Spotify – Web Player
https://open.spotify.com/

__Spotify Web Player Playlist functionality__

Table of Contents

1.   INTRODUCTION

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the modules: "Main page", "Sign Up", "Log In", "Playlist" of the website https://open.spotify.com/.

This document has clearly identified sub-tasks of website's module testing, types of testing, what devices, environments and tools to be used. It includes test cases with expected and actual results, schedule and executing estimated time each of sub-tasks and entire project.

2.   SCOPE

The document mainly targets the GUI, Exploratory, Functional Positive, Negative and Ad-hoc testing of User's Personal Account Address Menu, Performance and Security testing of the website.

3.   SUB-TASKS

3.1. WEBSITE MANUAL TEST

Execute manual testing for the User's Personal Account Address Menu based on the next types of testing:
-        GUI;
-        Exploratory testing;
-        Functional Positive testing:
-        Functional Negative testing;
-        Functional Ad-hoc testing.

   GUI Testing:
GUI testing includes testing the UI part. It covers look and feel, error messages, spelling mistakes, GUI guideline violations.

Exploratory Testing:

Exploratory testing will includes a type of software testing where Test cases are not created in advance but QA check system on the fly.

Functional Positive testing:

Checks whether an application behaves as expected with positive inputs.

Functional Negative testing:

Invalid data is inserted to compare the output against the given input.

Functional AD-hoc testing:

Includes an informal testing type with an aim to break the system.

Environment Support:

- Laptop
- OS: Windows 11/ 64
- Browsers (latest versions): Google Chrome, Firefox Mozilla, Microsoft Edge

The following test cases are covers these manual tests:

TC	Test Action	Expected Results
POSITIVE TESTS		
01	The site must be accessible at the following URL: https://open.spotify.com/	Https protocol is active
02	Verify that the user can sign up into the site using a correct mail and password	The user successfully registered on the sites
03	Verify that the user can log into the site using an existing mail and password	The user successfully entered on the sites
04	Validate that menu links lead to the right pages ("Home", "Search", "Your Library", "Liked Songs")	Each menu item is processed according to its name and URL
05	Verify that the user can find a specific query (artist, song, playlist, album)	Valid request found
06	Check that the user can create new "Playlist"	New playlist successfully created
07	Check that the user can change name of "Playlist"	Playlist name changed successfully
08	Check that the user can add item to a user's playlist	Item added to playlist successfully
09	Check that the user can remove item from a user's playlist.	Item deleted to playlist successfully
10	Check that the user can follow an existing playlist	Playlist successfully added in "Your Library"
11	Check that the user can unfollow an existing playlist	Playlist successfully deleted from "Your Library"
NEGATIVE TESTS		
12	Verify that the Log in with invalid email address is not accepted	Got the error message: "Incorrect username or password."

13	Verify that the Log in with invalid password is not accepted	Got the error message: "Incorrect username or password."
14	Verify that the Log in with empty email field is not accepted	Got the error message: "Incorrect username or password."
15	Verify that the Log in with empty password field is not accepted	Got the error message: "Incorrect username or password."
AD HOC TESTS		
16	Verify that the registration with invalid (ad hoc) email address (test@mail.comtest@mail.com) is not accepted	Got the error message: "Incorrect username or password."

Testing Tools used:

Process	Tools
Test case creation	Microsoft Word, Microsoft Excel, JIRA
Test case tracking	Microsoft Word, Microsoft Excel, JIRA
Test case execution	Manual
Test case management	Microsoft Excel, JIRA
Test reporting	Microsoft Excel, JIRA

3.2. WEBSITE AUTOMATION TEST

Automation tests are creates by using Selenium WebDriver Python UnitTest frameworks on the basis of Functional manual test cases and executes to check functionality of User's Personal Account Address Menu.

Environment Support:
- Laptop
- OS: Windows 11/ 64
- Browsers (latest versions): Google Chrome, Firefox Mozilla, Microsoft Edge

The following test cases are covers automation tests:

TC	Test Action	Expected Results
POSITIVE TESTS		
01	Verify that the user can log into the site using an existing mail and password	The user successfully entered on the sites
02	Verify that the user can find a specific query (artist, song, playlist, album)	Valid request found
03	Check that the user can create new "Playlist"	New playlist successfully created
04	Check that the user can change name of "Playlist"	Playlist name changed successfully

05	Check that the user can add item to a user's playlist	Item added to playlist successfully
06	Check that the user can remove item from a user's playlist.	Item deleted to playlist successfully
07	Check that the user can follow an existing playlist	Playlist successfully added in "Your Library"
08	Check that the user can unfollow an existing playlist	Playlist successfully deleted from "Your Library"
NEGATIVE TESTS		
09	Verify that the Log in with invalid email address is not accepted	Got the error message: "Incorrect username or password."
10	Verify that the Log in with invalid password is not accepted	Got the error message: "Incorrect username or password."
11	Verify that the Log in with empty email field is not accepted	Got the error message: "Incorrect username or password."
12	Verify that the Log in with empty password field is not accepted	Got the error message: "Incorrect username or password."
AD HOC TESTS		
13	Verify that the registration with invalid (ad hoc) email address (test@mail.comtest@mail.com) is not accepted	Got the error message: "Incorrect username or password."

