VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Software Testing - ITE2004

Fall Semester 2022-23

PROJECT TITLE:

INSTAGRAM TESTING

Submitted by

Manya S-20BIT0091
Aazim Ajmal PV-20BIT0125
Karan Ravi- 20BIT0162
Ananya Grover-20BIT176

Under the guidance of

Dr. Prasanna M
Professor Grade 1, SITE,
VIT, Vellore

TABLE OF CONTENTS

1. ABSTRACT

With over 500 million daily active users, the number of daily users on Instagram is rapidly increasing. Instagram is a free photo and video-sharing app available on iPhone and Android. People can upload photos or videos to it and share them with their followers or with a selected group of friends. They can also view, comment, and like posts shared by their friends on Instagram. Anyone aged 13 and over can create an account by registering an email address and selecting a username. This project aims to test this popular app and possibly discover bugs.

2. SOFTWARE REQUIREMENTS SPECIFICATION

2.1 INTRODUCTION

This document is prepared in the context of a software requirement specification for Instagram. Instagram (also known as IG or Insta) is a photo and video-sharing social networking service owned by Meta, Inc.
The primary scope of this document is to provide an overall description of the functioning of Instagram including its system features and its specific requirements. The functional and nonfunctional requirements of the system will be addressed as well.

PURPOSE

The SRS is needed to evolve as the development of the software production processes. The purpose of this document is to give a complete description of how a social network system can be developed. This document is to provide information about what the software product is to do to customers and establish an agreement between customers and suppliers and also become helpful for development. In addition to these, it provides a basis for validation and verification.

SCOPE

The name of the software product is Instagram. Instagram is a social network that connects people. The aim of Instagram is to provide information to the users about events and people they know. The main objective of the document is to provide the requirements for the development of Instagram. The users of Instagram can control following and unfollowing friends, share posts and reels with their friends, upload photos, comment on their friend's posts, and chat with their friends. In addition, people can create social groups for university clubs, football clubs, or for social awareness.

DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

When a user logins into Instagram, they can view their home page, to see posts shared by whom they follow. Additionally, users can add and view stories.

2.2 OVERALL DESCRIPTION

PRODUCT PERSPECTIVE:

Instagram is an independent and worldwide network system. Every person can use it without any cost. People from different regions of the world can connect to each other via this system. regions of the world can connect to it and exchange information with other people. In order to control the contents of the sharing and comments done by other people, Instagram has also a control mechanism.

PRODUCT FUNCTION:

After creating an account using Instagram people can search for each other's Instagram. After sending follow request they can see each other's posts such as photos, videos, newsfeeds, etc. They can like, comment, and share each other's posts. They can also chat with each other.

USER CHARACTERISTICS:

Instagram does not require any prior computer knowledge and can be used by anybody with features limited according to their age.

DESIGN IMPLEMENTATION:

Being a social networking System, its design should be user-friendly and inviting. It should be secure enough so that the user data and their personal information should not be leaked and should be securely preserved in the system.

2.3 EXTERNAL INTERFACE REQUIREMENT

INTERFACE REQUIREMENT:

Various interfaces for Instagram could be-

1. login page
2. Home page
3. There will be a screen to display follow requests, view/Search accounts, chat, etc.

HARDWARE INTERFACE:

The whole system runs over the internet. Without an internet connection, the system will not work. All the hardware should be connected to the internet.

SOFTWARE INTERFACE:

The system runs on a server hence, there will be scripting languages such as PHP, HTML, VBScript, etc. The database is also required for the storage purpose of any transaction like MYSQL.

2.4 SPECIFIC REQUIREMENTS

In this section, all software requirements will explain some information to the user.

All requirements are divided into two groups-

2.4.1 Functional
2.4.2 Non-functional

2.4.1 FUNCTIONAL REQUIREMENTS:

A functional requirement defines the function of a system or its components.

FR 1-CREATE ACCOUNT

The user can simply sign in using a Meta account or create a new account by providing required personal information about himself/herself. After creating the account, it will display the account created successfully.

FR 2-SEARCH ACCOUNTS

Search for people the user wants to view/follow.

FR3-SENDING FOLLOW REQUEST.

On finding an account, users can send follow request. The user can also accept or deny follow requests.
FR4-ACCEPTING FOLLOW REQUEST

By accepting others' follow requests, the user will allow another user to view their posts and stories. Similarly, when a follow request has been accepted, the user who sends the follow request can view their posts.

FR5-UPLOADING PHOTOS, REELS:

Photos and videos can be uploaded by the user on selecting the respective file from their gallery.

FR6-CREATING BUSINESS ACCOUNT:

There are two types of accounts on Instagram-
a.Personal Account (Public/Private)
b.Business Account
An account can be public I.e. visible to everyone or it can be private where the user controls who views their posts.
One can simply turn their personal account into a business account by connecting their account to a Meta page that personal account.

FR7-ADDING STORIES:

The user should be able to upload stories that can be viewed by others only for 24 hours. The user can also highlight and archive the stories.

FR8-EDIT PROFILE:

In this very section, the user can update or edit their profile information. One can update their bio, upload a new profile picture and manage posts.

FR9-LIKE, COMMENT, SHARE:

The user can like, comment and share posts.

FR10-SENDING MESSAGE

Users can send messages and posts to each other in the Direct messages section. Users can also send voice messages, and call each other (voice or video).

FR11-PRIVACY SETTING

In addition to being able to make their account public or private, the users can control which specific accounts can view their posts and stories.

2.4.2 NON-FUNCTIONAL REQUIREMENTS

SECURITY:

The system uses SSL(secured socket layer) in all transactions that include any other confidential passenger information. The system should be secure in such a way that it should not show any cookies regarding passwords or usernames so that no one other than the user can access the system. Instagram provides one-factor authentication, the user can also enable two-factor authentication if needed.

RELIABILITY:

The system provides a database for the storage of all kinds of devices. The reliability of the whole system depends on the reliability of the separate components. The system should be reliable and should not crash or hang in case of bugs or errors.
The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of the container and its underlying operating system.

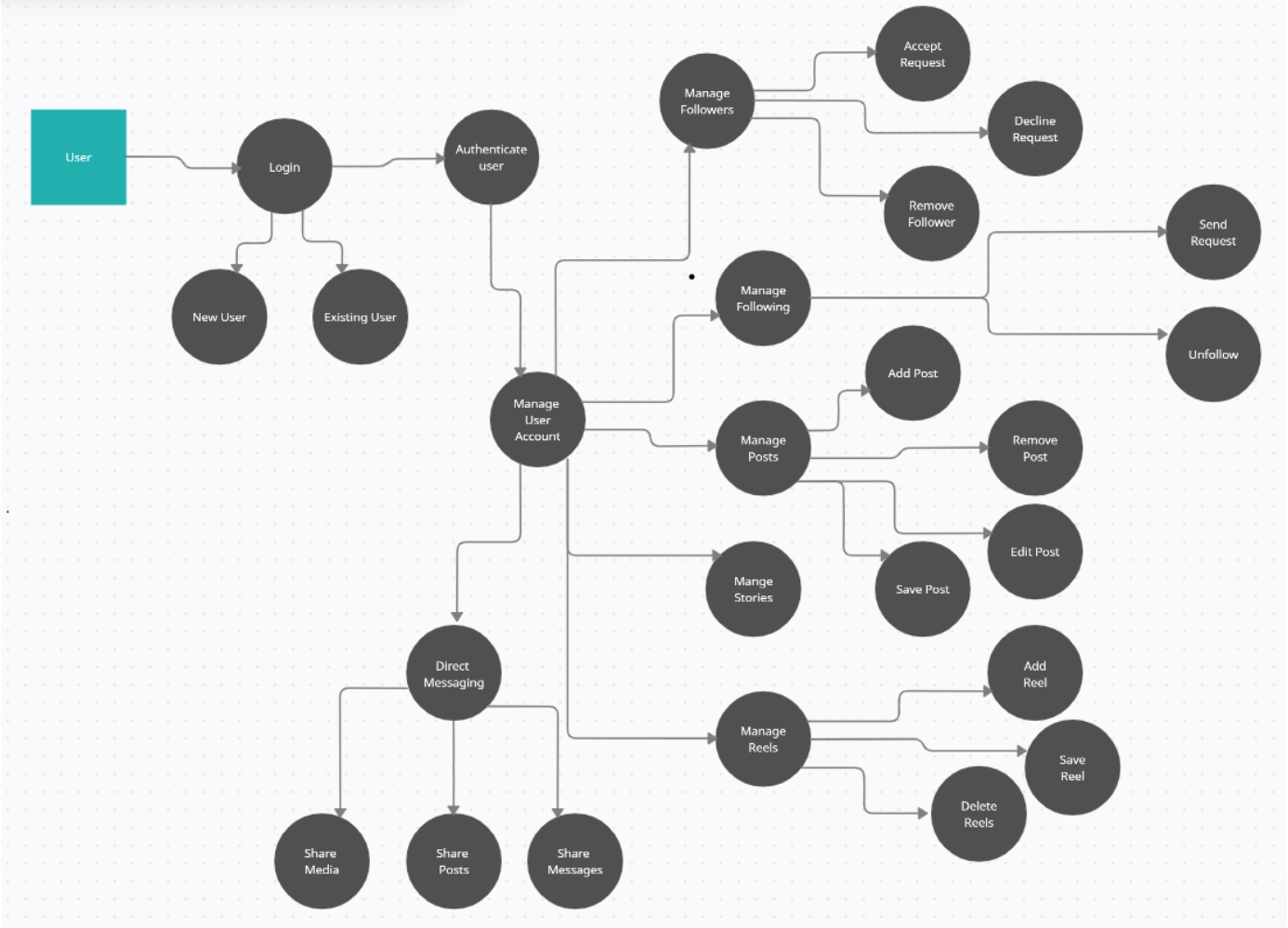
MAINTAINABILITY:

A commercial database is used for maintaining the database and the application server takes care of the site. In case of failure, a re-initialization of the program will be done. Also, the software design is being done with modularity in mind so that its maintainability can be done efficiently .

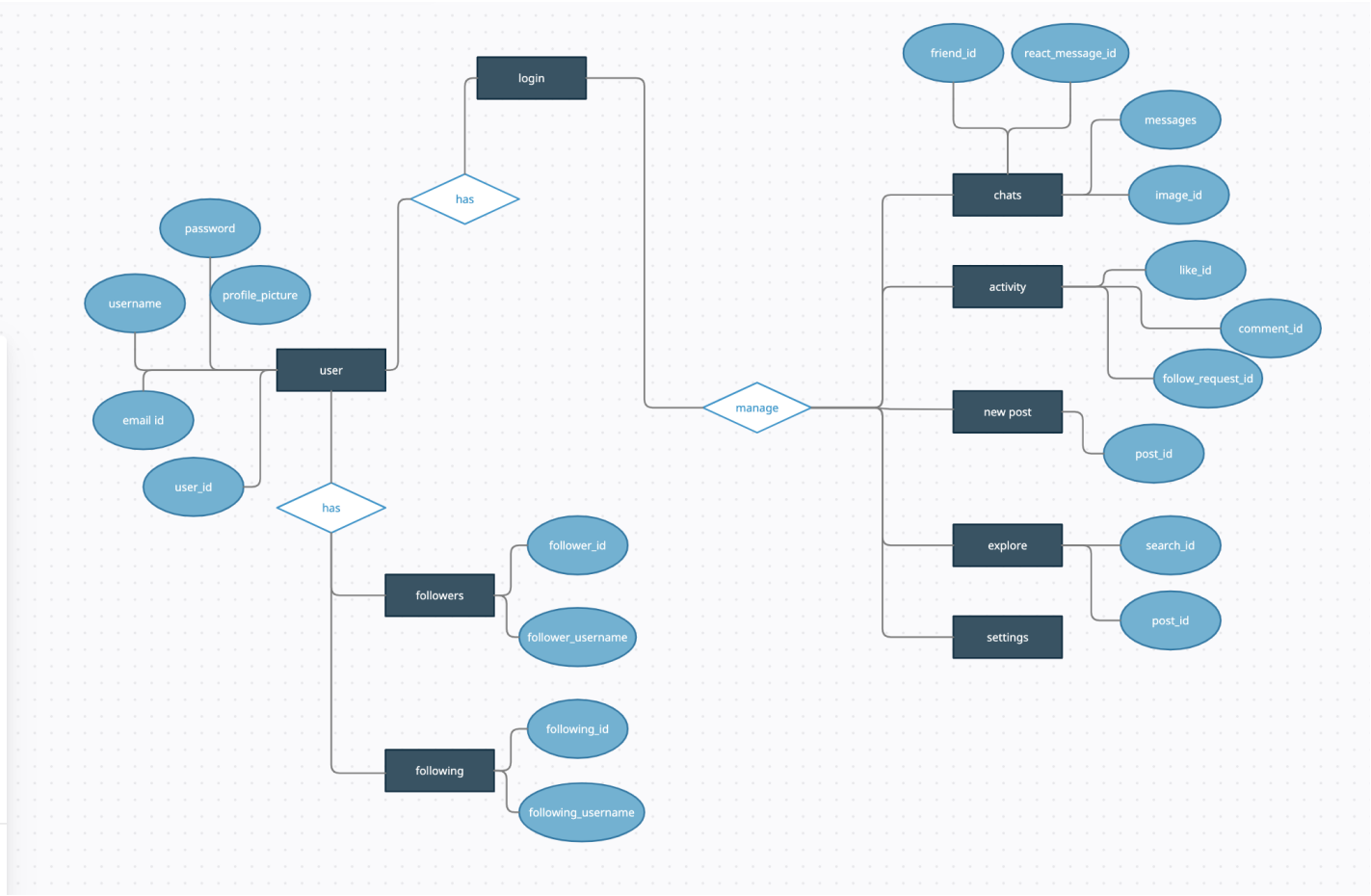
PORTABILITY:

The application is HTML and scripting language based. So That end user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user can use this system on any OS. The system should be able to run on any device.

