

ASCENDION

Capstone Project

Personal BACKGROUND

Name : Madhan Adithya KR

Past Experience : Technically NO,

But I have some Knowledge on React, Node.js, Large Language Models (LLMs), FastAPI, MongoDB, and PostgreSQL.

Qualification : B.E Computer Science and Engineering

Career Summary :

I started my career at Ascendion in July 2025. I was initially trained on the MERN stack. Later, I was assigned to the Citi Project, where I gained experience with FastAPI, MongoDB, and GenAI tools like LangGraph, LangChain, and various Large Language Models.



ASCENDION

Key Takeaways/Learnings from the Program (HTD)

Strong Testing Foundation – Gained hands-on experience in manual, automation, functional, and non-functional testing.

Real-World Agile Practices – Applied Agile, Scrum, and Jira effectively for end-to-end test lifecycle management.

Automation Skills – Built automation frameworks using Selenium (Java & Python) using maven, Page Object Model, logs, reports and executed scripts with JUnit, TestNG & PyTest.

API & Mobile Testing Exposure – Practiced API testing with Postman, Newman and mobile automation using Appium.

DevOps & CI/CD Integration – Gained knowledge on Integration using the tools like Git, Jenkins.

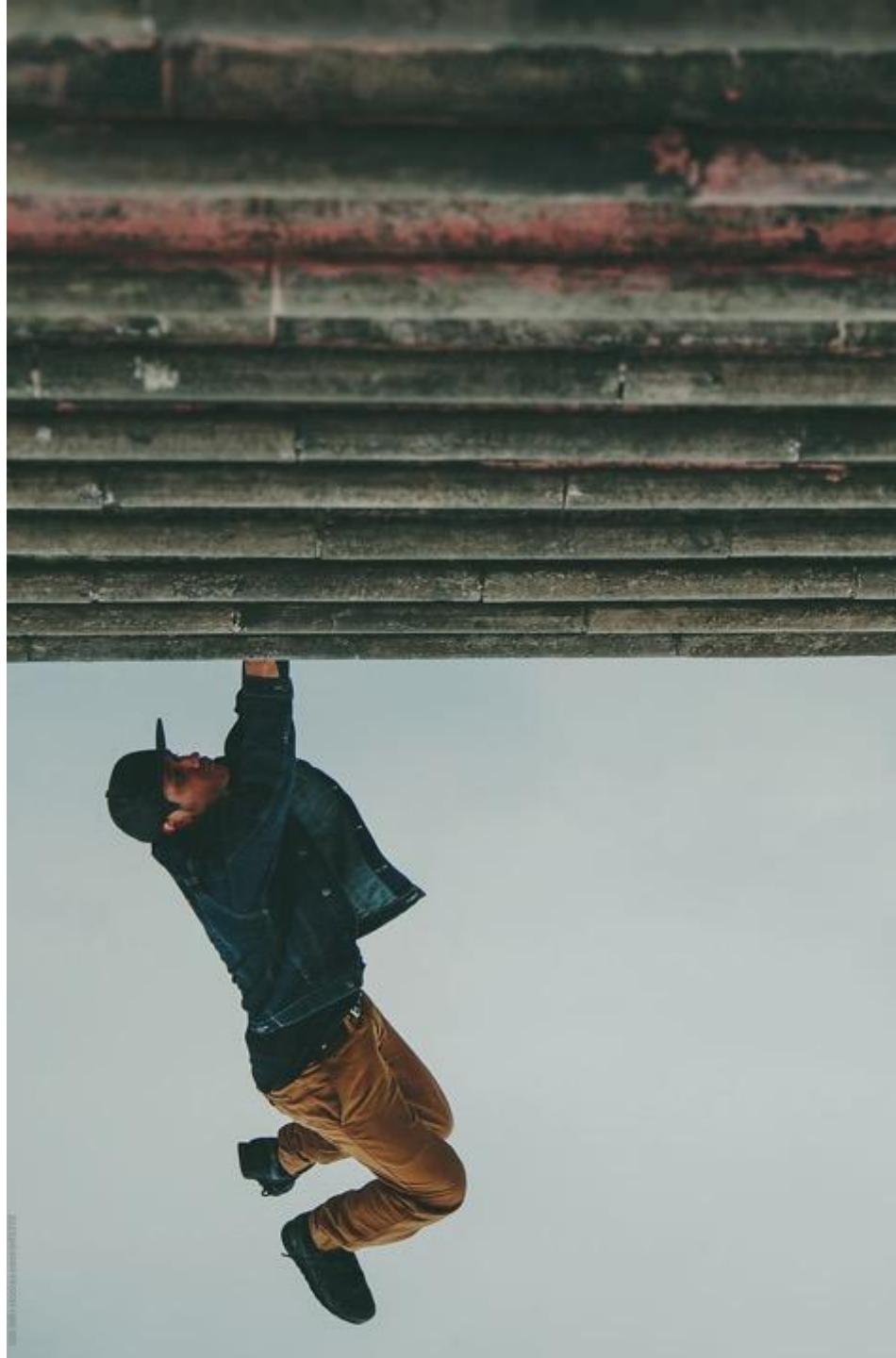
Performance Testing – Learned essentials of JMeter like making CRUD requests, samplers, listeners, config Elements and creating reports.



