Result: The workout summary and achievement details are successfully shared on the selected social media platform.

Severity: Major

Priority: High

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: FT-SS-002

Test Case Title: Verify that the app supports sharing on multiple social media platforms.

Test Case Description: This test verifies that users can share their fitness data on various social media platforms. This case ensures the integration of multiple social media platforms within the application.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

* The user is logged in to the Fitness Tracker App.

* The user has chosen to share a workout summary or achievement.

Test Data: No test data needed

Test Steps:

1. Initiate the sharing process for a workout summary or achievement.

2. Observe the list of available social media platforms for sharing.

3. Verify that the list includes popular platforms like Facebook, Instagram, Twitter, etc.

Postconditions: N/A

Expected Result: The app displays a list of multiple social media platforms for the user to choose from.

Severity: Major

Priority: High

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case ID: FT-SS-003

Test Case Title: Verify that users can customize the content and format of shared posts.

Test Case Description: This test case verifies the customization options available for users when sharing their fitness information. It ensures that users can edit the shared content and format it according to their preferences.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

* The user is logged in to the Fitness Tracker App.
* The user has initiated the sharing process for a workout summary or achievement.

Test Data: Sample customized message and formatting options.

Test Steps:

1. Attempt to edit the pre-populated content before sharing.
2. Check if the app allows adding a personal message.
3. Verify if users can format the text (bold, italics, emojis).
4. Check if the app provides options for adding images or GIFs.

Postconditions: N/A

Expected Result: The app allows users to customize the shared content with text edits, personal messages, formatting, and multimedia additions.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: FT-SS-004

Test Case Title: Verify that sharing features are integrated seamlessly with social media.

Test Case Description: This test case ensures smooth and efficient integration between the fitness tracker app and the chosen social media platforms during the sharing process.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

* The user is logged in to the Fitness Tracker App and has authorized access to their social media accounts.

* The user initiates the sharing process and selects a social media platform.

Test Data: No test data needed

Test Steps:

1. Initiate a share to a social media platform.

2. Verify that the app redirects to the correct platform's authorization page if not already authorized.

3. Upon successful authorization, confirm the app redirects back to the Fitness Tracker App.

4. Check that the shared post appears on the user's social media feed without errors.

Postconditions: N/A

Expected Result: The sharing process is completed without errors, redirecting users seamlessly between the app and the chosen social media platform.

Severity: Critical

Priority: High

Type of Testing: Integration, Functional

Test Case Approach: Positive

Test Case ID: FT-SS-005

Test Case Title: Verify that users can track the engagement of their shared posts.

Test Case Description: This test case checks if users can monitor the engagement (likes, comments, shares) their shared posts receive on social media platforms within the fitness tracker app.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

* The user has shared a post on social media from the Fitness Tracker App.

* The shared post has received engagement (likes, comments, shares) on the platform.

Test Data: No test data needed

Test Steps:

1. Navigate to a section within the app that should display social media engagement.

2. Verify if the app displays the number of likes, comments, or shares for the shared post.

3. Check for any discrepancies between the engagement data displayed on the app and the actual engagement on the social media platform.

Postconditions: N/A

Expected Result: The app accurately reflects the engagement (likes, comments, shares) the shared post received on the social media platform.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Integration

Test Case Approach: Positive

Data - 92

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Rest and Recovery Tracking

Feature description -

The app includes features for tracking rest days, recovery activities, and recovery metrics such as resting heart rate and sleep quality. Rest and recovery tracking helps users prioritize recovery, prevent overtraining, and optimize their overall health and performance.

Userstory title -

As a user, I want to track my rest and recovery so that I can optimize my fitness performance and prevent overtraining.

Acceptance criteria -

['Users can log rest days and recovery activities.', 'The app tracks metrics such as resting heart rate and sleep quality.', 'Users receive recommendations for recovery based on their activities.', 'The app provides alerts for insufficient recovery time.', 'Users can view trends and history of their recovery data.']

Generated Test Cases:

## Fitness Tracker App - Rest and Recovery Tracking Test Cases

Test Case ID: FR-001

Test Case Title: Verify that users can log rest days and recovery activities.

Test Case Description: This test case verifies the functionality of the app to allow users to input and save rest days and various recovery activities with details such as date, time, and activity type.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- The app is installed and launched.

- The user is logged in.

Test Data:

- Rest day: Current Date

- Recovery activity: Yoga, 30 minutes, Current Date

Test Steps:

1. Navigate to the "Rest and Recovery" section.

2. Select "Log Rest Day" and input the current date.

3. Save the rest day entry.

4. Select "Log Recovery Activity" and choose "Yoga" from the list of activities.

5. Input the duration as "30 minutes" and select the current date.

6. Save the recovery activity entry.

Postconditions:

- The rest day is logged and visible in the user's history.

- The recovery activity is logged with the correct details in the user's history.

Expected Result: The app successfully logs and displays the user's rest day and recovery activity information.

Severity: Critical

Priority: High

Type of Testing: Functional Testing, User Interface Testing

Test Case Approach: Positive

Test Case ID: FR-002

Test Case Title: Verify that the app tracks metrics such as resting heart rate and sleep quality.

Test Case Description: This test case verifies the app's ability to receive, process, and display data from integrated devices or user input regarding resting heart rate and sleep quality.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- The app is installed and launched.