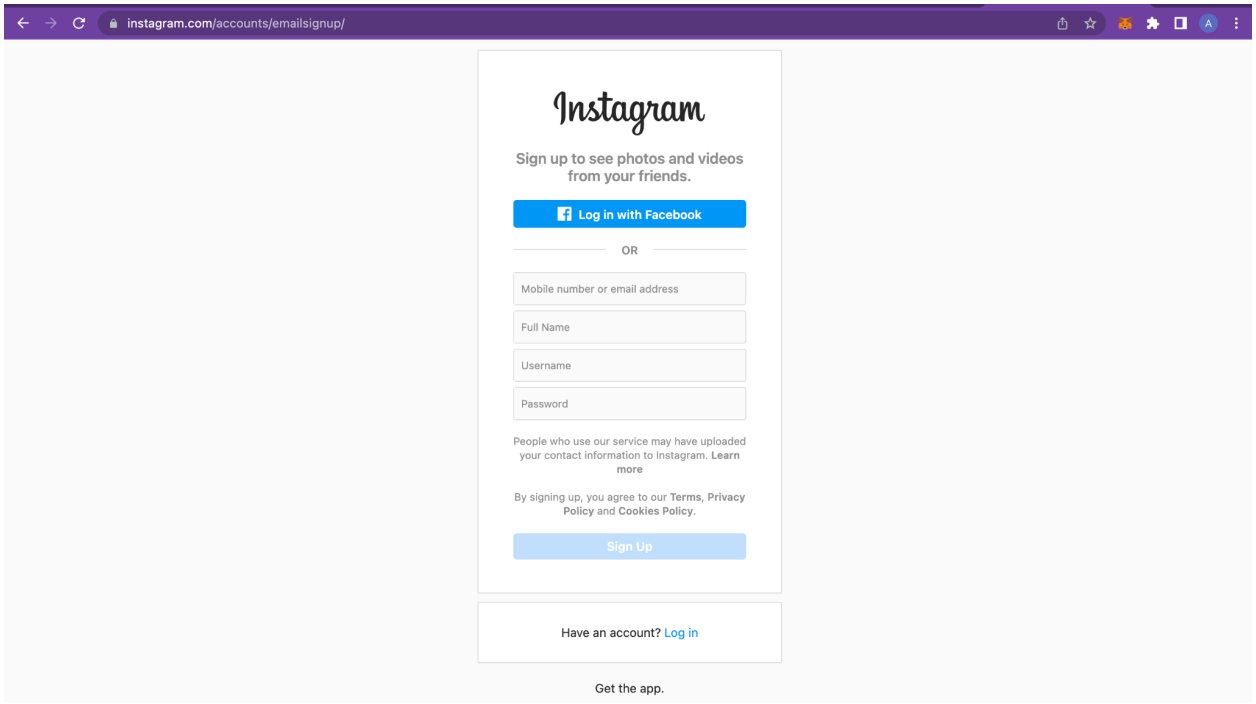
3. DATA FLOW DIAGRAM

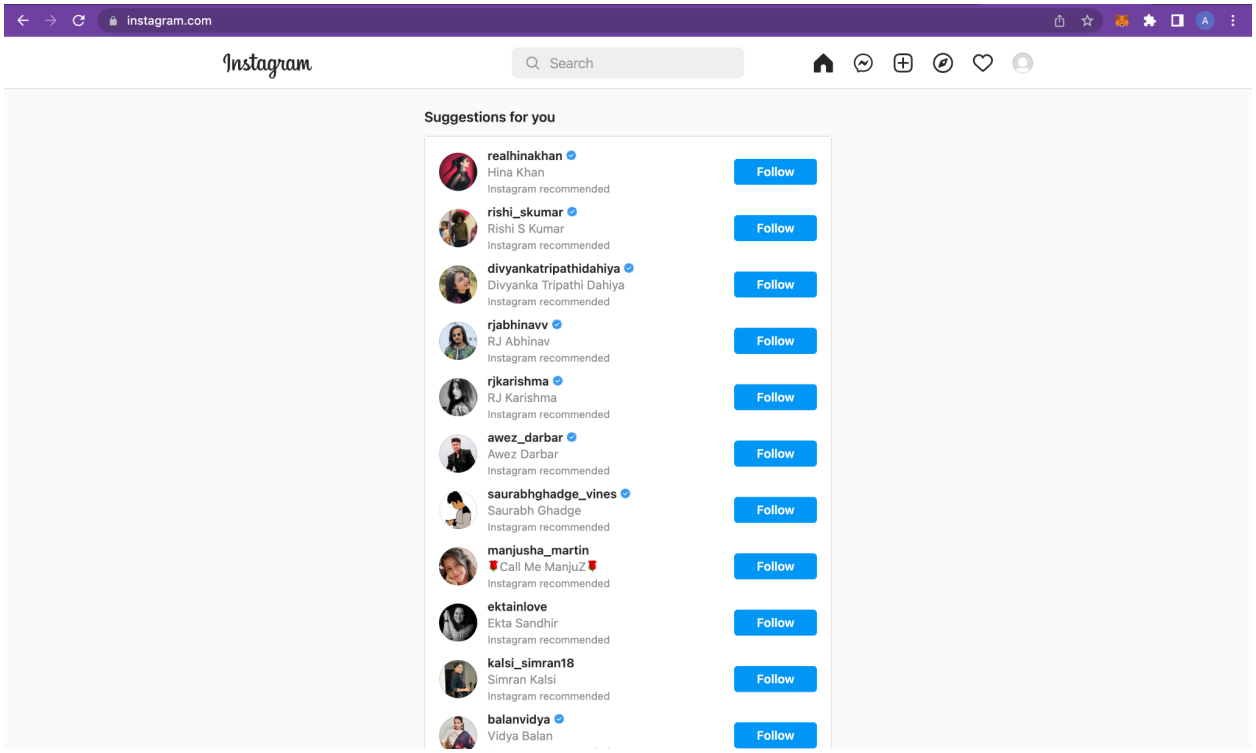


4. ER DIAGRAM


login
has
password
profile_picture
username
user
email id
user_id
has
followers
follower_id
follower_username
following
following_id
following_username
manage
friend_id
react_message_id
messages
chats
image_id
activity
like_id
comment_id
follow_request_id
new post
post_id
explore
search_id
post_id
settings

5. USER INTERFACE





SOFTWARE INTERFACES:

Operating system: Windows XP or higher and MAC

Language: JAVA

Web Browser: Mozilla Firefox, Google Chrome.

Tools: Apache Jmeter, Junit, OWASP Zap.

HARDWARE INTERFACES:

Processor: Pentium IV and above

RAM: 512 MB or above

Hard Disk: 40GB or above

Input Devices: Keyboard, mouse.

Output Devices: Monitor.

6. FUNCTIONAL TEST CASES

Sr.No	TestCase_ID	TestCase_Objective
1	Login_01	To verify Login functionality with valid email id and valid password.
2	Login_02	To verify Login functionality with valid email address and invalid password.

3	Login_03	To verify Login functionality with invalid email address and valid password.
4	Login_04	To verify Login functionality with invalid email address and invalid password.
5	Login_05	To verify Login functionality with blank email address and valid password.
6	Login_06	To check that Login functionality with valid email id and blank password.
7	Login_07	To verify Login functionality with blank email address and blank password.
8	Login_08	To check that Login functionality with valid phone number and valid password.
9	Login_09	To check that Login functionality with valid phone number and invalid password.
10	Login_10	To check that Login functionality with invalid phone number and valid password.
11	Login_11	To check that Login functionality with invalid phone number and invalid password.

12	Login_12	To check that Login functionality with blank phone number and valid password.
13	Login_13	To verify that length of email address field and password field.

14	Login_14	To verify that error message display when any field is left blank.
15	Login_15	To verify Tab key functionality on the Login page.
16	Login_16	To verify that remember me checkbox functionality
20	Login_20	To verify that entered multiple times incorrect passwords.
21	Login_21	To verify that welcome message after successfully login into application.
22	Login_22	To verify that Forgotten Password functionality.
23	Login_23	To verify if password text format is encrypted or not into password field.
24	Login_24	To verify that back button of the browser after log out.
25	Login_25	To check that message for entered invalid inputs.

26	Login_26	To verify that Login button with click and Enter key events.

UI Testing

Generally, in UI testing, we are testing the layouts, text fields, radio buttons, checkbox, and drop-down list. The Login page should have an easy interface for users. So users can easily interact with the application.

1	Login_UI_1	To check Login page layout is as per specification or not.
2	Login_UI_2	To check that placeholders are displayed properly or not.
3	Login_UI_3	To check that red (*) mark is properly displayed for mandatory fields or not.

4	Login_UI_4	To check that text fields,buttons,checkbox are displayed as per specification or not.
5	Login_UI_5	To check that the Login page is responsive or not.

Security Testing

1	Login_Security_01	To verify that SSL certificate is implemented or not.

2	Login_Security_02	To verify that “Back” button of the browser after successfully logged out from the application.
3	Login_Security_03	To verify that a user is able to login by directly entering url in the browser or not.
4	Login_Security_04	To verify that login session timeout functionality.
5	Login_Security_05	To verify that password format should be encrypted or not.
6	Login_Security_06	To verify that a user is able to enter more than characters as per specified into Email and Password fields or not.

Sign-up Test Cases

SR. No	**Testcase_ ID**	**Testcase_Objective**
1	SignUp_01	To verify that all required fields are present in the form.
2	SignUp_02	To verify that all required text fields have a valid placeholder.
3	SignUp_03	To verify that all required fields have a red * mark.
4	SignUp_04	To verify tab key functionality on the sign-up page.

5	SignUp_05	To verify Enter Key functionality for sign up button.
6	SignUp_06	To verify sign-up page load with none of the data into the text field.
7	SignUp_07	To verify the lower and upper limit of the Firstname text field.
8	SignUp_08	To verify the Firstname field without any data.
9	SignUp_09	To verify the lower and upper limits of the Lastname text field.
10	SignUp_10	To verify the Last name field without any data.
11	SignUp_11	To verify lower and upper limit numbers of Mobile number fields.
12	SignUp_12	To verify alpha characters allow in the Mobile number field.
13	SignUp_13	To verify special characters allowed in the Mobile number field.
14	SignUp_14	To verify the Mobile number field without any data.
15	SignUp_15	To verify the mobile number with the area code.

16	SignUp_16	To verify validation message for incorrect mobile number
17	SignUp_17	To verify email id text without @ Symbol
18	SignUp_18	To verify the email id field with Special characters.
19	SignUp_19	To verify email id field with garbage email id.
20	SignUp_20	To verify email id field with already registered email id.

21	SignUp_21	To verify validation message for incorrect email id
22	SignUp_22	To verify password field with lower and upper characters length.
23	SignUp_23	To verify password field with lower and upper characters length.
24	SignUp_24	To verify Password field with only strings
25	SignUp_25	To verify Password field with only enter numbers
26	SignUp_26	To verify Password field with only enter special characters

27	SignUp_27	To verify the Password field with alphanumeric characters.
28	SignUp_28	To verify Password field with alpha and special characters
29	SignUp_29	To verify the Password field with numeric and special characters.
30	SignUp_30	To verify the Password field without data.
31	SignUp_31	To verify validation message for incorrect password format
32	SignUp_32	To verify confirm password field with different password.
33	SignUp_33	To verify confirm password field with the same password
34	SignUp_34	To verify confirm password field without data.
35	SignUp_35	To verify validation messages for different passwords.
36	SignUp_36	To verify that select month from month drop downlist from DOB
37	SignUp_37	To verify that select date from Date drop downlist from DOB

38	SignUp_38	To verify that select Year from Year drop downlist from DOB
39	SignUp_39	To verify that select the Female radio button from gender.
40	SignUp_40	To verify that select the Male radio button from gender.
41	SignUp_41	To verify that select the Custom radio button from gender.
42	SignUp_42	To verify the sign-up button without filling up any data on the sign-up page.
43	SignUp_43	To verify the sign-up button by entering the key and click the event.
44	SignUp_44	To verify user gets a successful signup message or not.
45	SignUp_45	To verify sign up button with multiple click events

7. POSITIVE AND NEGATIVE TEST SCENARIOS

S.No.	Description	Positive/Negative
1.	Validate user and facilitate login if the entered credentials are correct	Positive
2.	Validate user and do not facilitate login if the entered the entered credentials are incorrect	Negative
3.	User profile should be displayed on searching a valid username	Positive
4.	Error message should be displayed on searching an invalid username	Negative
5.	Clicking on “Enable Two Factor Authentication” should activate this option.	Positive
6.	Entering the explore page should show suggested posts and reels.	Positive
7.	Users should be able to view feeds containing posts from the users they follow	Positive
8.	Users should not be able to view feeds containing posts from blocked users	Negative

9.	Stories uploaded by users should not be visible after 24 hours	Negative
10.	Clicking on the Profile and then clicking on the Saved tab should show all posts that have been saved.	Positive
12.	Clicking on create and choosing a post should facilitate the user to upload a photo to their profile.	Positive
13.	Users should not be able to view posts, stories, or reels of private accounts they don't follow	Negative
14.	Users should be able to send messages, and share posts and reels with other users	Positive
15.	Users can post only ten images per post	Negative

8. BLACK-BOX TESTING

8.1 REQUIREMENTS-BASED TESTING

Requirements Specification for Instagram Testing

S.No.	Requirements Identifier	Description	Priority (High, Med, Low)
1.	BR-01	Entering only the valid credentials should facilitate login.	H
2.	BR-02	Clicking on the logout button should facilitate logging out.	H

3.	BR-03	Searching for a valid username should show the user.	H
4.	BR-04	Clicking on “Enable Two Factor Authentication” should activate this option.	M
5.	BR-05	Entering any invalid credential should NOT facilitate login.	H
6.	BR-06	Entering the explore page should show suggested posts and reels.	M
7.	BR-07	Clicking on the Profile and then clicking on the Saved tab should show all posts that have been saved.	M
8.	BR-08	Clicking on create and choosing post should facilitate the user to upload a photo to his profile.	M
9.	BR-09	Clicking on create and choosing post should facilitate the user to upload a video to his profile.	M
10.	BR-10	Clicking on create and then choosing reels should provide the user with options to create reels with suggested filters and audios.	H

Requirements Traceability Matrix

Req. ID	Description	Priority	Test case IDs	Test Conditions	Expected Output	Phase of Testing