Note: For ease of perception, the tests are numbered starting from the first, separately from manual tests.

Testing Tools used:

Process	Tools
Test case creation	PyCharm, Selenium
Test case tracking	PyCharm, Browsers
Test case execution	Automation
Test case management	PyCharm
Test reporting	HTML reports, Allure reports

Additional automation tests execute on the cloud platform "Browserstack" :

- OS: Windows 10 (Browsers (latest versions): Google Chrome, , Firefox Mozilla, Microsoft Edge)
- OS: macOS Big Sur (Browsers: Safari, version 14.1)

Testing Tools used:

Process	Tools
Test case creation	PyCharm, Selenium
Test case tracking	Browserstack.com
Test case execution	Automation
Test case management	PyCharm
Test reporting	Video files (mp4)

Test results are the same as on local device.

3.3. WEBSITE API TEST

TC	Test Action	Expected Results
	POSITIVE TESTS	
19	POST. Create new Playlist	Status: 201 Created
20	PUT. Change Playlist Details	Status: 200 OK
21	POST. Add Items to Playlist	Status: 201 Created
22	DEL. Remove Playlist Items	Status: 200 OK
23	PUT. Follow Playlist	Status: 200 OK
24	DEL. Unfollow Playlist	Status: 200 OK
25	GET. Get Specific User's Playlists	Status: 200 OK
	NEGATIVE TESTS	
26.1	POST. Create playlist with invalid name format	"Error parsing JSON." Status: 400 Bad Request
26.2	POST. Create playlist with invalid public status format	"Error parsing JSON ." Status: 400 Bad Request
27.1	PUT. Change Playlist Details in invalid name format	"Error parsing JSON." Status: 400 Bad Request
27.2	PUT. Change playlist details in invalid description format	"Error parsing JSON." Status: 400 Bad Request
27.3	PUT. Change playlist details in invalid public status format	"Boolean value must be either true or false" Status: 400 Bad Request
28.1	POST. Add items to playlist with invalid data in body	"Error parsing JSON." Status: 400 Bad Request
28.2	POST. Add items to playlist with invalid uris (resource identifier)	"Invalid track uri: spotify:track:(invalid data)." Status: 400 Bad Request
29.1	DEL. Delete Playlist Items with invalid uris	"JSON body contains an invalid track uri: spotify:track:(invalid data)." Status: 400 Bad Request
29.2	DEL. Delete Playlist Items with invalid data in body	"JSON body doesn't conform to specification" Status: 400 Bad Request
29.3	DEL. Delete Playlist Items with empty data in body	"JSON body contains a track without URI." Status: 400 Bad Request
30.1	PUT. Follow Playlist with empty data in body	"Insufficient scope" Status: 403 Forbidden
30.2	PUT. Follow Playlist with invalid public data	"JSON body does not adhere to the defined endpoint parameters" Status: 400 Bad Request

API tests execute to determine whether the APIs that are developed meet expectations when it comes to functionality, performance, reliability and security for website.

Environment Support:
- Laptop
- OS: Windows 11/ 64
- Browsers (latest versions): Google Chrome (DevTools)

The following test cases are covers API tests:
(There are main tests results in the table)

Testing Tools used:

Process	Tools
Test case creation	Postman, Chrome DevTools
Test case tracking	Postman
Test case execution	Automation
Test case management	Postman
Test reporting	Newman

3.4. WEBSITE AUTOMATION PPERFORMANCE TEST

Performance automation tests execute for measures the speed, responsiveness and stability of the tested website. Tests execute in incognito environments.
After testing reports were generate.

Testing Tools used:

Process	Tools
Test case creation	Google Lighthouse, GTMetrix, Webpagetest, BrowserStack-SpeedLab
Test case tracking	Google Lighthouse, GTMetrix, Webpagetest BrowserStack-SpeedLab
Test case execution	Automation
Test case management	
Test reporting	txt -, html -, pdf - files

3.5. WEBSITE AUTOMATION SECURITY TEST

Security automation tests execute to reveal potential flaws or weaknesses of software and website. Testing focuses on whether the application is designed and configured correctly.
Tests execute in incognito environments.
After testing reports were generate.

Testing Tools used:

Process	Tools
Test case creation	Mozilla Observatory, Snyk
Test case tracking	Mozilla Observatory, Snyk
Test case execution	Automation
Test case management	
Test reporting	txt -, html -, pdf - files

4.   TESTS SCHEDULE AND ESTIMATED TIME

Sub-Task Name	Start	Est. time (hours)	Finish
Manual Testing			
Automation Testing			
API Testing			
Performance testing			
Security testing			

Test Execution
QA   -
Test phase   -
Status   -

APPROVALS:

	Project Manager	QA Lead
Signature		