- The user is logged in.

- The user has granted permission for data access from a wearable device or has manually inputted the data.

Test Data:

- Resting heart rate: 65 bpm

- Sleep quality: 8 hours, "Good"

Test Steps:

1. Sync data from the integrated wearable device or manually input the data.

2. Navigate to the "Rest and Recovery" section.

3. Check the displayed resting heart rate.

4. Check the displayed sleep quality data.

Postconditions:

- The app displays the resting heart rate and sleep quality data accurately.

Expected Result: The app accurately reflects the resting heart rate and sleep quality data received.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: FR-003

Test Case Title: Verify that users receive recommendations for recovery based on their activities.

Test Case Description: This test case verifies that the app analyzes user activity data and provides relevant recovery recommendations, such as stretching, hydration, or specific recovery activities.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- The app is installed and launched.
- The user is logged in.
- The user has logged multiple workout sessions with varying intensity levels.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Rest and Recovery" or "Recommendations" section.
2. Check if the app provides personalized recommendations based on recent workout intensity and frequency.
3. Log a high-intensity workout session.
4. Check if the app adjusts the recommendations to prioritize recovery based on the high-intensity session.

Postconditions:

- The recommendations section displays personalized advice for recovery.

Expected Result: The app provides personalized and dynamic recovery recommendations based on user activity data.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Data-Driven Testing

Test Case Approach: Positive

Test Case ID: FR-004

Test Case Title: Verify that the app provides alerts for insufficient recovery time.

Test Case Description: This test case verifies the app's functionality to analyze user data, including logged workouts and rest, and generate alerts when insufficient recovery time is detected.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- The app is installed and launched.

- The user is logged in.

- The user has enabled notifications for recovery alerts.

Test Data:

- Two consecutive days of high-intensity workouts logged.

Test Steps:

1. Log a high-intensity workout on two consecutive days without sufficient rest logged in between.

2. Check if the app generates a notification or alert about insufficient recovery time.

3. Dismiss the alert and navigate to the "Rest and Recovery" section.

4. Verify if the app recommends rest or low-intensity activities.

Postconditions:

- The user receives an alert for insufficient recovery time.

Expected Result: The app successfully identifies patterns of insufficient recovery and provides timely alerts to the user.

Severity: Major

Priority: High

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Negative

Test Case ID: FR-005

Test Case Title: Verify that users can view trends and history of their recovery data.

Test Case Description: This test case verifies the app's ability to store, process, and display historical data related to rest, recovery activities, resting heart rate, and sleep quality for user analysis.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- The app is installed and launched.
- The user is logged in.
- The user has logged several weeks of rest, recovery activities, and relevant metrics.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Rest and Recovery" section.
2. Access the "History" or "Trends" view within the section.
3. Verify that the app displays historical data for rest days, recovery activities, resting heart rate, and sleep quality.
4. Check if the data is presented in an easily understandable format, such as graphs or charts, showing trends over time.

Postconditions:

- The user can view a comprehensive history of their recovery data.

Expected Result: The app displays a user-friendly and informative overview of historical recovery data, allowing users to track their progress and identify patterns.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing, Usability 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 Cases for Fitness Tracker App - Audio Coaching Feature

Test Case Creation Template:

Field	Description
Test Case ID	(Provide a unique identifier for each test case)
Test Case Title	(Must start with "Verify that..." and clearly state the feature/function being tested)
Test Case Description	(A brief description of the test case, between 70-100 words)
Test Suite	(Name of the test suite or module)
Test Priority	(Priority level: High, Medium, Low)
Preconditions	(List any prerequisites before executing the test case, max 3 items, in an unordered list)
Test Data	(Specify data required for execution; if none, state "No test data needed")
Test Steps	(List steps for executing the test case, min 3 to max 6)
Postconditions	(Any follow-up actions after test execution, max 3 items, in an unordered list)
Expected Result	(Overall expected outcome of the test case)
Severity	(Blocker, Critical, Major, Minor, Trivial)
Priority	(High, Medium, Low)
Type of Testing	(Functional, Integration, System, etc.)
Test Case Approach	(Positive, Negative, Destructive)

Test Case 1:

Field	Value
Test Case ID	FT-AC-001
Test Case Title	Verify that users can access a variety of audio-guided workouts.
Test Case Description	This test case verifies that the app offers a diverse selection of audio-guided workouts

across different workout types, durations, and intensity levels. |

Test Suite	Audio Coaching
Test Priority	High
Preconditions	- User is logged in to the app. - User is on the Workout Library screen.
Test Data	No test data needed.
Test Steps	1. Navigate to the Audio Coaching section within the Workout Library. 2. Observe the list of available audio-guided workouts. 3. Verify that workouts are categorized by type (e.g., Cardio, Strength, Yoga). 4. Verify that workouts offer varying durations (e.g., 15 min, 30 min, 45 min). 5. Verify that workouts provide clear indications of intensity level (e.g., Beginner, Intermediate, Advanced).
Postconditions	None.
Expected Result	The app should display a diverse selection of audio-guided workouts, categorized by type, duration, and intensity level.
Severity	Major
Priority	High
Type of Testing	Functional
Test Case Approach	Positive

Test Case 2:

Field	Value
Test Case ID	FT-AC-002
Test Case Title	Verify that audio coaching includes motivational messages and coaching cues.
Test Case Description	This test case verifies the presence and quality of motivational messages and coaching cues during an audio-guided workout.
Test Suite	Audio Coaching
Test Priority	High

Preconditions	- User is logged in to the app. - User has started an audio-guided workout.
Test Data	No test data needed.
Test Steps	1. Start an audio-guided workout. 2. Listen for motivational messages throughout the workout duration. 3. Observe the timing and relevance of coaching cues related to exercise technique, pace, and breathing. 4. Verify that the tone and language of the coaching are encouraging and supportive.
Postconditions	None.
Expected Result	The audio coaching should deliver clear and timely motivational messages and coaching cues, enhancing the workout experience.
Severity	Major
Priority	High
Type of Testing	Functional
Test Case Approach	Positive

Test Case 3:

Field	Value
Test Case ID	FT-AC-003
Test Case Title	Verify that users can choose coaching based on workout type and intensity.
Test Case Description	This test case verifies that users can filter or select audio coaching options based on their chosen workout type and desired intensity level.
Test Suite	Audio Coaching
Test Priority	High
Preconditions	- User is logged in to the app. - User is on the Workout Selection screen.
Test Data	- Workout type: Yoga - Intensity level: Beginner
Test Steps	1. Select "Yoga" as the workout type. 2. Choose "Beginner" as the intensity level. 3.

Browse available audio coaching options.<br> 4. Verify that the displayed options align with the selected workout type and intensity level. |
Postconditions	None.
Expected Result	The app should present audio coaching options tailored to the user's selected workout type and intensity level.
Severity	Major
Priority	High
Type of Testing	Functional
Test Case Approach	Positive

Test Case 4:

Field	Value
Test Case ID	FT-AC-004
Test Case Title	Verify that audio coaching integrates seamlessly with other app features.
Test Case Description	This test case ensures that audio coaching functions smoothly alongside other app features, such as workout tracking, music playback, and notifications.
Test Suite	Audio Coaching
Test Priority	High
Preconditions	- User is logged in to the app.<br> - User has started an audio-guided workout.<br> - User has background music playing.
Test Data	No test data needed.
Test Steps	1. Start an audio-guided workout.<br> 2. Begin playing music from the app's music player or a connected device.<br> 3. Verify that audio coaching volume automatically adjusts to ensure clear audibility over the music.<br> 4. Simulate receiving a phone call or notification.<br> 5. Verify that audio coaching pauses during the interruption and resumes seamlessly afterward.

Postconditions	None.
Expected Result	The audio coaching feature should seamlessly integrate with other app features, providing a smooth and enjoyable user experience.
Severity	Critical
Priority	High
Type of Testing	Integration
Test Case Approach	Positive

Test Case 5:

Field	Value
Test Case ID	FT-AC-005
Test Case Title	Verify that users can control audio settings and preferences.
Test Case Description	This test case confirms that users have control over audio settings related to volume, voice guidance, and sound effects during audio-guided workouts.
Test Suite	Audio Coaching
Test Priority	Medium
Preconditions	- User is logged in to the app. - User is on the App Settings screen.
Test Data	No test data needed.
Test Steps	1. Navigate to the audio settings section within the app settings. 2. Verify the presence of volume control for audio coaching. 3. Check for options to enable/disable voice guidance and sound effects. 4. Adjust the volume and toggle settings to confirm their functionality during a workout.
Postconditions	None.
Expected Result	Users should have the ability to customize their audio settings and preferences to personalize their audio coaching experience.
Severity	Minor

| Priority | Medium |

| Type of Testing | Functional |

| Test Case Approach | Positive |

Data - 94

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Health Apps

Feature description -

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

Userstory title -

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

Acceptance criteria -

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

Generated Test Cases:

## Test Cases for Fitness Tracker App: Integration with Health Apps

Test Case ID: FTA-IHA-001

Test Case Title: Verify that the fitness tracker app integrates with popular health and wellness apps.

Test Case Description: This test case verifies the app's ability to connect with a variety of popular health apps, ensuring users can integrate data from their preferred platforms.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- The fitness tracker app is installed on the test device.

- A stable internet connection is available.

Test Data: List of popular health and wellness apps (e.g., MyFitnessPal, Headspace, Calm)

Test Steps:

1. Open the fitness tracker app.

2. Navigate to the app's settings or integration section.

3. Attempt to connect with each app listed in the test data.

4. Observe if the connection process initiates successfully.

Postconditions:

- Disconnect from the integrated apps.

Expected Result: The app should successfully initiate the connection process with each of the listed popular health and wellness apps.

Severity: Blocker

Priority: High

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: FTA-IHA-002

Test Case Title: Verify that users can sync data from connected health apps.

Test Case Description: This test verifies that once connected, the app successfully pulls and displays relevant data from the integrated health apps.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- The fitness tracker app is connected to at least one health app (e.g., MyFitnessPal).
- Data exists within the connected health app (e.g., calorie intake).

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app.
2. Navigate to the section where integrated data is displayed.
3. Initiate a data sync if not automatically performed.
4. Observe if data from the connected app (e.g., calorie intake) is displayed.