BR-01	Entering only the valid credentials should facilitate login.	H	Instagram01	Use only valid credential.	Logged in Successfully	Integration
BR-02	Clicking on the logout button should facilitate logging out.	H	Instagram02	Click on More, then choose “Log out” option.	Logged Out Successfully	Integration
BR-03	Searching for a valid username should show the user.	H	Instagram03 Instagram-04	Click on the Search Button, then Search for “leomessi”. Search for “brfootball”	Shows the results with the most relevant result on top.	Integration
BR-04	Clicking on “Enable Two Factor Authentication” should activate this option.	M	Instagram05	From homepage, Click on More ->Privacy And Security -> Enable Two Factor Authentication	Two Factor Authentication Enabled Successfully.	Component
BR-05	Entering any invalid credential should NOT facilitate login.	H	Instagram06	Use any random (invalid) credential.	Please Enter valid Credentials or Sign up.	Integration

BR-06	Entering the explore page should show suggested posts and reels.	M	Instagram07	Go to homepage and then click on the Explore option	Explore page is opened	Integration
BR-07	Clicking on the Profile and then clicking on the Saved tab should show all posts that have been saved.	M	Instagram08	Go to homepage, then click on profile tab and then click on Saved.	Shows all saved posts and collections	Integration
BR-08	Clicking on create and choosing post should facilitate the user to upload a photo to his profile.	M	Instagram09	From homepage, click the create button and chose photo to upload.	Photo uploaded successfully to the users profile.	Unit
BR-09	Clicking on create and choosing post should facilitate the user to upload a video to his profile.	M	Instagram10	From homepage, click the create button and chose video to upload.	Video uploaded successfully to the users profile.	Unit

BR-10	Clicking on create and then choosing reels should provide the user with options to create reels with suggested filters and audios.	H	Instagram11	From homepage, click the create button and choose Create Reel..	Reel created successfully	Integration

Test Execution Data

S.No.	Req. ID	Priority	Test cases	Total test cases	Test cases passed	Test cases failed	% pass	No. of Defects
1	BR-01	H	Instagram-01	4	4	0	100	1
2	BR-02	H	Instagram-02	3	3	0	100	0
3	BR-03	H	Instagram03,04	4	3	1	75	2
4	BR-04	M	Instagram-05	2	2	0	100	1
5	BR-05	M	Instagram-06	2	1	1	50	4
6	BR-06	L	Instagram-07	1	1	0	100	1

7	BR-07	L	Instagram-08	1	1	0	100	2
8	BR-08	M	Instagram-09	2	2	0	100	1
9	BR-09	M	Instagram-10	4	4	0	100	3
10	BR-10	H	Instagram-11	1	0	1	0	1
Total	10			24	21	3	87	16

The test cases BR-01,02,03,04,05,06,08,09 should be automated and the test cases BR-07,10 should be done manually.

Hence we have automated 80% of our test cases.

8.2 USABILITY TESTING

Guerrilla usability testing to increase user understanding and engagement of existing 'Archive' and 'Collections' features.

Example Testing tasks included:

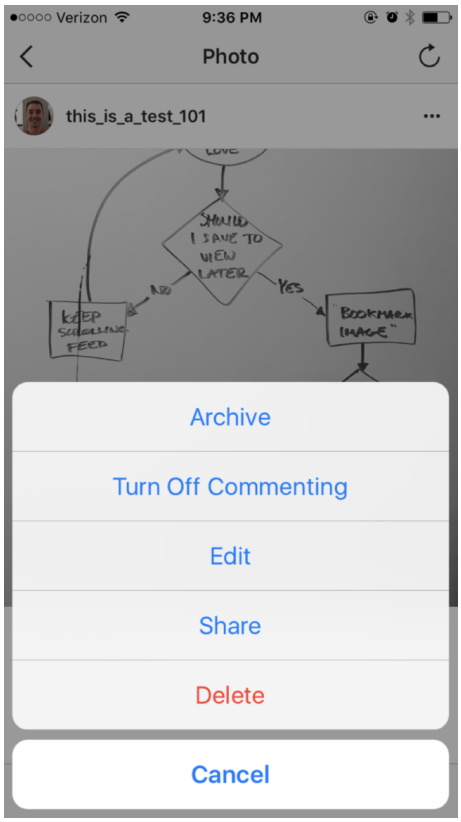
- Take a photo in app, geotag it, post it
- Search for a item (watch, shoes, sweater, etc...) and save a photo to view later
- Create a folder or group with a saved photo
- Share a saved photo
- Make a photo on your account no longer visible without deleting it

Solutions included:

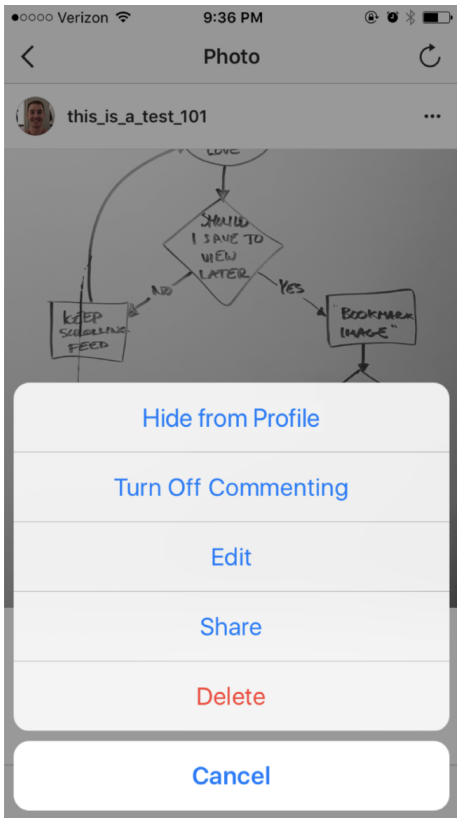
- Change 'Archive' to 'Hide from Profile'
- Incorporate existing notifications
- Incorporate existing micro interactions
- Group 'Hide' and 'Saved/Bookmarked' icons on Account screen
- IF 4. is validated, consider combining 'Archive' and 'Saved/Bookmarked' icons
- Redesign 'Archive' icon and validate

The following are side-by-side comparisons of Instagram before and after implementing design solutions:

1. Changed 'Archive' to 'Hide from Profile'

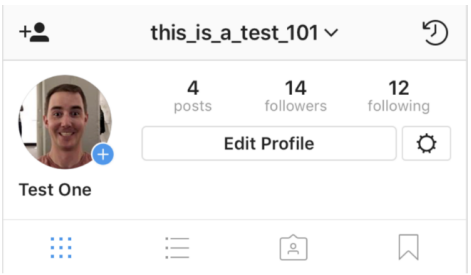


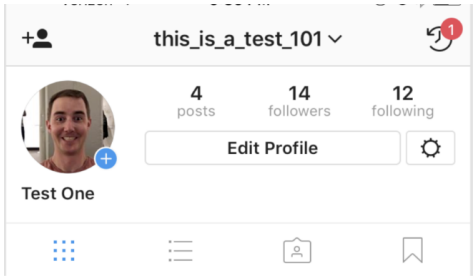
Before



After

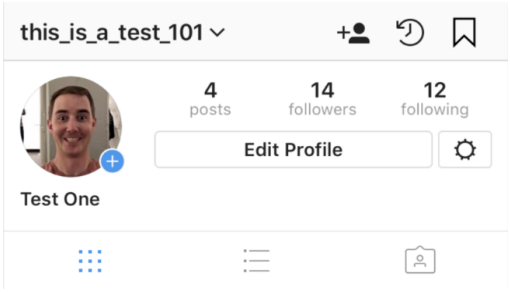
2. Incorporated existing notifications which will help users to locate 'Archive' icon





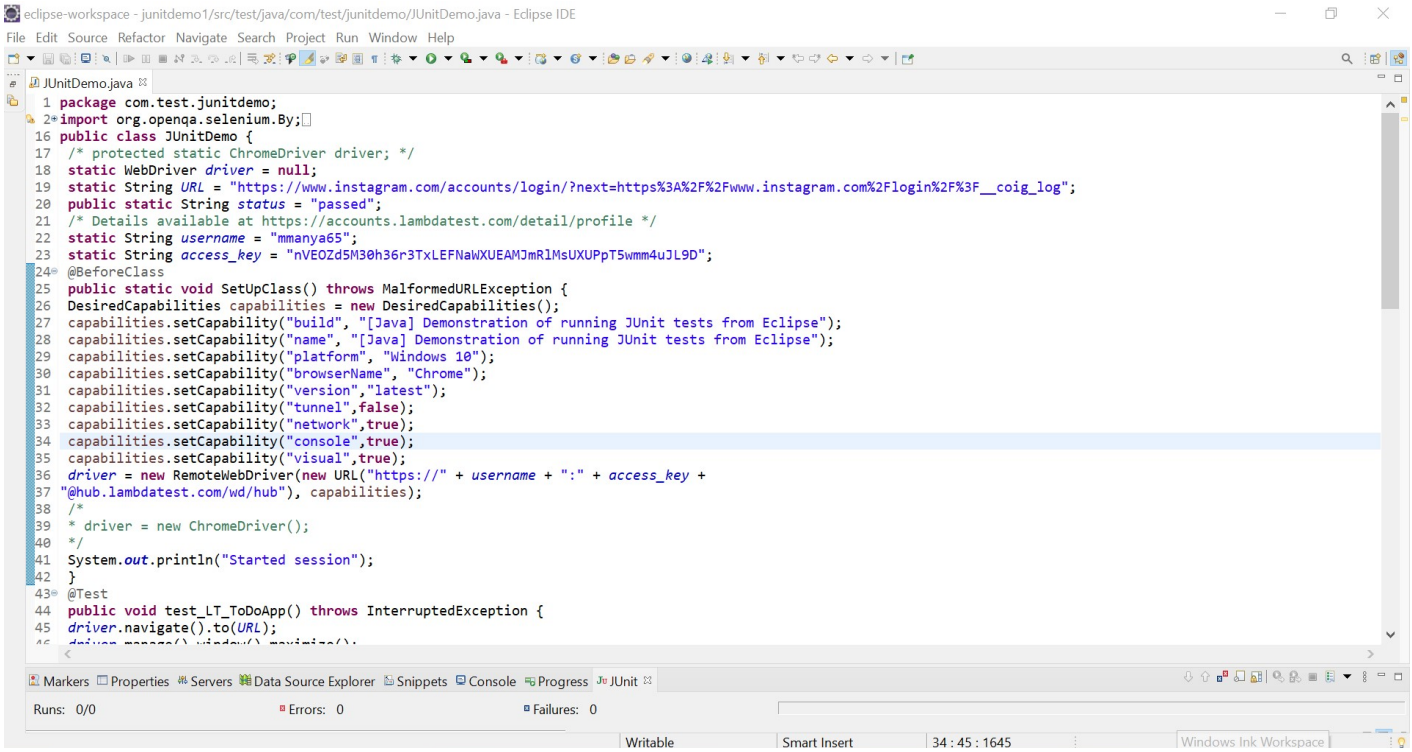
Before After

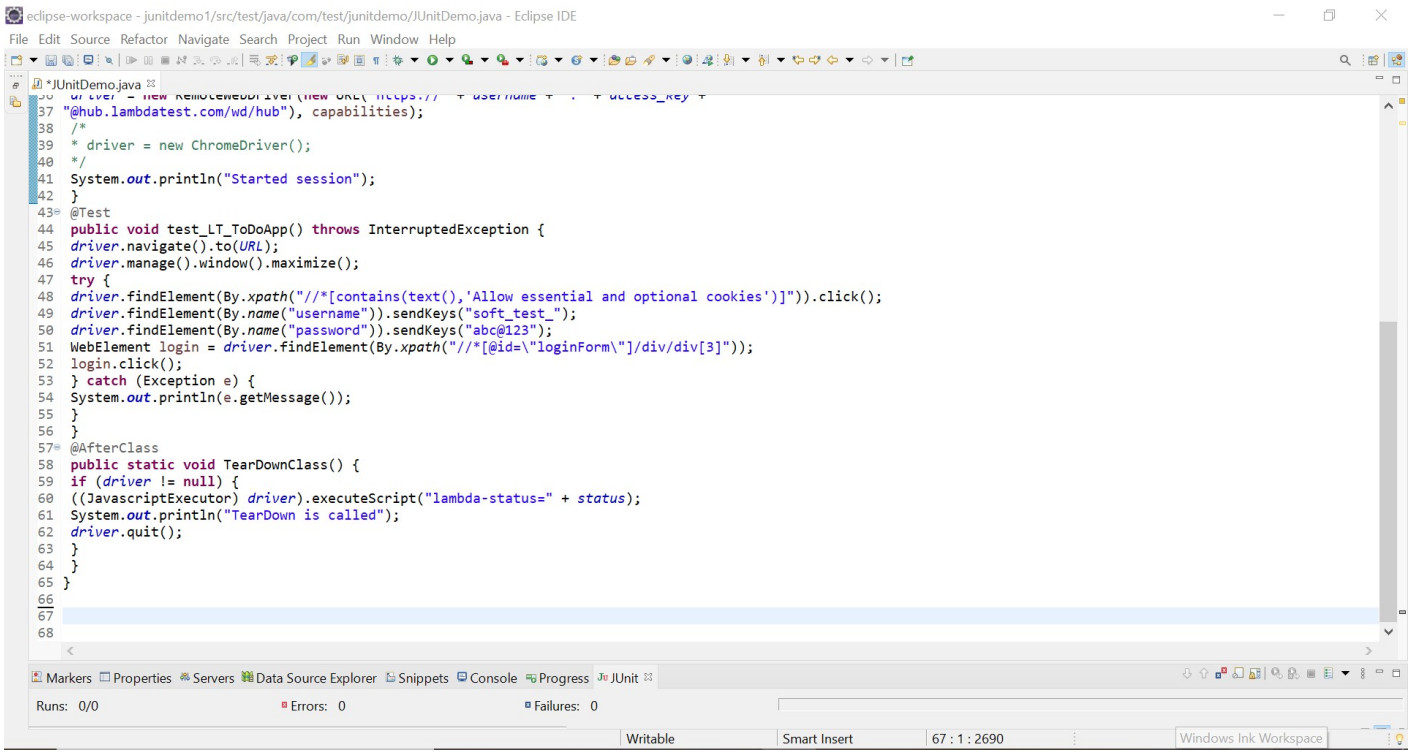
3. Grouped 'Archive' and 'Saved/Collections' icons in header navigation of Account screen