ASCENDION

Problem Statement of the Capstone Project

- Automate and validate key workflows for the TakeuForward application across UI, API, and performance layers.
- Implement Selenium Java automation for user scenarios, API testing with Postman/Newman, and performance testing using JMeter.
- Integrate test execution with Jenkins CI and maintain code in GitHub for version control.
- Use Jira with Xray plugin for comprehensive test management, reporting, and sprint tracking.



Reference Links

GitHub Repository: [madhanadithya/madhanadithya-Capstone-hp-training-may2025](https://github.com/madhanadithya/madhanadithya-Capstone-hp-training-may2025)

Jira project Link: [Sprint 2 - Api testing - HCM board - Scrum Board - Jira](#)

Web-Application: <https://takeuforward.org/>

API-Endpoint: Local Host using Json-server

Base url : <http://localhost:3000>

End-points: /users, /products, /orders

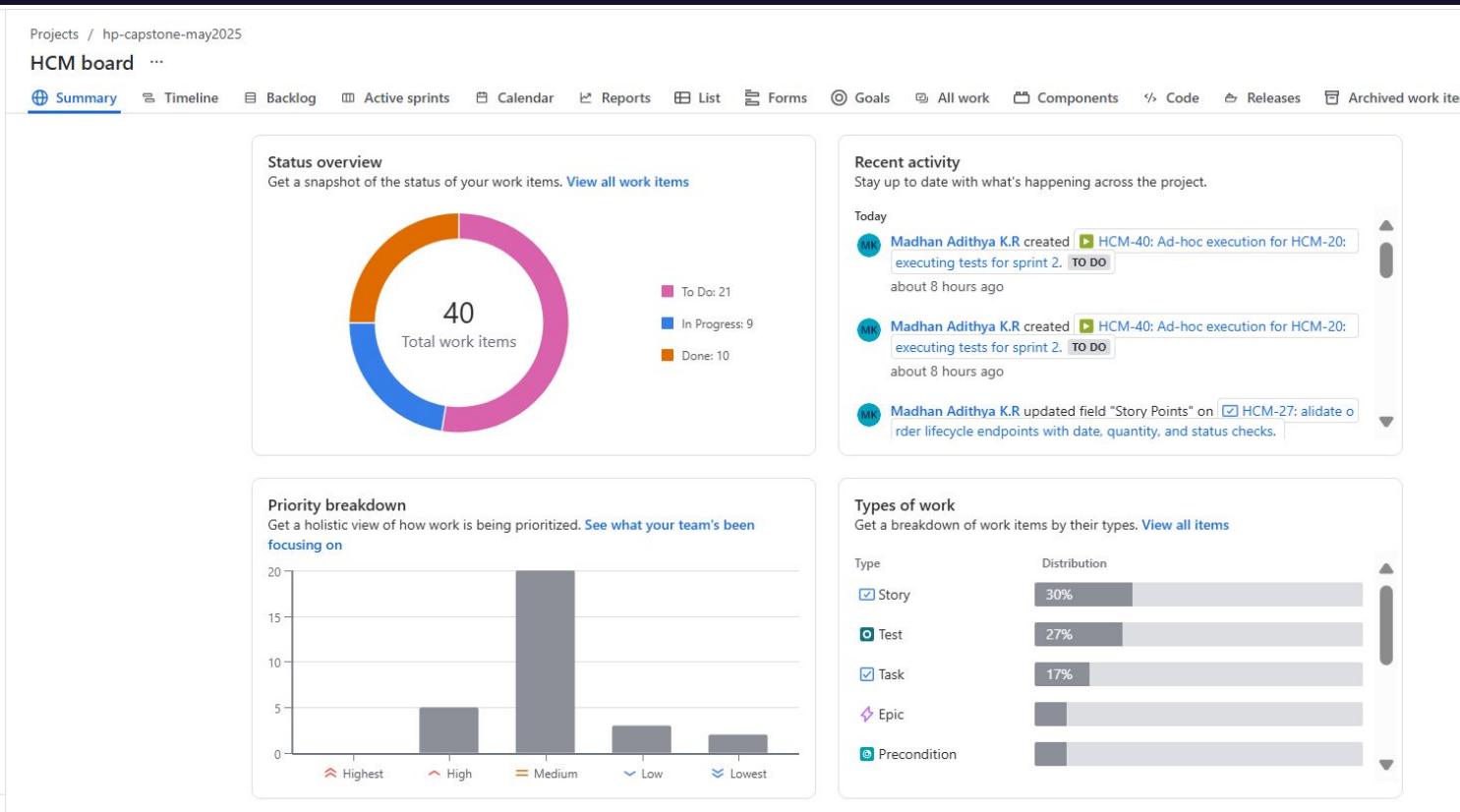
ASCENDION

JIRA

1. Created a project in Jira with 3 epics: Selenium Java Automation, API Testing, and Performance Testing using JMeter.
2. Created stories and tasks for those epics.
3. Using the Xray plugin, created preconditions, test cases, and executed them.
4. Created filters to display only specific issue types.
5. Created a dashboard with multiple gadgets to get a clear overview of the sprint, tasks, and test executions.

Jira project Link: [Sprint 2 - Api testing - HCM board - Scrum Board - Jira](#)

Summary Page and Sprints



Summary

- > Sprint 1 - Selenium Automation 20 May – 27 May (20 work items)
- > Sprint 2 - Api testing 28 May – 3 Jun (3 work items)
- > Sprint 3 - Performance testing 1 Jun – 5 Jun (6 work items)

Sprints



Burn-up chart

ASCENDION

Sprint for My Selenium Project - TakeuForward

Projects / hp-capstone-may2025

HCM board ...

Summary Timeline Backlog Active sprints Calendar Reports List Forms Goals All work Components Code Releases Archived work items More 3 +

Search board Epic Type Label Quick filters Complete sprint Group: Stories

TO DO 14

- Update and verify user profile fields.
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 25, 2025
 HCM-8
- Edit Profile Fields
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 24, 2025
 HCM-9
- Verify Saved Data
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 25, 2025
 HCM-10
- TC_03: Edit Profile Test – Update Personal and Social Details
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 23, 2025
 HCM-11
- TC_04: Verify Profile Personal Data and Check Social Links Open Correct URLs