Postconditions: None

Expected Result: The app should successfully sync and display the relevant data from the connected health app.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, Functional Testing

Test Case Approach: Positive

Test Case ID: FTA-IHA-003

Test Case Title: Verify that the app provides a consolidated view of health metrics from different apps.

Test Case Description: This test validates the app's ability to aggregate and display data from multiple sources in a clear and understandable format.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- The fitness tracker app is connected to at least two different health apps (e.g., MyFitnessPal, Headspace).
- Data exists within both connected apps.

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app.
2. Navigate to the main dashboard or health overview section.
3. Observe if data from both connected apps are displayed together in a consolidated manner.

Postconditions: None

Expected Result: The app should present a comprehensive view, integrating data from different apps without redundancy or confusion.

Severity: Critical

Priority: High

Type of Testing: Integration Testing, UI Testing

Test Case Approach: Positive

Test Case ID: FTA-IHA-004

Test Case Title: Verify that users receive insights based on combined data from multiple apps.

Test Case Description: This test case ensures that the app leverages the combined data to provide users with personalized insights or recommendations.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- The fitness tracker app has been connected to multiple health apps and synced data for a reasonable period.

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app.
2. Navigate to the insights, recommendations, or a similar section.
3. Observe if the app generates insights that consider data from multiple integrated apps.

Postconditions: None

Expected Result: The app should generate relevant insights or recommendations based on the combined data, demonstrating value in integration.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing, Data Analysis

Test Case Approach: Positive

Test Case ID: FTA-IHA-005

Test Case Title: Verify that integration settings are easy to manage and configure.

Test Case Description: This test case examines the user-friendliness of the integration settings, ensuring users can easily connect, disconnect, and manage permissions for connected apps.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions: The fitness tracker app is installed on the test device.

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app.
2. Navigate to the app's settings or integration section.
3. Attempt to connect a new health app, modify permissions of an existing connection, and disconnect an app.
4. Observe the clarity of instructions, ease of navigation, and overall user experience within the integration settings.

Postconditions: None

Expected Result: Users should find the process of managing integration settings intuitive and straightforward.

Severity: Minor

Priority: Medium

Type of Testing: Usability Testing

Test Case Approach: Positive

Data - 95

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Body Composition Tracking

Feature description -

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and the effectiveness of exercise and nutrition interventions.

Userstory title -

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed decisions about my training and nutrition.

Acceptance criteria -

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set goals for body composition metrics.']

Generated Test Cases:

## Test Cases for Body Composition Tracking Feature

Test Case ID: TC_BCT_001

Test Case Title: Verify that users can log body composition metrics.

Test Case Description: This test case verifies that the app allows users to successfully enter and save body composition metrics, including weight, body fat percentage, and muscle mass. It tests various input values and ensures data is stored correctly.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged into the Fitness Tracker App.

- User has navigated to the Body Composition section.

Test Data:

- Valid weight values (e.g., 50kg, 180lbs).

- Valid body fat percentage values (e.g., 10%, 35%).

- Valid muscle mass values (e.g., 40kg, 90lbs).

Test Steps:

1. Open the "Add Body Composition" screen.

2. Enter a valid weight value in the designated field.

3. Enter a valid body fat percentage value in the designated field.

4. Enter a valid muscle mass value in the designated field.

5. Save the entered data.

6. Verify that the entered data is displayed correctly in the body composition history.

Postconditions:

- Logged data is saved and accessible for future reference.

Expected Result: The app should successfully save the entered body composition data and display it correctly in the user's history.

Severity: Critical

Priority: High

Type of Testing: Functional, Regression

Test Case Approach: Positive

Test Case ID: TC_BCT_002

Test Case Title: Verify that the app calculates and displays BMI based on logged data.

Test Case Description: This test case verifies the app's ability to accurately calculate and display the user's BMI (Body Mass Index) based on their entered height and weight data. It also checks the BMI classification based on calculated values.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:
- User is logged into the app.
- User has entered their height in the profile settings.

Test Data:
- Weight: 70 kg
- Height: 175 cm

Test Steps:

1. Navigate to the Body Composition section.

2. Log a new weight entry (e.g., 70 kg).

3. Navigate to the BMI section.

4. Verify that the app displays the calculated BMI value based on entered weight and height.

5. Verify that the app displays the correct BMI classification (e.g., Normal, Overweight) based on the calculated value.

Postconditions:

- Calculated BMI is displayed alongside other body composition data.

Expected Result: The app should accurately calculate and display the user's BMI and classify it according to standard BMI ranges.

Severity: Major

Priority: High

Type of Testing: Functional, Calculation

Test Case Approach: Positive

Test Case ID: TC_BCT_003

Test Case Title: Verify that users can view trends and history of body composition changes.

Test Case Description: This test case ensures users can easily access and view their historical body composition data, presented in an understandable format like graphs or charts to visualize trends and track progress over time.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:
- User is logged in.
- User has logged at least three body composition entries over different dates.

Test Data: No specific test data needed.

Test Steps:
1. Navigate to the Body Composition section.
2. Access the "History" or "Trends" view.
3. Verify that the screen displays logged body composition data (weight, body fat %, muscle mass) over time.
4. Check if the data is presented visually through graphs or charts.
5. Verify that users can adjust the timeframe for viewing historical data (e.g., last week, last month, all-time).

Postconditions: No specific postconditions.

Expected Result: The app should display the user's historical body composition data clearly and allow them to see trends and changes over different time periods.