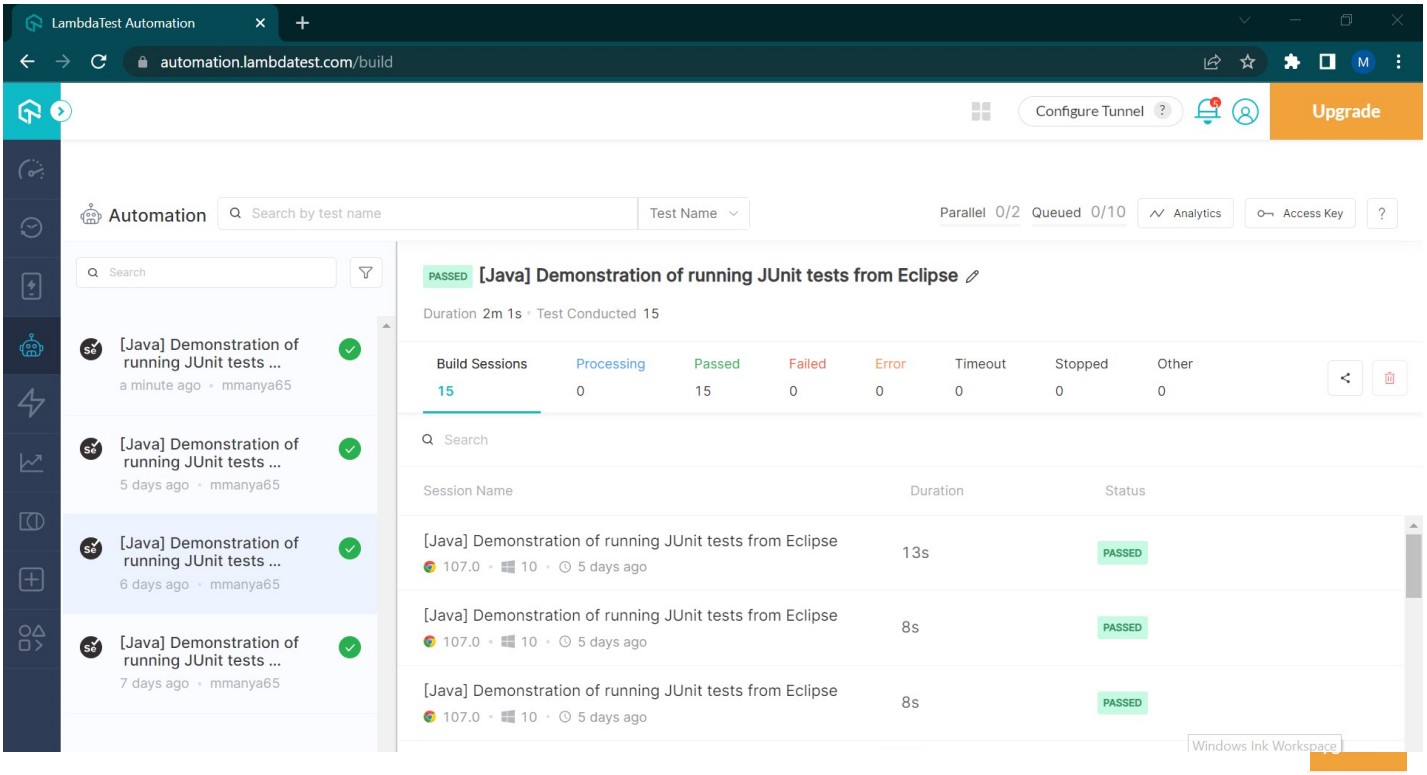


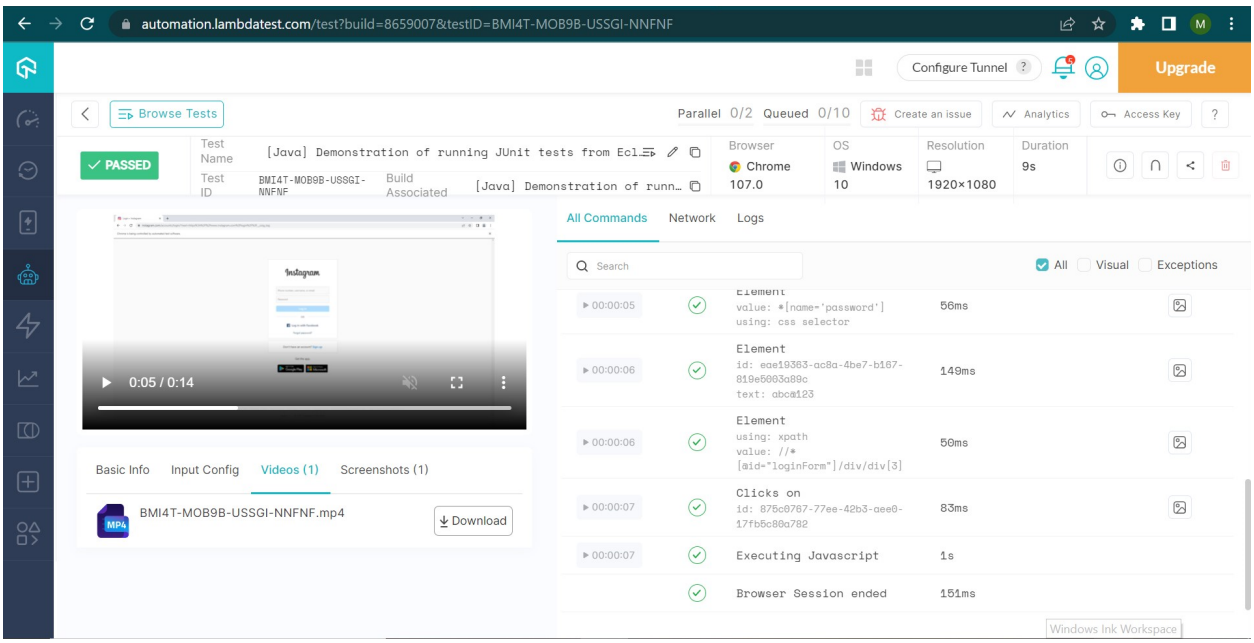
Before After

9. JUNIT TESTING






LambdaTest Automation
automation.lambdatest.com/build
Configure Tunnel
Upgrade
Automation
Search by test name
Test Name
Parallel 0/2 Queued 0/10
Analytics
Access Key
[Java] Demonstration of running JUnit tests ...
a minute ago • mmanya65
5 days ago • mmanya65
6 days ago • mmanya65
7 days ago • mmanya65
PASSED
[Java] Demonstration of running JUnit tests from Eclipse
Duration 2m 1s • Test Conducted 15
Build Sessions 15
Processing 0
Passed 15
Failed 0
Error 0
Timeout 0
Stopped 0
Other 0
Search
Session Name
Duration
Status
[Java] Demonstration of running JUnit tests from Eclipse
107.0 • 10 • 5 days ago
13s
8s
PASSED


automation.lambdatest.com/test?build=8659007&testID=BMI4T-MOB9B-USSGI-NNFNF
Configure Tunnel
Upgrade
Browse Tests
Parallel 0/2 Queued 0/10
Create an issue
Analytics
Access Key
PASSED
Test Name
[Java] Demonstration of running JUnit tests from Ecl...
Test ID
BMI4T-MOB9B-USSGI-NNFNF
Build Associated
[Java] Demonstration of runn...
Browser
Chrome 107.0
OS
Windows 10
Resolution
1920×1080
Duration
9s
0:05 / 0:14
All Commands
Network
Logs
Search
All
Visual
Exceptions
Element
value: *[name='password']
using: css selector
56ms
Element
id: eae19363-ac8a-4be7-b187-819e5003a89c
text: abcm123
149ms
Element
using: xpath
value: //*[@id="loginForm"]/div/div[3]
50ms
Clicks on
id: 875c0787-77ee-42b3-aee0-17fb5c80a782
83ms
Executing Javascript
1s
Browser Session ended
151ms
Basic Info
Input Config
Videos (1)
Screenshots (1)
BMI4T-MOB9B-USSGI-NNFNF.mp4
Download
MP4

10. TESTING TOOLS

10.1 AUTOMATION TESTING: SELENIUM

Selenium is an open-source umbrella project for a range of tools and libraries aimed at supporting browser automation. Without having to understand a test scripting language, it provides a playback tool for creating functional tests that work across the majority of current web browsers (Selenium IDE). Additionally, it gives users the ability to build tests in a variety of well-known programming languages, such as JavaScript (Node.js), C#, Groovy, Java, Perl, PHP, Python, Ruby, and Scala. Linux, macOS, and Windows all support Selenium. It is Apache License 2.0-licensed open-source software**.**

Code-

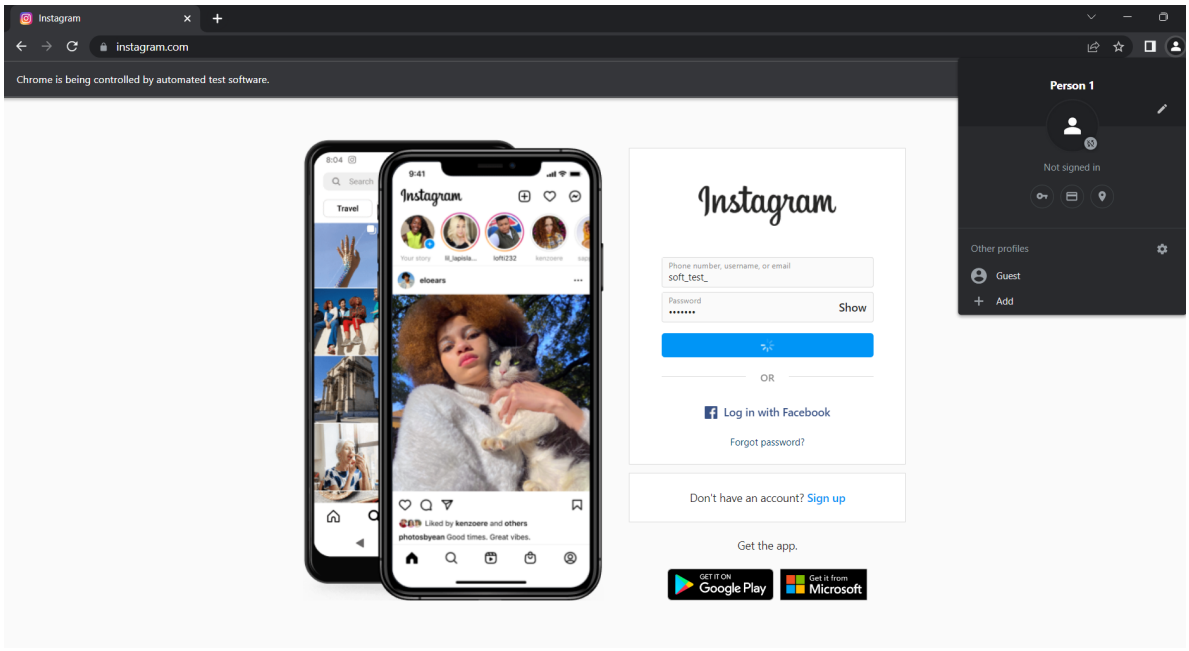
InstagramLogin.java | login.java

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.concurrent.TimeUnit;

