**Metacritic website**

**GENERAL TEST PLAN:**

1. **Manual testing**. Performing the manual testing of the Metacritic website: <https://www.metacritic.com/> which includes exploratory, positive, and negative testing. The plan for the manual testing is to check the registration and login functionality with additional checkup of the personal account features.

**Manual testing test cases**:

1) Positive test. Check if the registration functions properly.

2) Positive test. Make sure the login works properly.

3) Positive test. Submit the review for the video game.

4) Negative test. See if we can register the new account with the invalid Username.

5) Negative test. See if you can login using the invalid email.

6) Negative test. See if you could write the review with less than 75 characters.

1. **Automation testing.** Full automatization of our manual testing by using the Selenium WebDriver tool in PyCharm environment. The plan for the automation testing is to create the cross-browser testing Selenium automation scripts which examine login and other additional personal account features such as submitting the review.

**Automation testing test cases**

1) Positive test. Create an automation script which checks the login functionality.

2) Positive test. Create a script which automatically submits the valid review for the video game.

3) Negative test. Automate the registration script in which we input the invalid username.

4) Negative test. Automate the invalid email input in the login section.

5) Negative test. Create an automation script in which we write the review with less than 75 characters.

1. **API Testing.** Create the API collection and fully automatize it in the Postman app. The plan for the API testing is to create requests to POST, GET and DELETE address on our account on California Marketing Website. We need to fully inspect response from the server by creating prescripts, tests, and the appropriate environment. 1) Create the POST request which creates a new address and additional tests that check the response. Create the GET request which fetches the info about the newly created address. Create a PATCH request that makes changes to the address. Create the DELETE request that deletes this new address. Create the DELETE request again to check if the address is deleted indeed. Then create the GET request to make sure that we cannot fetch the deleted address.
2. **Automated Performance testing**. Completing the performance testing by using: Lighthouse, GTMetrix and SpeedLab tools. The plan is to use the appropriate tools to confirm that the performance on the website is plausible.
3. **Automated Security testing.** Perform the security testing by using Mozilla Observatory tools. The plan is to use contrasting tools to investigate the security level on the website.