Severity: Major

Priority: High

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: TC_BCT_004

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 app analyzes logged body composition data and provides personalized insights and recommendations to the user. These recommendations could be related to nutrition, exercise, or achieving their fitness goals.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:
- User is logged in.
- User has logged body composition data for a significant period (e.g., at least two weeks).

Test Data: No specific test data needed.

Test Steps:

1. Navigate to the Body Composition section.
2. Look for a dedicated area where insights and recommendations are displayed.
3. Verify that the app provides insights related to the user's body composition trends (e.g., "You've lost 2% body fat in the last month!").
4. Check if the app offers personalized recommendations based on the logged data and user's goals (e.g., "Consider increasing protein intake for muscle growth.").

Postconditions: No specific postconditions.

Expected Result: The app should analyze the user's data and provide relevant insights and personalized recommendations to improve their fitness journey.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case ID: TC_BCT_005

Test Case Title: Verify that users can set goals for body composition metrics.

Test Case Description: This test case verifies that users can set specific and measurable goals for their body composition metrics like weight, body fat percentage, or muscle mass. This feature allows users to work towards their desired physique.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:
- User is logged in.
- User has accessed the Body Composition section.

Test Data:
- Target Weight: 75 kg
- Target Body Fat Percentage: 18%

Test Steps:
1. Locate the "Set Goals" or similar option within the Body Composition section.
2. Choose a specific metric to set a goal for (e.g., weight).
3. Enter the desired target value for the chosen metric.
4. Set a timeframe for achieving the goal (optional).
5. Save the set goal.
6. Verify that the app displays the set goal clearly within the Body Composition section.

Postconditions:
- Set goals are stored and can be tracked by the user.

Expected Result: The app should allow users to set, save, and track their desired goals for different body composition metrics.

Severity: Major

Priority: High

Type of Testing: Functional, Usability

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 Cases for Fitness Tracker App - Workout Analysis Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
WA-001	Verify that users can view detailed performance metrics for each workout	This test case verifies that the app displays accurate and detailed performance metrics for each completed workout.	Workout Analysis	High	- User account created. - Workout logged and completed. - Workout data synced with the app.	No test data needed	1. Open the Fitness Tracker app. 2. Navigate to the "Workout History" section. 3. Select a completed workout from the list. 4. Verify that the app displays detailed metrics such as duration, distance, pace, heart rate (if applicable), calories burned, and any other relevant metrics specific to the workout type.	None	The app should accurately display all relevant workout metrics for the selected workout.	Critical	High	Functional, Regression	Positive
WA-002	Verify that the app provides trends and insights into workout intensity and duration over time	This test case verifies that the app analyzes workout data and presents trends and insights related to workout

intensity and duration. | Workout Analysis | High | - User account created. - Multiple workouts logged over a period of time (at least 1 week). - Workout data synced with the app. | No test data needed | 1. Open the Fitness Tracker app. 2. Navigate to the "Workout Analysis" or "Insights" section. 3. Select the "Intensity & Duration" analysis view (or similar). 4. Verify that the app displays trends in workout intensity and duration over the selected time period (e.g., weekly, monthly). 5. Verify that the app provides insights based on the analyzed data, such as changes in average intensity, workout frequency, and total workout duration. | None | The app should accurately display trends in workout intensity and duration and provide meaningful insights based on the analyzed data. | Major | High | Functional, Regression | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| WA-003 | Verify that users can track calorie burn and other health metrics related to their workouts | This test case verifies that the app accurately calculates and displays calorie burn and other relevant health metrics associated with each workout. | Workout Analysis | High | - User account created. - Workout logged and completed. - Workout data synced with the app. | No test data needed | 1. Open the Fitness Tracker app. 2. Navigate to the "Workout History" section. 3. Select a completed workout from the list. 4. Verify that the app displays the estimated calorie burn for the selected workout. 5. Verify that the app displays other relevant health metrics if applicable, such as average heart rate, steps taken during the workout, and active time. | None | The app should accurately display estimated calorie burn and other relevant health metrics for the selected workout. | Major | High | Functional, Regression | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| WA-004 | Verify that the app offers personalized recommendations based on workout analysis data | This test case verifies that the app provides personalized recommendations to users based on their workout history and performance analysis. | Workout Analysis | Medium | - User account created. - Multiple workouts logged over a period of time (at least 1 week). - Workout data synced with the app. - User profile with fitness goals and preferences (optional). | No test data needed | 1. Open the Fitness Tracker app. 2. Navigate to the "Workout Analysis" or "Insights" section. 3. Verify that the app provides personalized recommendations based on workout data such as: - Suggestions to adjust workout frequency, intensity, or duration based on progress toward goals. - Recommendations for new workout types or routines to try based on user preferences and

fitness level. - Tips for improving performance in specific workout categories. | None | The app should provide relevant and personalized recommendations based on the user's workout history and analysis data. | Minor | Medium | Functional, Regression | Positive |

|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| WA-005 | Verify that users can compare their current workout data with past performance for similar workouts | This test case verifies that the app allows users to compare their current workout data with past performance for similar workout types. | Workout Analysis | Medium | - User account created. - Multiple workouts of the same type logged over a period of time. - Workout data synced with the app. | No test data needed | 1. Open the Fitness Tracker app. 2. Navigate to the "Workout History" section. 3. Select a completed workout from the list. 4. Verify that the app provides an option to compare the selected workout with past performances of the same workout type (e.g., "Compare with last 5 runs"). 5. Verify that the comparison view displays relevant data points for comparison, such as duration, distance, pace, calories burned, and other relevant metrics. | None | The app should allow users to easily compare their current workout data with past performances for similar workout types, highlighting improvements or areas for attention. | Minor | Medium | Functional, Regression | Positive |

Data - 97

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Weather Integration

Feature description -

Integrated weather forecasts and conditions allow users to plan outdoor workouts and activities based on current weather conditions and forecasts. Weather integration helps users make informed decisions, stay safe, and optimize their outdoor exercise routines according to weather conditions.