public class InstagramLogin {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\karan\\Downloads\\chromedriver\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.instagram.com/");
        WebElement username=driver.findElement(By.name("username"));
        WebElement password=driver.findElement(By.name("password"));
        WebElement login=driver.findElement(By.xpath( xpathExpression: "//*[@id=\"loginForm\"]/div/div[3]/button"));
        username.sendKeys( ...keysToSend: "soft_test_");
        password.sendKeys( ...keysToSend: "abc@123");
        login.click();
        driver.manage().timeouts().implicitlyWait( time: 30, TimeUnit.SECONDS) ;
        driver.findElement(By.xpath( xpathExpression: "//*[contains(text(),'Not Now')]")).click();
        driver.findElement(By.xpath( xpathExpression: "//*[contains(text(),'Not Now')]")).click();
        driver.findElement(By.xpath( xpathExpression: "//*[contains(text(),'Follow')]")).click();
        dri                                                                              "//*[contains(text(),'Follow')]")).click();
        dri                                                                              "//*[contains(text(),'Follow')]")).click();
    }
}
```

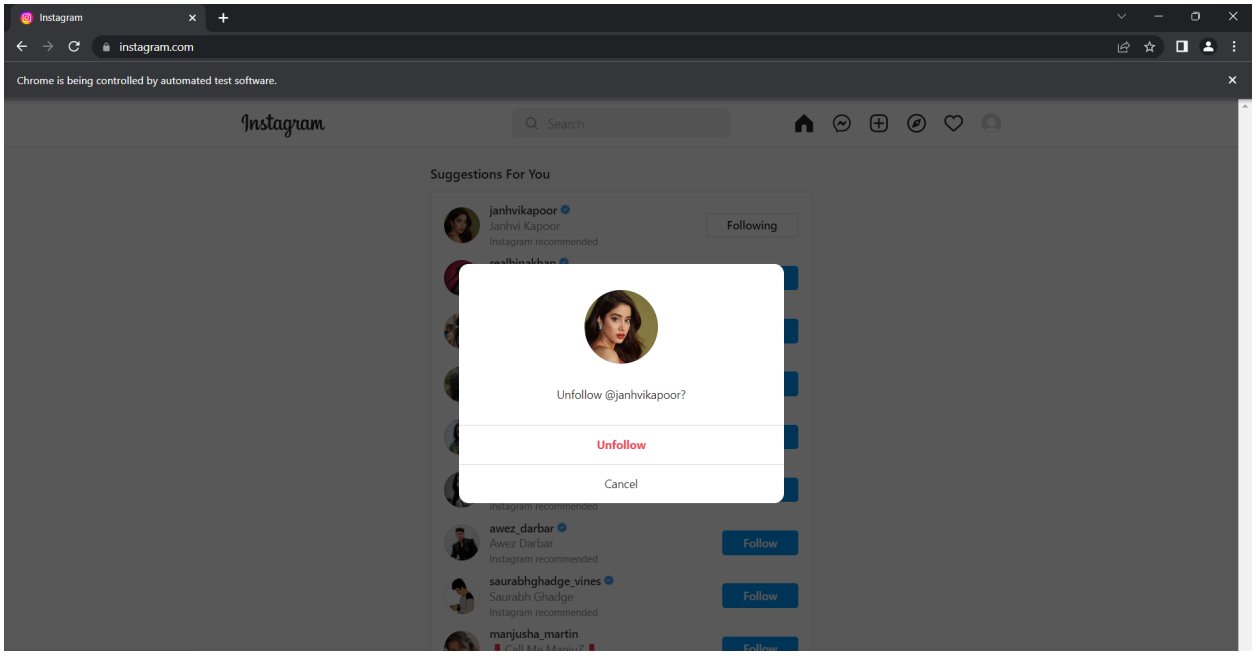
WebDriver driver = new ChromeDriver()
Junit

AUTOMATION

1)LOGIN FUNCTIONALITY.



2) FOLLOWING FUNCTIONALITY.

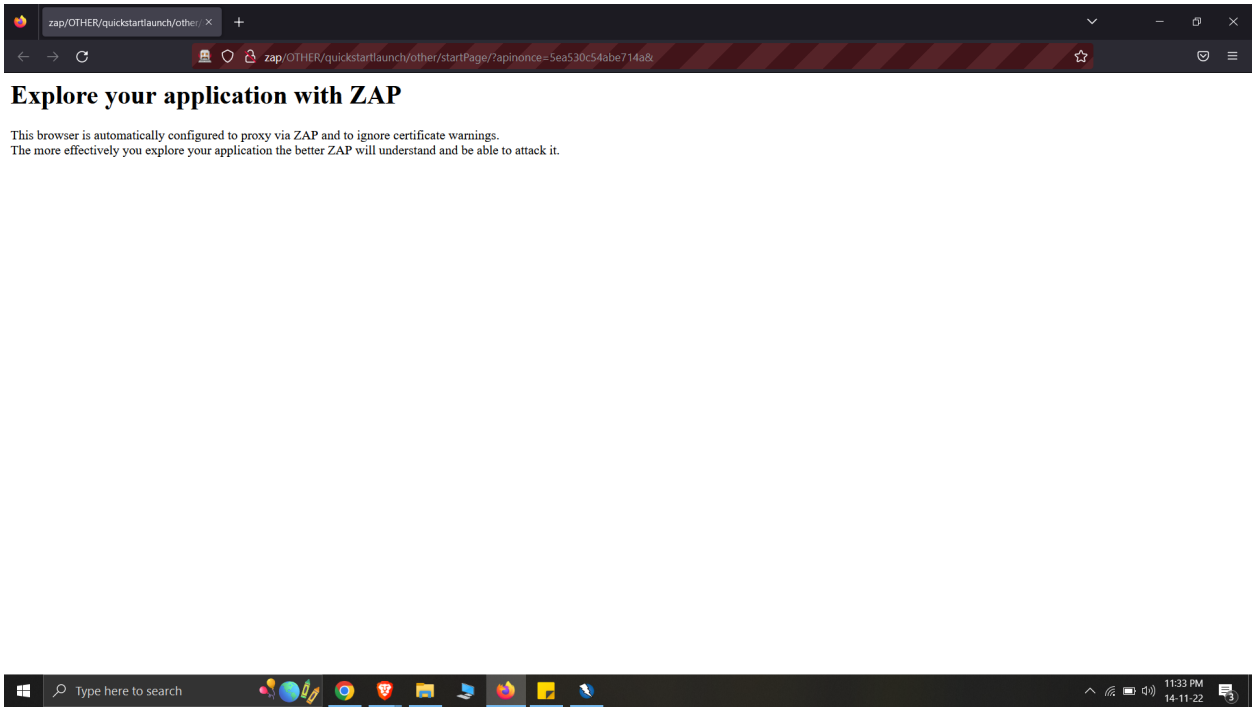


10.2 SECURITY TESTING: OWASP ZAP

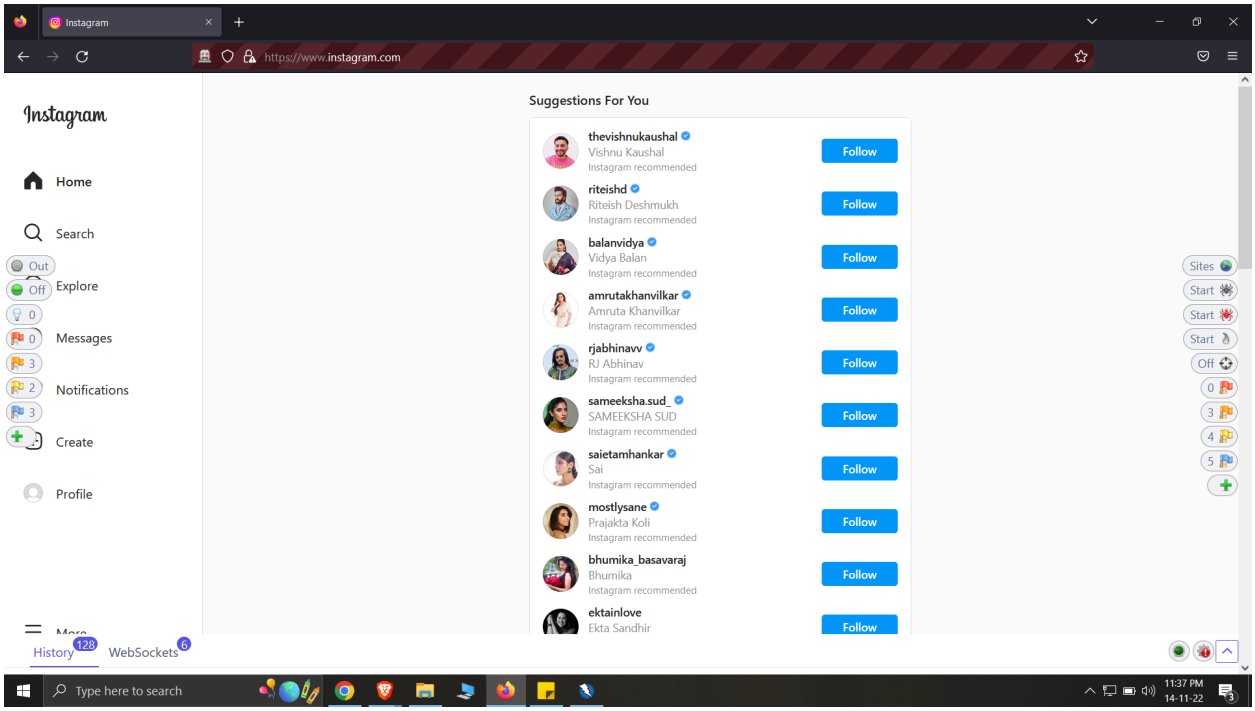
OWASP ZAP (short for Zed Attack Proxy) is an open-source web application security scanner. It is intended to be used by both those new to application security as well as professional penetration testers.
OWASP ZAP is a dynamic application security testing (DAST) tool for finding vulnerabilities in web applications.

Authenticated Scan: We are performing authenticated scan on instagram website by providing credentials to Zap and it will then crawl the website and attack every possible URL on the website after logging into the account.

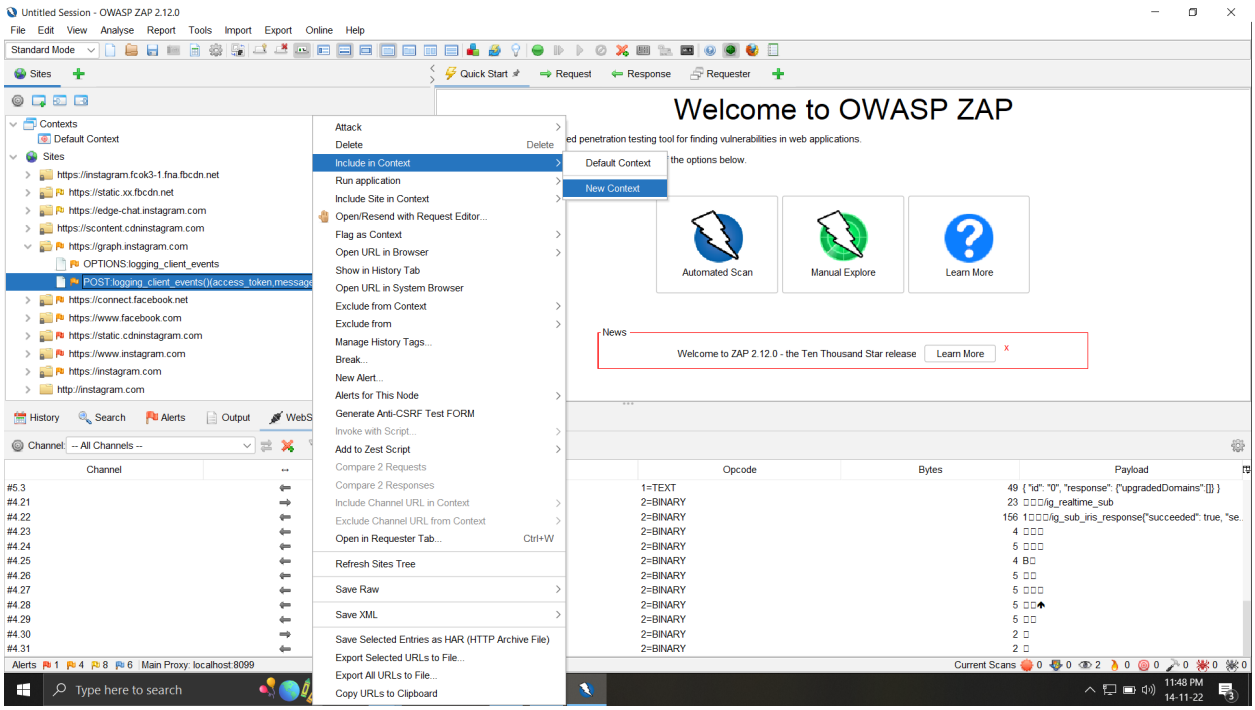
1) Launch the browser from Zap



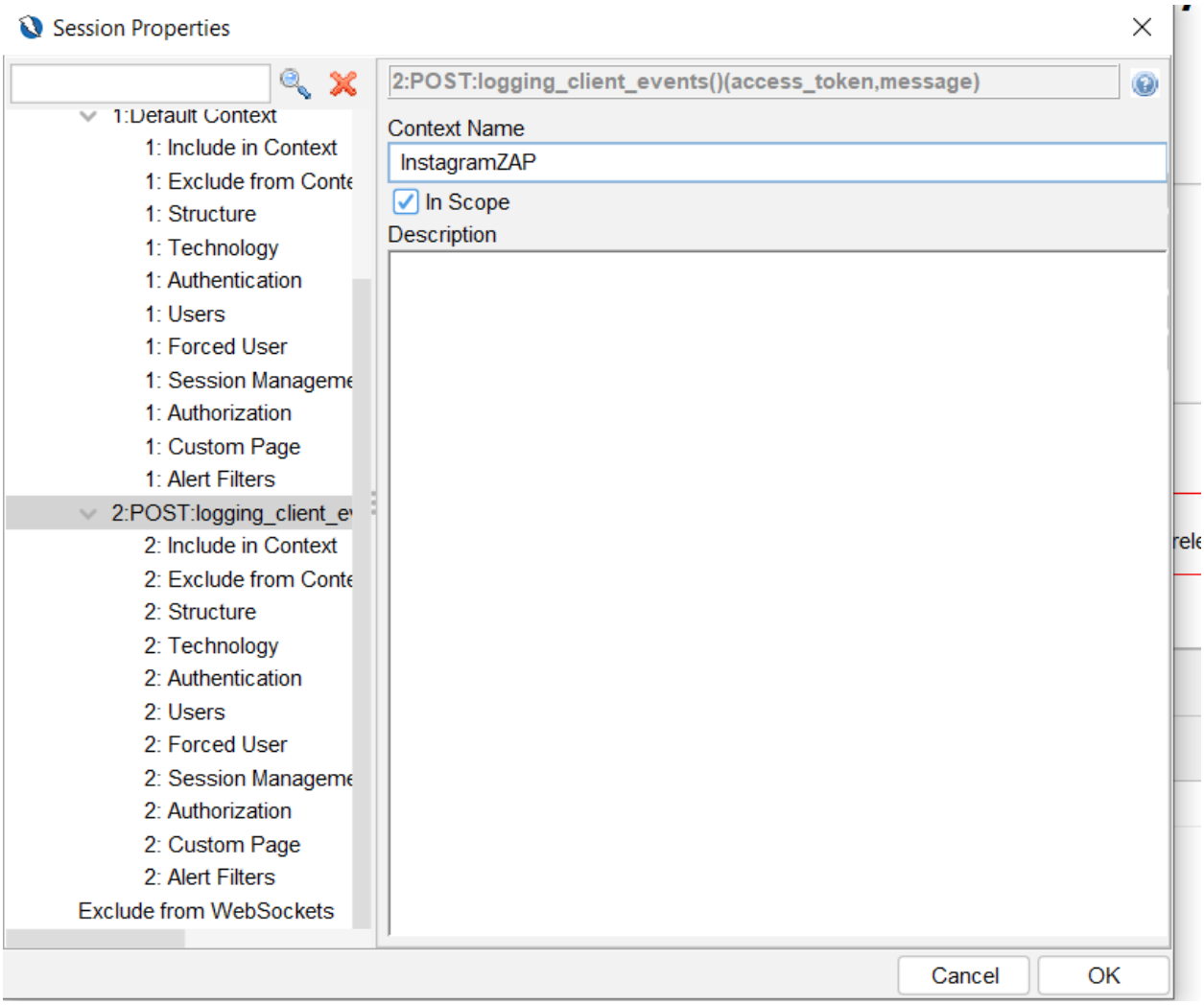
2) Go to www.instagram.com
And then login to your account.



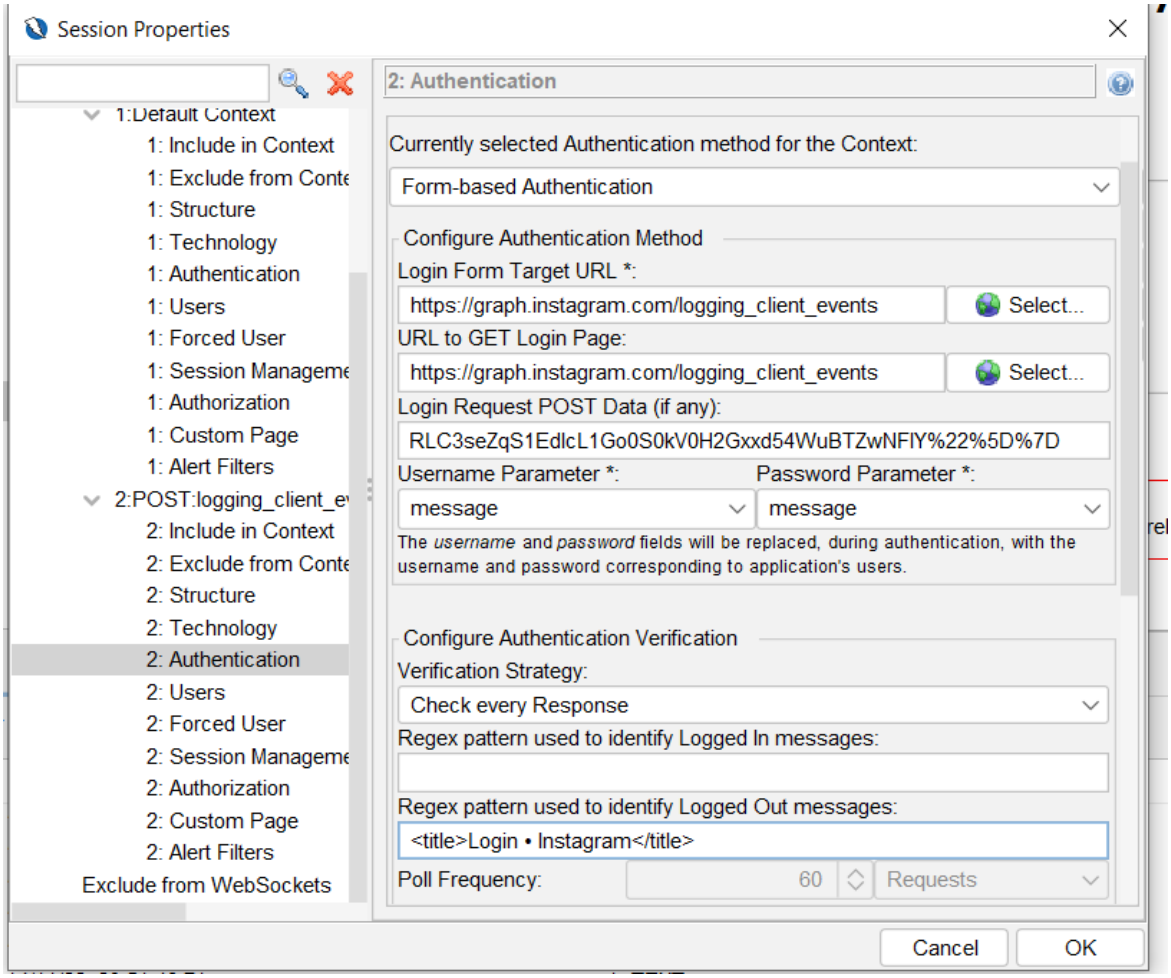
3) Create a New Context from the POST URL generated.



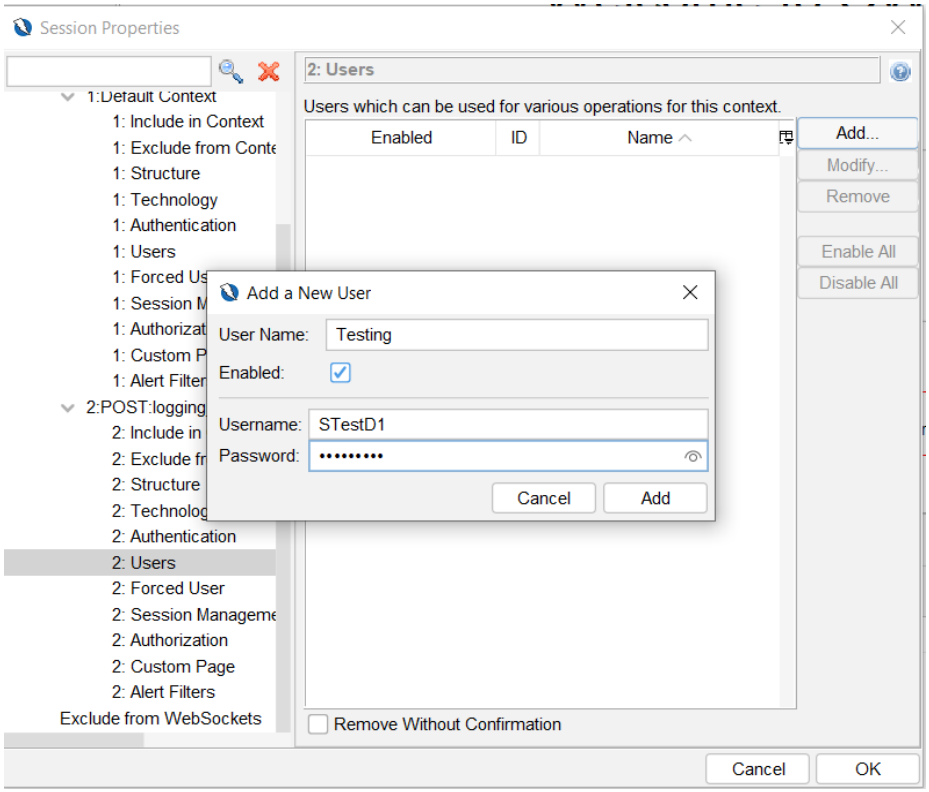
4) Give the Context a Name



5) In authentication, Choose Form Based Authentication. Then choose Login Form Target URL as the POST url generated earlier. Then choose username and password parameter as message. Set Logged out message as the title tag of the Instagram page after logging out.

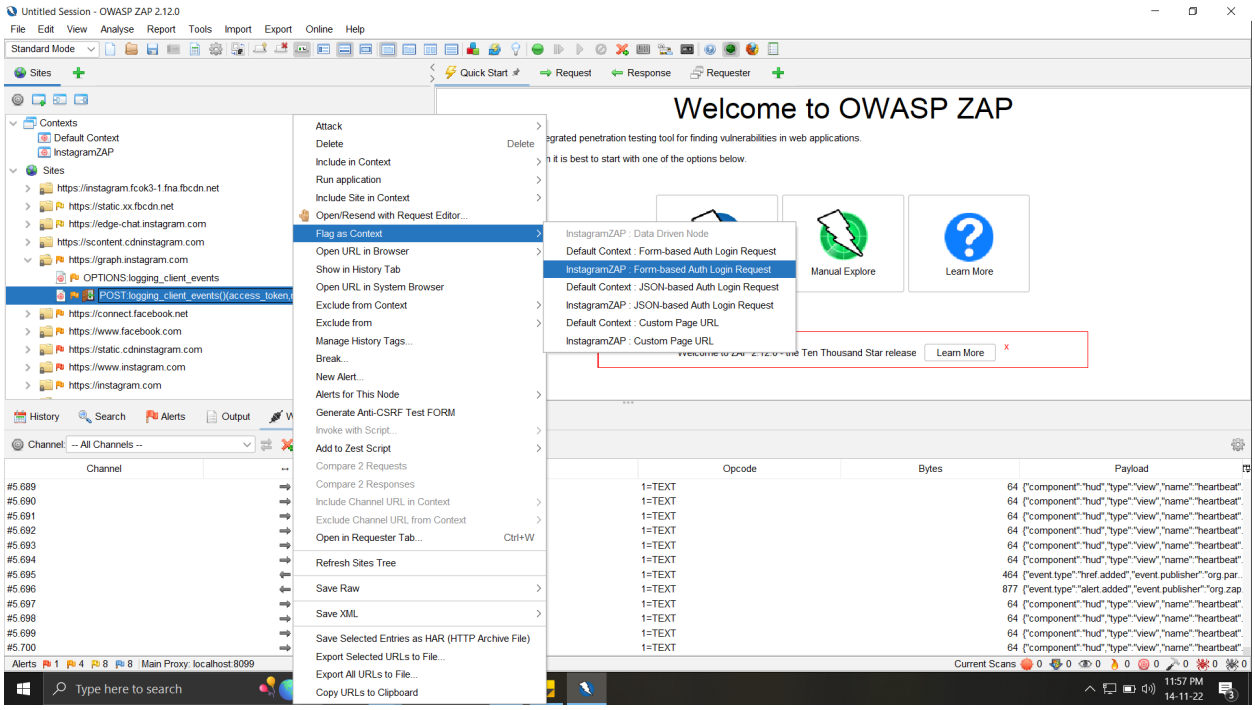

Session Properties
1:Default Context
1: Include in Context
1: Exclude from Conte
1: Structure
1: Technology
1: Authentication
1: Users
1: Forced User
1: Session Manageme
1: Authorization
1: Custom Page
1: Alert Filters
2:POST:logging_client_e
2: Include in Context
2: Exclude from Conte
2: Structure
2: Technology
2: Authentication
2: Users
2: Forced User
2: Session Manageme
2: Authorization
2: Custom Page
2: Alert Filters
Exclude from WebSockets
2: Authentication
Currently selected Authentication method for the Context:
Form-based Authentication
Configure Authentication Method
Login Form Target URL *:
https://graph.instagram.com/logging_client_events
Select...
URL to GET Login Page:
https://graph.instagram.com/logging_client_events
Select...
Login Request POST Data (if any):
RLC3seZqS1EdlcL1Go0S0kV0H2Gxxd54WuBTZwNFIY%22%5D%7D
Username Parameter *:
Password Parameter *:
message
message
The username and password fields will be replaced, during authentication, with the username and password corresponding to application's users.
Configure Authentication Verification
Verification Strategy:
Check every Response
Regex pattern used to identify Logged In messages:
Regex pattern used to identify Logged Out messages:
<title>Login • Instagram</title>
Poll Frequency:
60
Requests
Cancel
OK

6) In users, click on add user and enter the credentials.



And then click add .

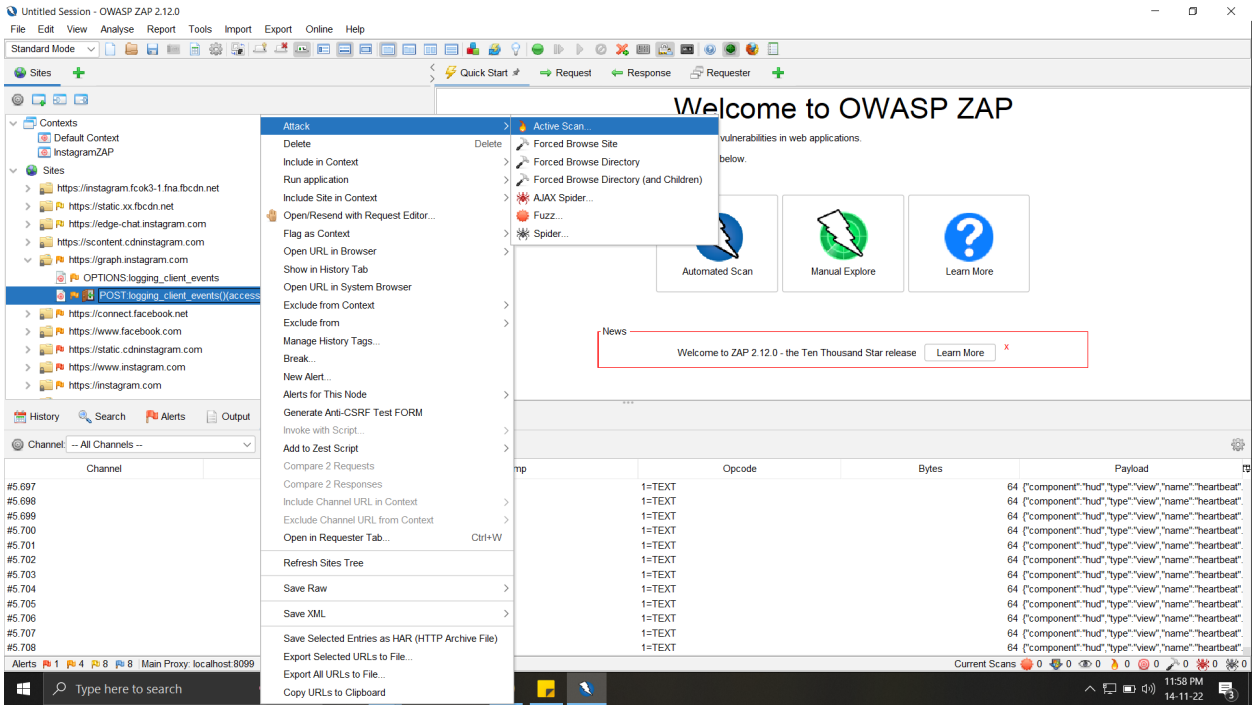
7) Now flag this context on the Zap Homepage



And click OK.

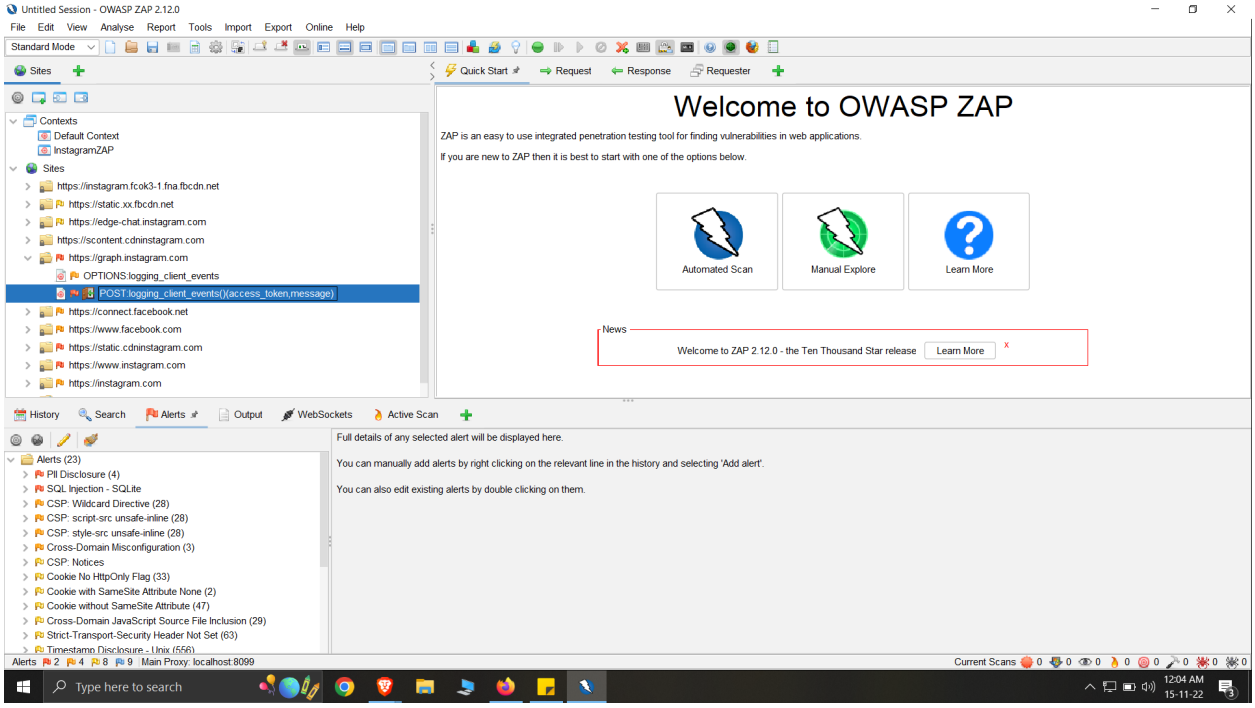
8) Turn on Forced User Mode.

9) Start the attack

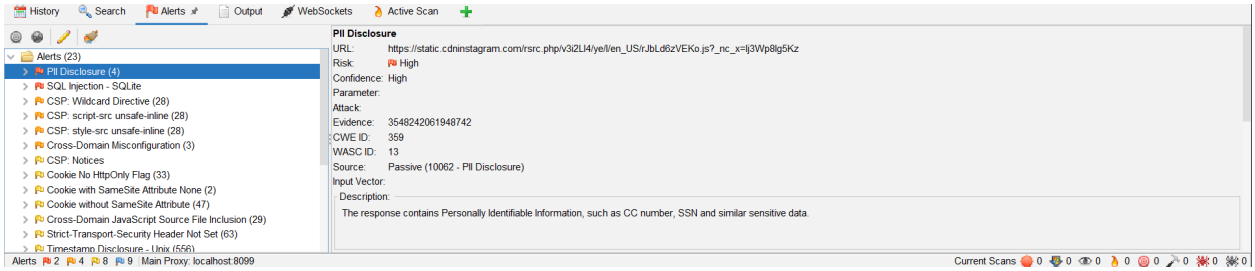


10) Now we view the results

We can see the alerts.



The alerts in Red are High risk alerts.



The confidence is an indicator as to how sure ZAP is in the fact that it is an alert.

10.3 PERFORMANCE TESTING: APACHE JMETER

The **Apache JMeter** application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications. It may be used to test performance both on static and dynamic resources, Web dynamic applications. It can be used to simulate a heavy load on a server, group of servers, network or object to test its strength or to analyze overall performance under different load types.

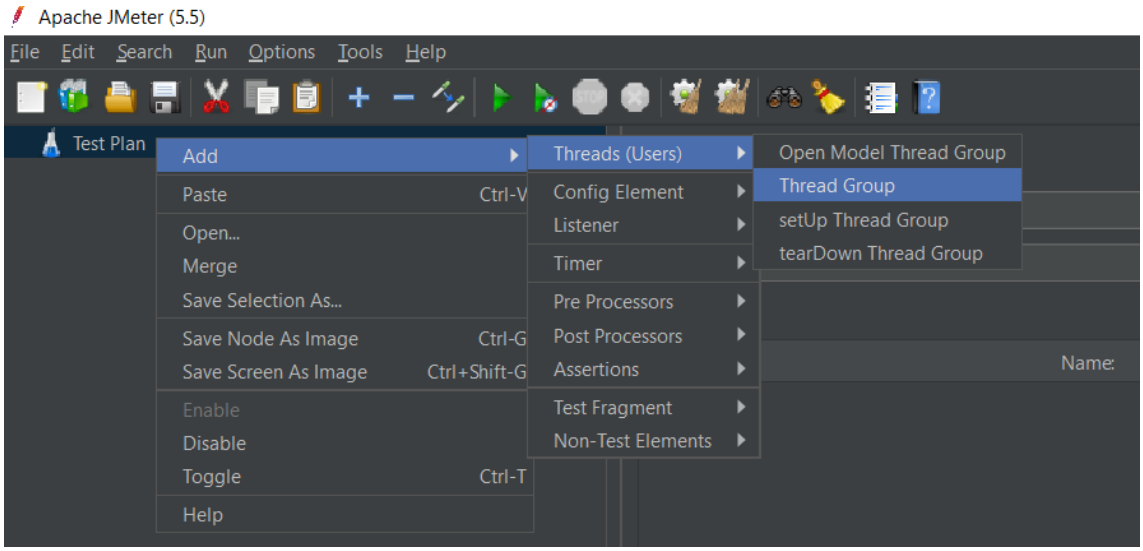
Apache JMeter features include:

- Ability to load and performance test many different applications/server/protocol types
- Full featured Test IDE that allows fast Test Plan recording (from Browsers or native applications), building and debugging.
- CLI mode to load test from any Java compatible OS.
- A complete and ready to present dynamic HTML report
- Easy correlation through ability to extract data from most popular response formats, HTML, JSON , XML or any textual format
- Complete portability and 100% Java purity.

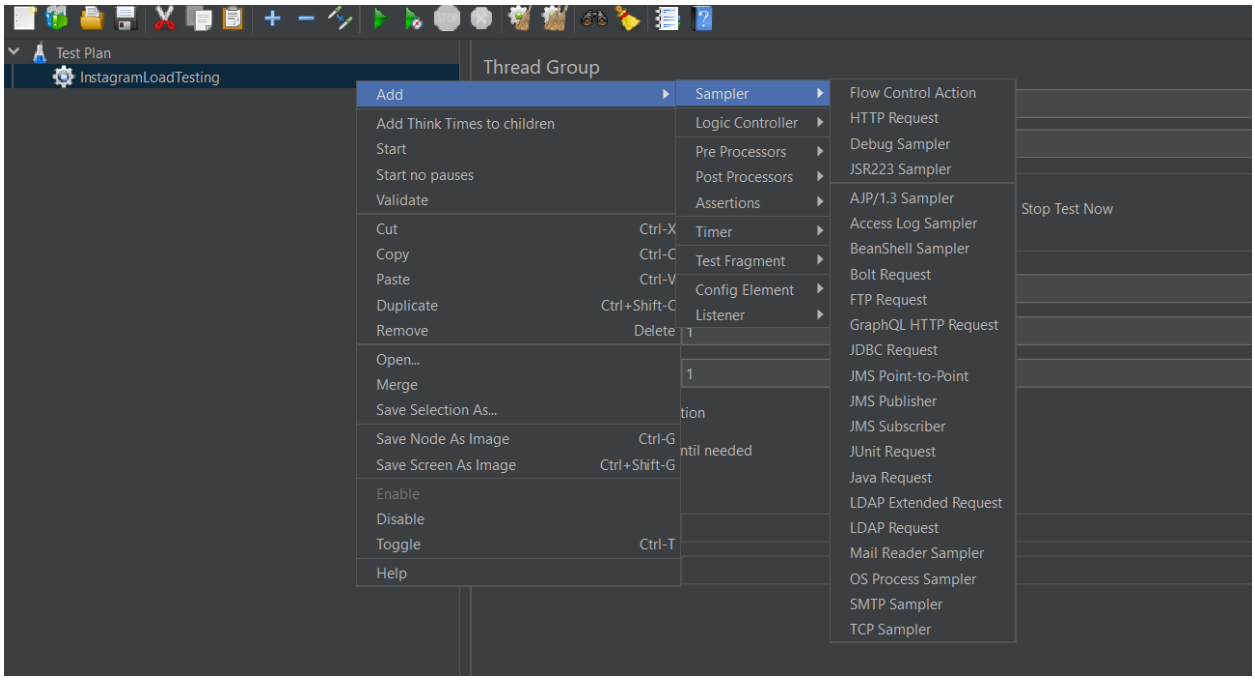
- Full multi-threading framework allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.
- Caching and offline analysis/replaying of test results