Userstory title -

As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

Acceptance criteria -

['The app provides current weather conditions and forecasts.', 'Users can view weather details for different locations.', 'Weather data is updated in real-time.', 'The app offers recommendations based on weather conditions.', 'Users can receive weather alerts for extreme conditions.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Weather Integration

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
FT-WI-001	Verify that the app displays current weather conditions	This test case verifies that the app correctly fetches and displays the current weather conditions for the user's location.	Weather Integration	High	- User has granted location permissions to the app. <br> - The app has an active internet connection. <br> - User is on the main dashboard or workout planning screen.	No test data needed	1. Open the Fitness Tracker app. <br> 2. Navigate to the screen displaying weather information. <br> 3. Observe the displayed weather data.	None	The app displays current temperature, weather condition (e.g., sunny, cloudy), humidity, wind speed, and relevant weather icons.	Blocker	High	Functional, Integration, Acceptance	Positive

| FT-WI-002 | Verify that the app displays weather forecasts for different locations | This test case verifies that users can search for and view weather forecasts for different locations, allowing them to plan workouts in various areas. | Weather Integration | High | - The app has an active internet connection.
 - User is on the screen displaying weather information. | Location: "New York City, USA" | 1. Open the Fitness Tracker app.
 2. Navigate to the weather section.
 3. Use the search bar to input "New York City, USA".
 4. Select the correct location from the search results.
 5. Observe the displayed weather forecast. | None | The app displays a weather forecast for the next few hours or days for New York City, including temperature, conditions, and precipitation chances. | Critical | High | Functional, Integration, Acceptance | Positive |

| FT-WI-003 | Verify that the weather data updates in real-time | This test case ensures that the weather information displayed in the app is current and updates automatically to reflect the latest conditions. | Weather Integration | Medium | - The app has an active internet connection.
 - User is on the screen displaying weather information. | No test data needed | 1. Open the Fitness Tracker app.
 2. Navigate to the weather section.
 3. Note the current time and the last updated timestamp for weather data.
 4. Wait for a few minutes.
 5. Observe the weather data and the last updated timestamp. | None | The weather data, including temperature, conditions, and other details, should update automatically within a reasonable timeframe, and the last updated timestamp should reflect the recent update. | Major | Medium | Functional, Integration, Performance | Positive |

| FT-WI-004 | Verify that the app provides recommendations based on weather conditions. | This test case validates that the app analyzes the current and forecasted weather conditions and provides relevant recommendations to users for planning outdoor workouts. | Weather Integration | Medium | - The app has an active internet connection.
 - User is on the workout planning screen with weather integration enabled. | Weather Condition: Rainy | 1. Open the Fitness Tracker app.
 2. Navigate to the workout planning screen.
 3. Simulate rainy weather conditions.
 4. Observe the recommendations provided by the app. | None | The app recommends indoor workout alternatives or suggests postponing the outdoor workout due to the rain. | Minor | Medium | Functional, Integration, Acceptance | Positive |

| FT-WI-005 | Verify that users can receive weather alerts for extreme conditions | This test case ensures users can opt-in to receive timely weather alerts for extreme weather events, such as heavy rain, storms, or extreme

heat, to prioritize safety during outdoor activities. | Weather Integration | High | - User has enabled weather alerts in the app settings. <br> - User has allowed notifications from the app. | Weather Alert: Extreme heat warning | 1. Open the Fitness Tracker App. <br> 2. Navigate to the app settings. <br> 3. Enable weather alerts. <br> 4. Simulate an extreme heat warning scenario for the user's location. <br> 5. Observe the app's behavior. | None | The app should display a clear and timely notification alerting the user about the extreme heat warning. | Critical | High | Functional, Integration, Acceptance | 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 Cases for Fitness Tracker App - Mood Tracking Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Priority	Type of Testing	Test Case Approach
FT-MT-001	Verify that users can log their mood, energy levels, and motivation before and after workouts.	This test case verifies the core functionality of mood tracking. It ensures users can successfully input and save their mood, energy levels, and motivation both before starting and after completing a workout.	Mood Tracking	High	- User account created and logged in.<br>- Workout logging feature enabled.	No test data needed	1. Navigate to the 'Workout' section.<br>2. Start a new workout or select an existing one.<br>3. Before starting, log mood, energy, and motivation levels using the provided scales/inputs.<br>4. Complete the workout.<br>5. After finishing, log mood, energy, and motivation levels again.	- Verify data is saved with the workout.<br>- Optionally, clear logged data for future tests.	Mood, energy, and motivation levels are successfully logged and associated with the corresponding workout.	Blocker	High	Functional, Regression	Positive
FT-MT-002	Verify that the app provides insights into mood and energy trends over time.	This test case examines the app's ability to analyze and present meaningful insights from collected mood and energy data. It ensures users can view trends, patterns, and correlations between their workouts and mental well-being over a