We are performing Load Testing on the Instagram Website using Jmeter.

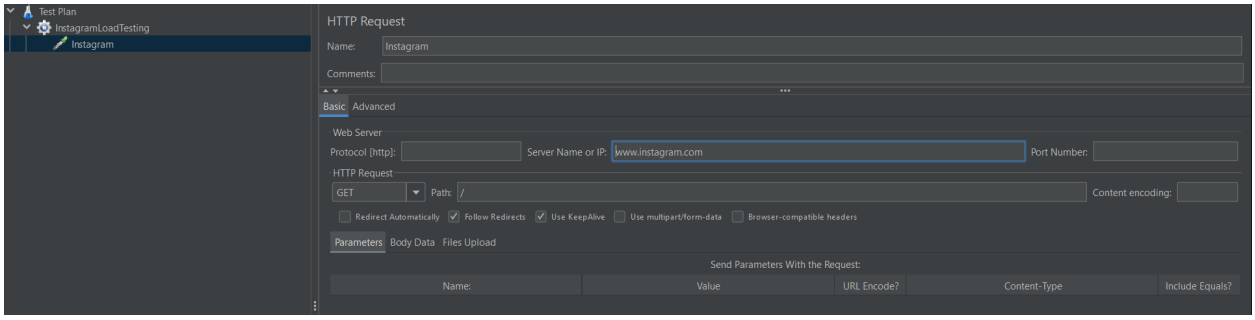
1) Open apache Jmeter in GUI.
2) Then add a new Thread Group and name it InstagramLoadTesting



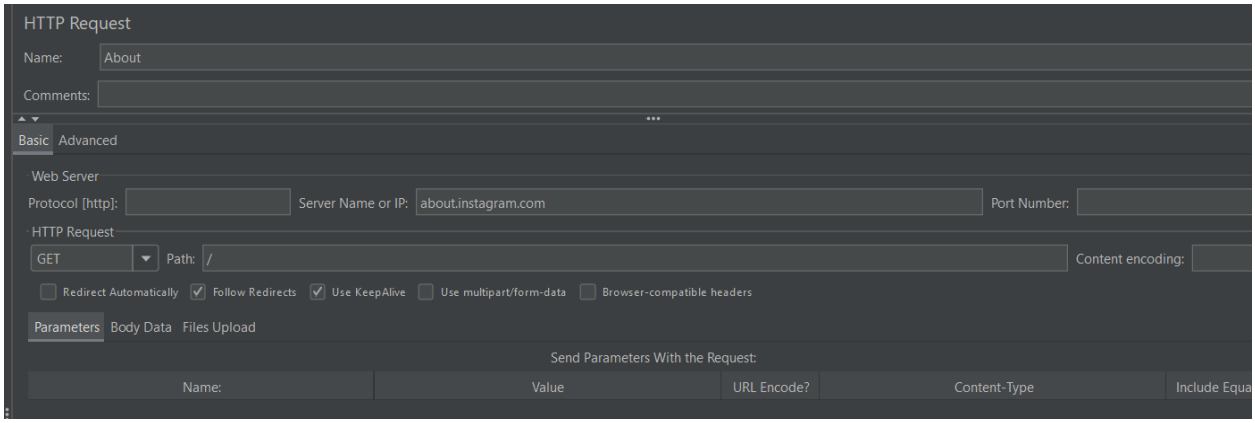
3) Then add an HTTP request and name it Instagram

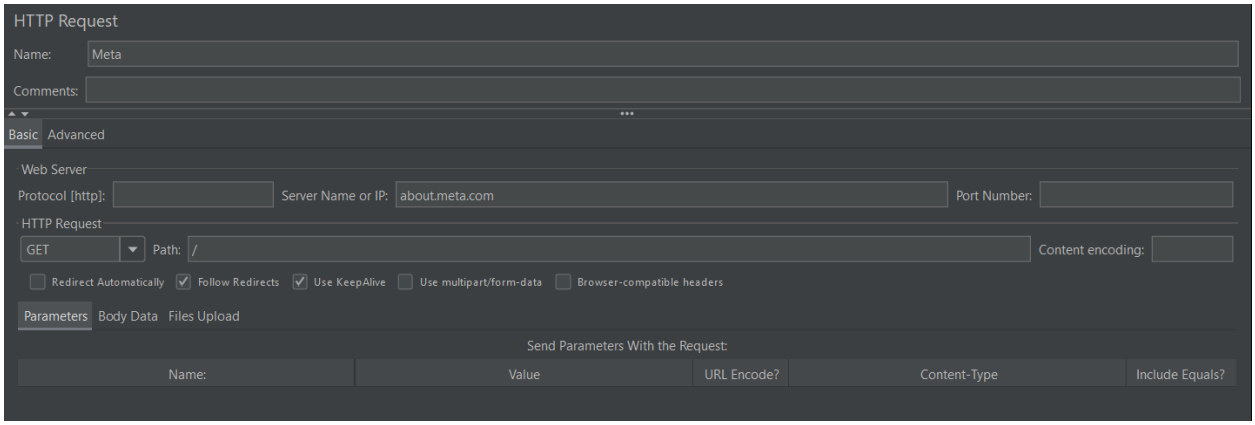


4) Set server name as www.instagram.com and path



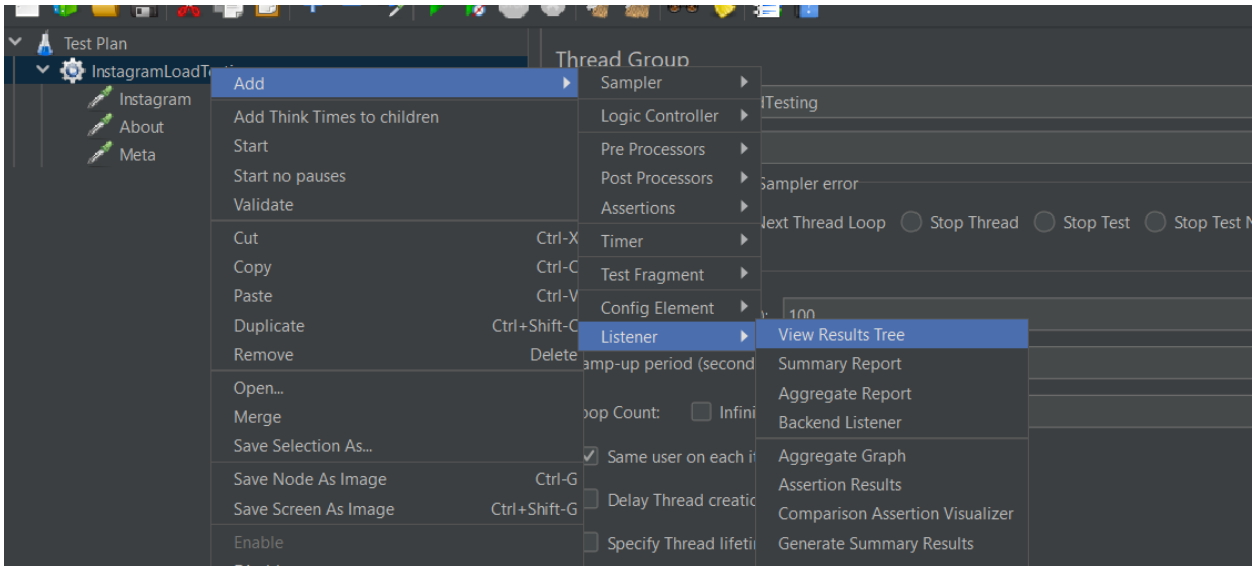
5) Add 2 more listeners from the instagram HomePage. Here we are adding the About and Meta Pages.





6) Now we set the number of users to 100 and ramp up period as 5 seconds on the Thread Group Page

7) Now we add listeners to see the output.

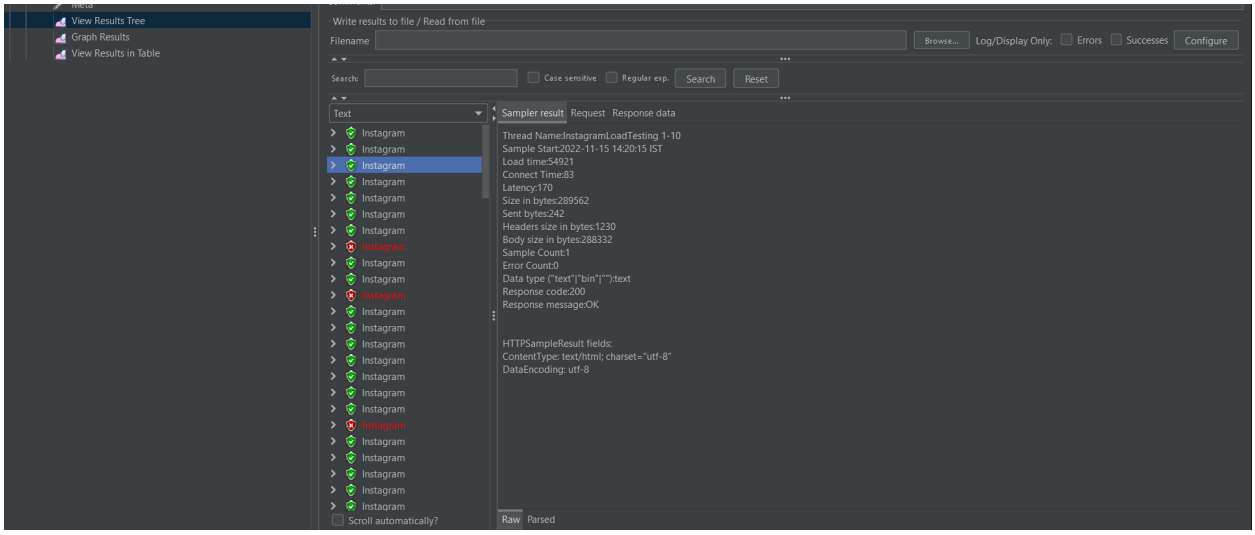


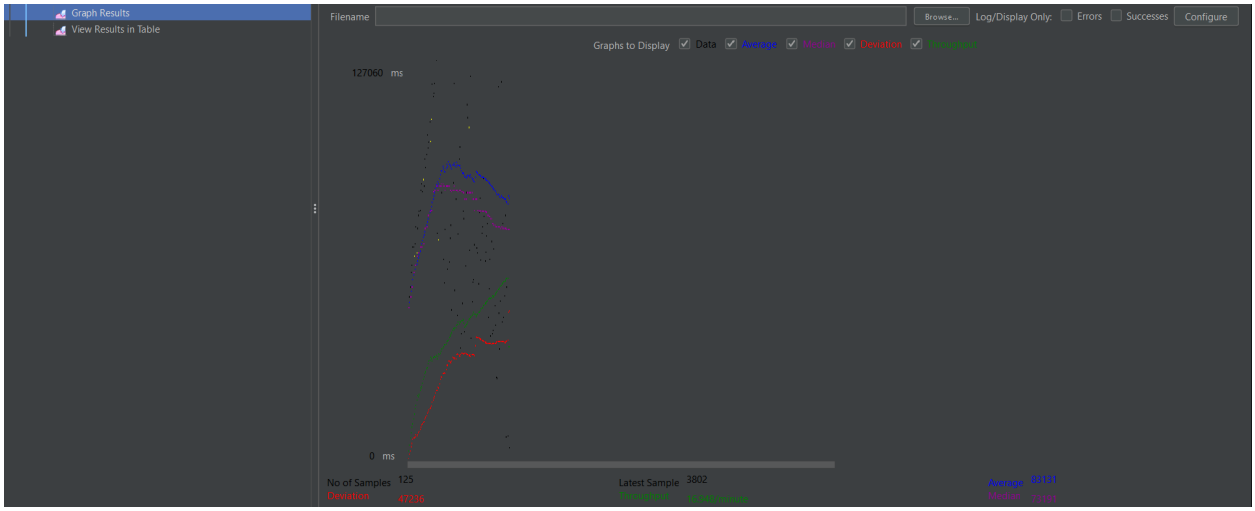
Add results tree, graph results and results in a table.

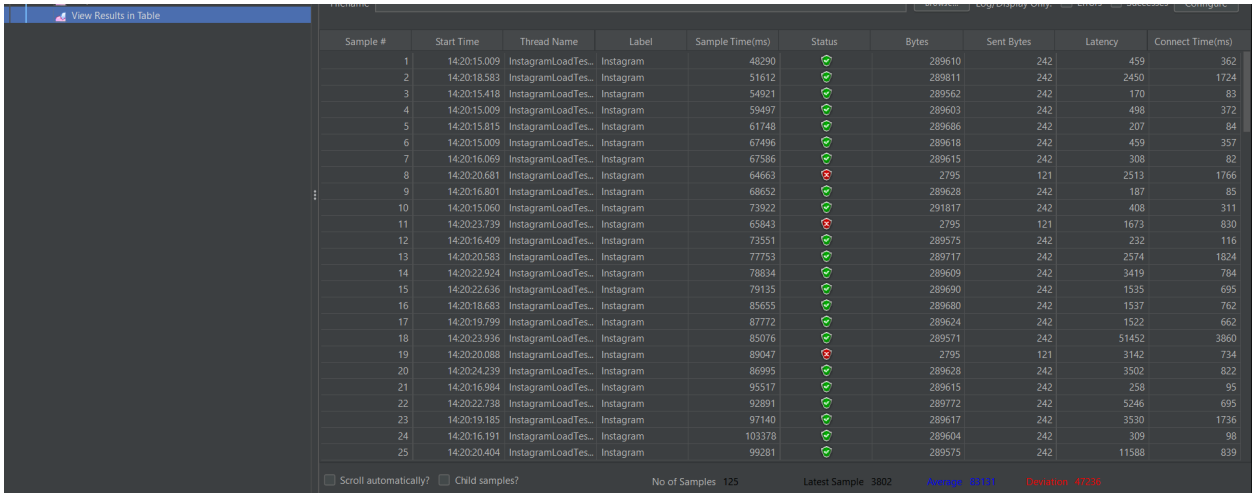
8) Save the file.

9) Now run the program.

10) Now we view the results.


View Results Tree
Graph Results
View Results in Table
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Search:
Case sensitive
Regular exp.
Search
Reset
Text
Sampler result
Request
Response data
Instagram
Thread Name:InstagramLoadTesting 1-10
Sample Start:2022-11-15 14:20:15 IST
Load time:54921
Connect Time:83
Latency:170
Size in bytes:289562
Sent bytes:242
Headers size in bytes:1230
Body size in bytes:288332
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK
HTTPSampleResult fields:
ContentType: text/html; charset="utf-8"
DataEncoding: utf-8
Scroll automatically?
Raw
Parsed


Graph Results
View Results in Table
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Graphs to Display
Data
Average
Median
Deviation
Throughput
127060 ms
0 ms
No of Samples 125
Latest Sample 3802
Deviation 47236
Average 83131
Median 73169


View Results in Table
Sample # | Start Time | Thread Name | Label | Sample Time(ms) | Status | Bytes | Sent Bytes | Latency | Connect Time(ms)
1 | 14:20:15.009 | InstagramLoadTes... | Instagram | 48290 | | 289610 | 242 | 459 | 362
2 | 14:20:18.583 | InstagramLoadTes... | Instagram | 51612 | | 289811 | 242 | 2450 | 1724
3 | 14:20:15.418 | InstagramLoadTes... | Instagram | 54921 | | 289562 | 242 | 170 | 83
4 | 14:20:15.009 | InstagramLoadTes... | Instagram | 59497 | | 289603 | 242 | 498 | 372
5 | 14:20:15.815 | InstagramLoadTes... | Instagram | 61748 | | 289686 | 242 | 207 | 84
6 | 14:20:15.009 | InstagramLoadTes... | Instagram | 67496 | | 289618 | 242 | 459 | 357
7 | 14:20:16.069 | InstagramLoadTes... | Instagram | 67586 | | 289615 | 242 | 308 | 82
8 | 14:20:20.681 | InstagramLoadTes... | Instagram | 64663 | | 2795 | 121 | 2513 | 1766
9 | 14:20:16.801 | InstagramLoadTes... | Instagram | 68652 | | 289628 | 242 | 187 | 85
10 | 14:20:15.060 | InstagramLoadTes... | Instagram | 73922 | | 291817 | 242 | 408 | 311