specified period. | Mood Tracking | High | - User account with at least 7 days of mood and workout data.<br>- 'Insights' or 'Trends' section accessible. | No test data needed | 1. Navigate to the 'Insights' or 'Trends' section related to mood tracking.<br>2. Select a time frame (e.g., past week, month).<br>3. Observe the displayed charts, graphs, or summaries visualizing mood and energy trends.<br>4. Interact with any filters or drill-down options (if available). | None | The app displays clear and accurate visualizations of mood and energy trends over the selected period, including (but not limited to) average mood scores, energy level fluctuations, and potential correlations with workout types or frequency. | Critical | High | Functional, Regression | Positive |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FT-MT-003 | Verify that users receive recommendations based on their mood and energy data. | This test case focuses on the app's recommendation engine and its ability to provide personalized suggestions based on the user's mood and energy patterns. It ensures recommendations are relevant, actionable, and tailored to individual user data. | Mood Tracking | Medium | - User account with sufficient mood and workout data for personalized recommendations.<br>- 'Recommendations' section accessible. | No test data needed | 1. Navigate to the 'Recommendations' section.<br>2. Observe the presented recommendations related to workouts, mood-boosting activities, or other relevant suggestions. | None | The app provides personalized recommendations that align with the user's tracked mood and energy data, aiming to improve mental well-being and encourage positive behavioral changes. | Major | Medium | Functional, Regression | Positive |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FT-MT-004 | Verify that the app supports journaling for personal reflections on mood and workouts. | This test case verifies the journaling functionality within the mood tracking feature. It ensures users can create, save, and access their personal journal entries, allowing them to reflect on their workouts and emotional well-being. | Mood Tracking | Medium | - User account created and logged in. | No test data needed | 1. Navigate to the 'Journal' or 'Notes' section within mood tracking.<br>2. Create a new journal entry.<br>3. Input text reflecting on mood, workout experience, or any relevant thoughts.<br>4. Save the journal entry.<br>5. Access and view previously saved journal entries. | None | Users can successfully create, save, and access their journal entries within the app, providing a space for personal reflections and self-analysis. | Minor | Medium | Functional, Regression | Positive |

FT-MT-005	Verify that users can set goals for improving mood and energy levels.	This test case confirms the availability and functionality of goal setting specifically related to mood and energy improvement. It ensures users can define, track, and manage their goals within the app.	Mood Tracking	Medium	- User account created and logged in. - 'Goals' section accessible within mood tracking.	No test data needed	1. Navigate to the 'Goals' section related to mood and energy. 2. Select the option to create a new goal. 3. Define the goal (e.g., increase average mood score, improve energy levels by X%). 4. Set a target date or timeframe for achieving the goal (if applicable). 5. Save the goal. 6. View and track progress towards the set goal.	None	Users can successfully set, save, and track their mood and energy-related goals within the app, allowing them to work towards specific mental well-being objectives.	Minor	Medium	Functional, Regression	Positive

---

Data - 99

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Creation

Feature description -

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

Userstory title -

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

Acceptance criteria -

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', "Recipes are saved in the user's profile.", 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

Generated Test Cases:

## Test Cases for Recipe Creation Feature

Test Case ID: TC_RC_001

Test Case Title: Verify that users can input and save ingredients and cooking instructions.

Test Case Description: This test case verifies the functionality of adding and saving recipe ingredients and instructions in the recipe creation form.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User has navigated to the recipe creation page.

Test Data:

- Recipe Name: "Spaghetti Aglio e Olio"

- Ingredients: "Spaghetti, Garlic, Olive Oil, Chili Flakes, Salt, Pepper, Parsley"

- Instructions: "1. Cook spaghetti. 2. Sauté garlic in olive oil. 3. Add chili flakes. 4. Toss with spaghetti. 5. Season and garnish."

Test Steps:

1. Enter the recipe name in the designated field.

2. Enter each ingredient on a new line in the "Ingredients" field.

3. Enter the cooking instructions in the designated field.

4. Click the "Save Recipe" button.

Postconditions:

- Recipe is successfully saved.

- User is redirected to the recipe view page.

Expected Result: The recipe should be saved with the provided name, ingredients, and instructions, and be visible on the user's profile.

Severity: Blocker

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_RC_002

Test Case Title: Verify that users can upload photos for their recipes.

Test Case Description: This test case verifies the functionality of uploading a photo while creating a new recipe.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User is on the recipe creation page.

Test Data:

- Valid image file (JPEG or PNG format).

Test Steps:

1. Click on the designated area to upload a recipe photo.

2. Select a valid image file from your device.

3. Verify that the image preview is displayed correctly.

4. Click the "Save Recipe" button.

Postconditions:

- Recipe is saved with the uploaded photo.

Expected Result: The recipe should be saved, and the uploaded photo should be displayed along with the recipe details.

Severity: Major

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_RC_003

Test Case Title: Verify that recipes are saved in the user's profile.