11 | 14:20:23.739 | InstagramLoadTes... | Instagram | 65843 | | 2795 | 121 | 1673 | 830
12 | 14:20:16.409 | InstagramLoadTes... | Instagram | 73551 | | 289575 | 242 | 232 | 116
13 | 14:20:20.583 | InstagramLoadTes... | Instagram | 77753 | | 289717 | 242 | 2574 | 1824
14 | 14:20:22.924 | InstagramLoadTes... | Instagram | 78834 | | 289609 | 242 | 3419 | 784
15 | 14:20:22.636 | InstagramLoadTes... | Instagram | 79135 | | 289690 | 242 | 1535 | 695
16 | 14:20:18.683 | InstagramLoadTes... | Instagram | 85655 | | 289680 | 242 | 1537 | 762
17 | 14:20:19.799 | InstagramLoadTes... | Instagram | 87772 | | 289624 | 242 | 1522 | 662
18 | 14:20:23.936 | InstagramLoadTes... | Instagram | 85076 | | 289571 | 242 | 51452 | 3860
19 | 14:20:20.088 | InstagramLoadTes... | Instagram | 89047 | | 2795 | 121 | 3142 | 734
20 | 14:20:24.239 | InstagramLoadTes... | Instagram | 86995 | | 289628 | 242 | 3502 | 822
21 | 14:20:16.984 | InstagramLoadTes... | Instagram | 95517 | | 289615 | 242 | 258 | 95
22 | 14:20:22.738 | InstagramLoadTes... | Instagram | 92891 | | 289772 | 242 | 5246 | 695
23 | 14:20:19.185 | InstagramLoadTes... | Instagram | 97140 | | 289617 | 242 | 3530 | 1736
24 | 14:20:16.191 | InstagramLoadTes... | Instagram | 103378 | | 289604 | 242 | 309 | 98
25 | 14:20:20.404 | InstagramLoadTes... | Instagram | 99281 | | 289575 | 242 | 11588 | 839
Scroll automatically?
Child samples?
No of Samples 125
Latest Sample 3802
Average 83131
Deviation 47236

11. TEST EXECUTION CLOSURE PREPARATION

Purpose of the document

This document explains the various activities performed as part of the Testing of Instagram application**.**

Application Overview

Instagram is a photo and video-sharing social networking service owned by the American company Meta Platforms. The app allows users to upload media that can be edited with filters and organized by hashtags and geographical tagging. Posts can be shared publicly or with pre-approved followers.

Testing Scope

In-Scope: Functional Testing for the following modules is in the Scope of Testing

1. Login
2. Follow

Out of Scope: Integration Testing was not done for this application

Items not tested:

1. DMs Functionality

Metrics

- No. of test cases planned vs executed
- No. of test cases passed/failed

Test cases planned	Test cases executed	Test cases passed	Test cases failed
10	10	10	0

No of defects identified and their Status & Severity

	Critical	Major	Medium	Cosmetic	Total
Closed	**4**	**7**	**7**	**5**	**23**
Open	**0**	**0**	**0**	**0**	**0**

Types of Testing performed

1. Requirements-Based Testing
2. Usability Testing
3. JUnit Testing
4. Automation Testing: Selenium
5. Security Testing: OWASP Zap
6. Performance Testing: JMeter

Test Environment & Tools

Application URL	https://www.instagram.com/
Apps Server	157.240.208.174
Database	PostgreSQL