Test Case Description: This test case verifies that newly created recipes are saved and accessible from the user's profile page.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in to the platform.

- User has created a new recipe.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user's profile page.

2. Locate the "My Recipes" section.

Postconditions: No specific postconditions.

Expected Result: The newly created recipe should be listed in the "My Recipes" section on the user's profile page.

Severity: Blocker

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_RC_004

Test Case Title: Verify that recipes are accessible to other users on the platform.

Test Case Description: This test case verifies that other logged-in users can view the newly created recipe.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- Two users are logged in to the platform (User A and User B).
- User A has created a new recipe.

Test Data: No test data needed.

Test Steps:

1. Log in as User B.
2. Use the platform's search functionality to locate the recipe created by User A.

Postconditions: No specific postconditions.

Expected Result: User B should be able to find and view the recipe created by User A.

Severity: Blocker

Priority: High

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Test Case ID: TC_RC_005

Test Case Title: Verify that users receive a confirmation upon successful recipe submission.

Test Case Description: This test case verifies that a confirmation message is displayed to the user upon successful recipe creation.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in to the platform.

- User is on the recipe creation page.

Test Data: No test data needed.

Test Steps:

1. Fill in all required recipe details (name, ingredients, instructions).

2. Click the "Save Recipe" button.

Postconditions: No specific postconditions.

Expected Result: A success message (e.g., "Recipe successfully created!") should be displayed to the user, confirming that the recipe has been saved.

Severity: Minor

Priority: Medium

Type of Testing: Functional, Acceptance

Test Case Approach: Positive

Data - 100

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Discovery

Feature description -

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

Userstory title -

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance criteria -

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

Generated Test Cases:

## Test Cases for Recipe Sharing Platform - Recipe Discovery

Test Case ID: RD-001

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

Test Case Description: This test case verifies that the recipe search functionality correctly returns relevant results when a user searches using a single keyword related to a recipe title or description.

Test Suite: Recipe Search

Test Priority: High

Preconditions:

- User is logged into the platform.

- User is on the "Search Recipes" page.

Test Data:

- Keyword: "Chicken"

Test Steps:

1. Enter the keyword "Chicken" into the search bar.

2. Click the "Search" button or press Enter.

3. Observe the search results page.

Postconditions:

- None

Expected Result: The search results page should display a list of recipes containing "Chicken" in their title or description.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RD-002

Test Case Title: Verify that users can search for recipes using a combination of keywords and ingredients.

Test Case Description: This test case ensures users can search for recipes using multiple keywords and specific ingredients, refining the search results to match their specific requirements.

Test Suite: Recipe Search

Test Priority: High

Preconditions:

- User is logged into the platform.

- User is on the "Search Recipes" page.

Test Data:

- Keywords: "Pasta", "Tomato"

- Ingredients: "Garlic", "Basil"

Test Steps:

1. Enter keywords "Pasta", "Tomato" into the search bar.

2. Select "Garlic" and "Basil" from the suggested ingredient list or manually add them.

3. Click the "Search" button or press Enter.

4. Observe the search results page.

Postconditions:

- None

Expected Result: The results should display recipes containing all specified keywords and ingredients ("Pasta", "Tomato", "Garlic", and "Basil").

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RD-003

Test Case Title: Verify that users can filter search results by cuisine type, dietary preferences, and other criteria.

Test Case Description: This test case ensures the search functionality allows for further refinement of results based on various filters like cuisine type, dietary needs, and other relevant criteria.

Test Suite: Recipe Search

Test Priority: High

Preconditions:

- User has searched for recipes using keywords or ingredients.

- Search results page is displayed.

Test Data:

- Cuisine type: Italian

- Dietary preference: Vegetarian

Test Steps:

1. On the search results page, locate the filter options.

2. Select "Italian" from the cuisine type filter.

3. Select "Vegetarian" from the dietary preferences filter.

4. Observe the updated search results.

Postconditions:

- None

Expected Result: The search results should be dynamically updated to only display recipes tagged as both "Italian" and "Vegetarian".

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RD-004

Test Case Title: Verify that recipes in the search results are accessible and detailed.

Test Case Description: This test case validates that each recipe in the search results can be accessed, and upon opening, displays detailed information, including ingredients, instructions, and user reviews.

Test Suite: Recipe Search

Test Priority: High

Preconditions:

- User has performed a recipe search.

- Search results page displays a list of recipes.

Test Data:

- None

Test Steps:

1. Select a recipe from the search results list.

2. Observe the displayed recipe page.

Postconditions:

- None

Expected Result: The recipe page should load correctly, displaying the recipe title, author, image (if available), list of ingredients, detailed instructions, nutritional information (if provided), user ratings, and reviews.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: RD-005

Test Case Title: Verify that users can save search results for future reference.

Test Case Description: This test case checks if users can save entire search results or individual recipes for easy access later, enhancing user experience and encouraging recipe exploration.

Test Suite: Recipe Search

Test Priority: Medium

Preconditions:

- User has performed a recipe search.

- Search results page displays a list of recipes.

Test Data:

- None

Test Steps:

1. Choose a recipe from the search results.

2. Click on the "Save Recipe" button (or similar call to action).

3. Navigate to the user's saved recipes section.

Postconditions:

- None

Expected Result: The saved recipe should appear in the user's saved recipes list for future access.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive