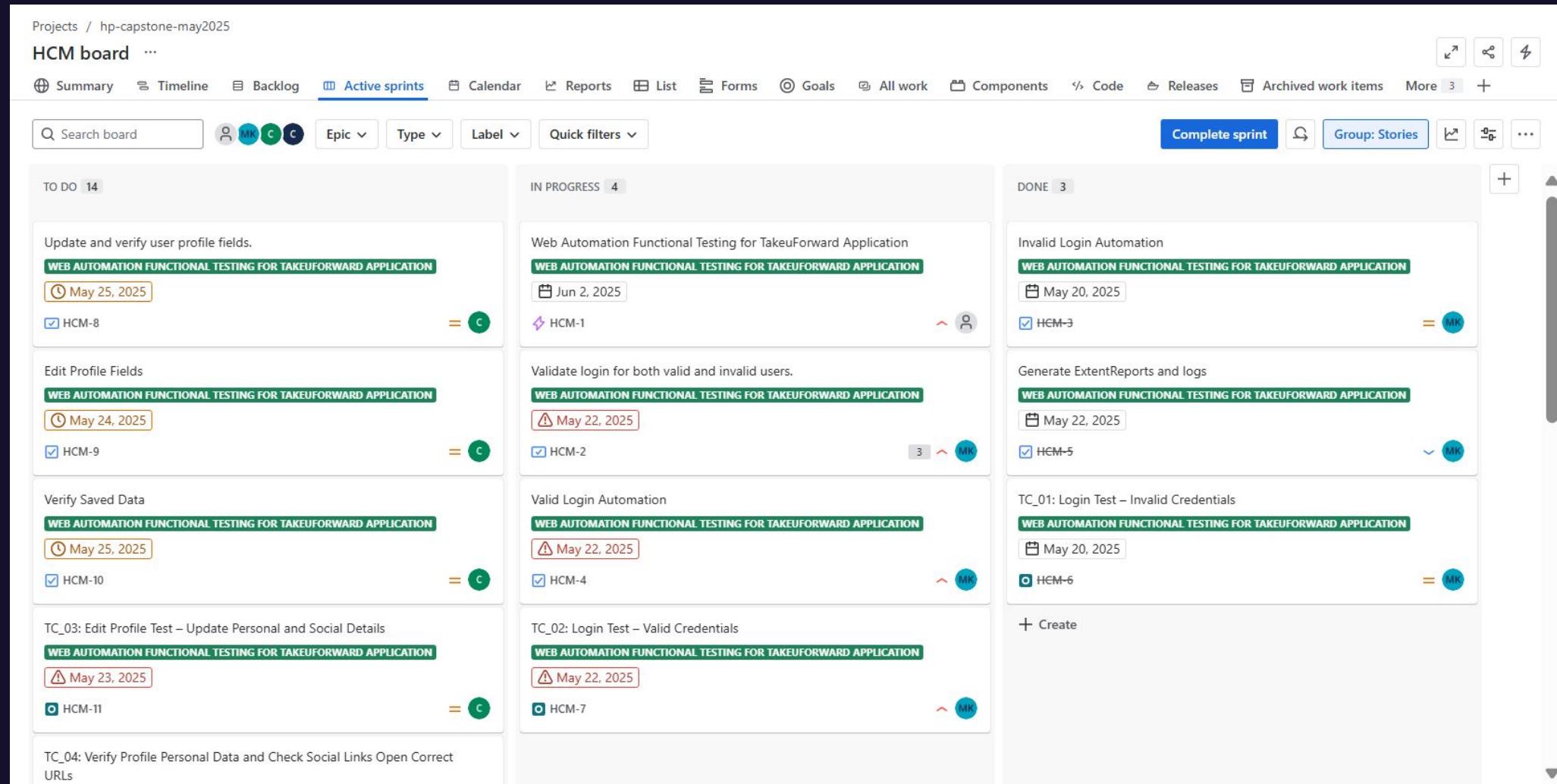
IN PROGRESS 4

- Web Automation Functional Testing for TakeuForward Application
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
Jun 2, 2025
 HCM-1
- Validate login for both valid and invalid users.
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 22, 2025
 HCM-2
- Valid Login Automation
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 22, 2025
 HCM-4
- TC_02: Login Test – Valid Credentials
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 22, 2025
 HCM-7

DONE 3

- Invalid Login Automation
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 20, 2025
 HCM-3
- Generate ExtentReports and logs
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 22, 2025
 HCM-5
- TC_01: Login Test – Invalid Credentials
WEB AUTOMATION FUNCTIONAL TESTING FOR TAKEU FORWARD APPLICATION
May 20, 2025
 HCM-6

+ Create



sample precondition, test executions and Evidence

Preconditions 1

HCM-36 - precondition for TC_01, TC_02: Login Test – Invalid And Valid Credentials

- user should have access to the [takeufoward.org](#) application.
- user should already have the credentials which is registered with the takeufoward application.

Evidence 3

STEP 1 Screenshot 2025-05-25 120310.png

211.3 kB 25/May/2025 12:04 PM

STEP 2 Screenshot 2025-05-25 120536.png

14.1 kB 25/May/2025 12:09 PM

STEP 3 Screenshot 2025-05-25 120710.png

42.3 kB 25/May/2025 12:11 PM

1	Action	Data	Expected Result
	Open login page : takeufoward - Best Coding Tutorials for Free	None	Login page is opened
	Actual Result	Comment	Defects 0
	Evidence (1)		Step State PASSED
2	Action	Data	Expected Result
	Enter email and invalid password	Constant.email_id, Constant.invalid_password	Credentials are entered
	Actual Result	Comment	Defects 0
	Evidence (1)		Step State PASSED
3	Action	Data	Expected Result
	Click on login button	None	Login attempt is made, and invalid credentials toast message appears.
	Actual Result	Comment	Defects 0
	Evidence (1)		Step State PASSED
4	Action	Data	Expected Result
	Assert toast message equals expected	"Invalid Password"	Assertion passes
	Actual Result	Comment	Defects 0
	Evidence		Step State PASSED

Overall Execution Status

6 PASSED

TOTAL TESTS: 6

Rank	Key	Summary	Test Type	Dataset	#Defects	Status
1	HCM-6	TC_01: Login Test – Invalid Credentials	Manual		0	PASSED
2	HCM-7	TC_02: Login Test – Valid Credentials	Manual		0	PASSED
3	HCM-11	TC_03: Edit Profile Test – Update Personal and Social D...	Manual		0	PASSED
4	HCM-12	TC_04: Verify Profile Personal Data and Check Social Lin...	Manual		0	PASSED
5	HCM-17	TC_06: Difficulty Filter Test	Manual		0	PASSED
6	HCM-16	TC_05: Search Functionality Test	Manual		0	PASSED

JIRA Filters

issuetype - Precondition Search ★ Filter details

Basic JQL issuetype = Precondition AND sprint = "Sprint 1 - Selenium Automation"

Order by ▾

precondition for TC_05, TC_06: search and filter Test
HCM-38

precondition for TC_03, TC_04: Edit and Verify Test
HCM-37

precondition for TC_01, TC_02: Login Test – Invalid And Valid Credentials
HCM-36

precondition for TC_05, TC_06: search and filter Test

+ Add @ Apps

Description

to start the test for search and filter, first we have satisfy these preconditions mentioned.

Precondition details

Precondition details Tests

Manual

Steps

- user should have the access to the takeforward application.
- user should able to use the sheets in which we are going to perform search and filters.

Strivers A2Z DSA Course/Sheet - Crack Any FAANG or PBCs

Filter to display only the “preconditions Issue-type” from the Sprint - 1 “Selenium Automation”

issuetype - Test Search ★ Filter details

Basic JQL issuetype = Test AND sprint = "Sprint 1 - Selenium Automation"

Order by ▾

TC_06: Difficulty Filter Test
HCM-17

TC_05: Search Functionality Test
HCM-16

TC_04: Verify Profile Personal Data and Check Social Links Open Correct URLs
HCM-12

TC_03: Edit Profile Test – Update Personal and Social Details
HCM-11

TC_02: Login Test – Valid Credentials
HCM-7

TC_01: Login Test – Invalid Credentials
HCM-6

TC_06: Difficulty Filter Test

+ Add @ Apps

Description

Apply filter, verify results are filtered and relevant.

Linked work items

blocks

HCM-15 Filter Test

Test details

Test details Preconditions Test Sets Test Plans Test Runs

Test Type

Manual

Edit in Dialog ⌂

Filter to display only the “test Issue-type” from the Sprint - 1 “Selenium Automation”

ASCENDION

Dash Board

Days Remaining in Sprint Gadget

Sprint 1 - Selenium Automation

1 Day Remaining

Just now

Overall Test Results: hp-capstone-may2025

Status	Count
PASSED	6
TO DO	4
EXECUTING	5
FAILED	0
N/A	0

Total Tests: 15

Issue Statistics: hp-capstone-may2025 (Assignee)

Assignee	Count	Percentage
Madhan Adithya K.R.	9	30%
capstonetestuser02	9	30%
capstonetestuser01	9	30%
Unassigned	3	10%
Total	30	

Just now

Test Executions List: hp-capstone-may2025

Key	Summary	Test Plan	Test Environment	Status
HCM-40	Ad-hoc execution for HCM-20: executing tests for s...			<div style="width: 50%;">Yellow</div>
HCM-39	Ad-hoc execution for HCM-6: executing testcases of sp...			<div style="width: 100%;">Green</div>

Prev 1 Next

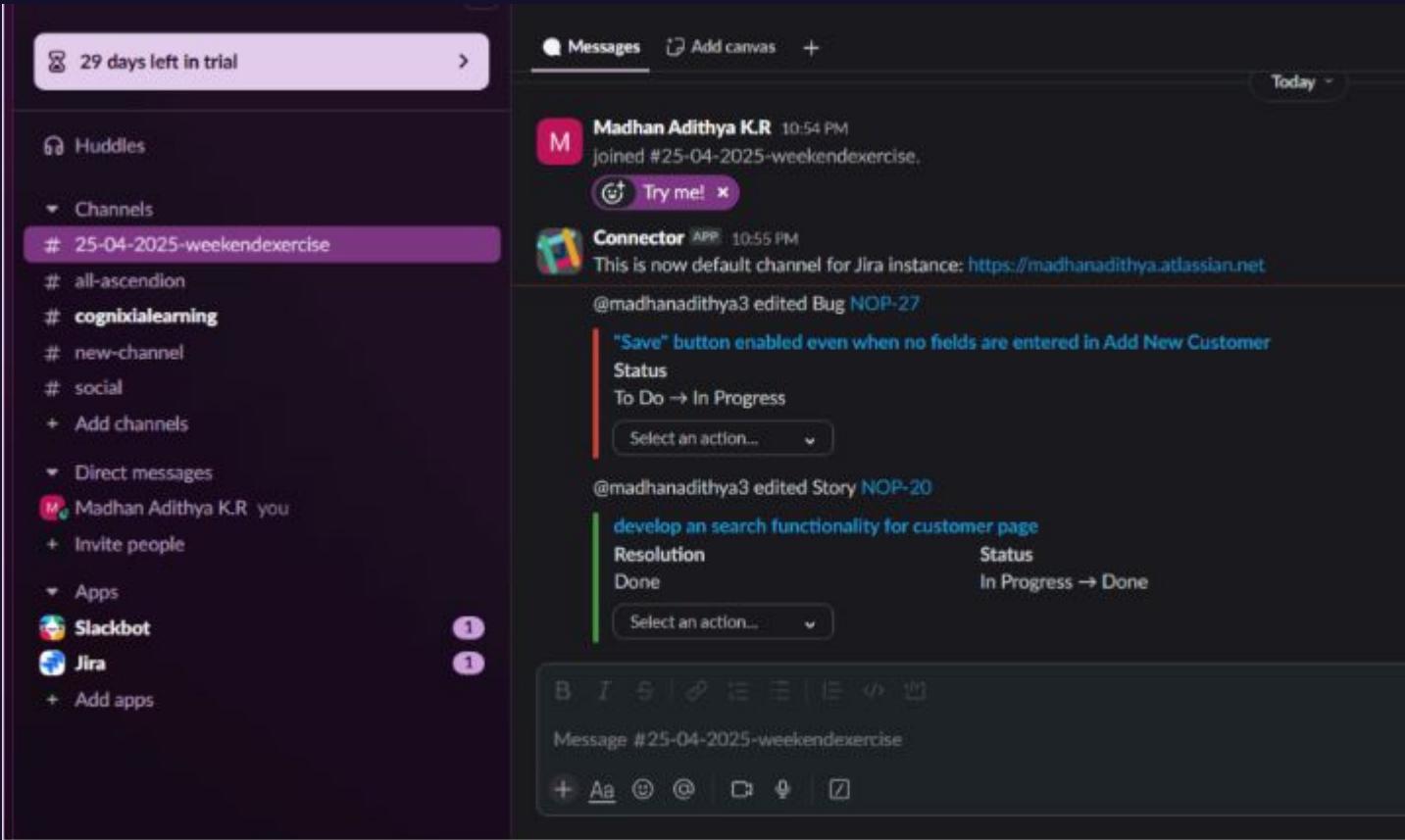
Filter Results: issuetype - Test Search

T	Key	Summary	P
<input checked="" type="checkbox"/>	HCM-17	TC_06: Difficulty Filter Test	<div style="width: 50%;"></div>
<input checked="" type="checkbox"/>	HCM-16	TC_05: Search Functionality Test	<div style="width: 50%;"></div>
<input checked="" type="checkbox"/>	HCM-12	TC_04: Verify Profile Personal Data and Check Social Links Open Correct URLs	<div style="width: 50%;"></div>
<input checked="" type="checkbox"/>	HCM-11	TC_03: Edit Profile Test – Update Personal and Social Details	<div style="width: 50%;"></div>
<input checked="" type="checkbox"/>	HCM-7	TC_02: Login Test – Valid Credentials	<div style="width: 50%;"></div>
<input checked="" type="checkbox"/>	HCM-6	TC_01: Login Test – Invalid Credentials	<div style="width: 50%;"></div>

1-6 of 6

Just now

Slack Integration



Selenium Java Automation Test

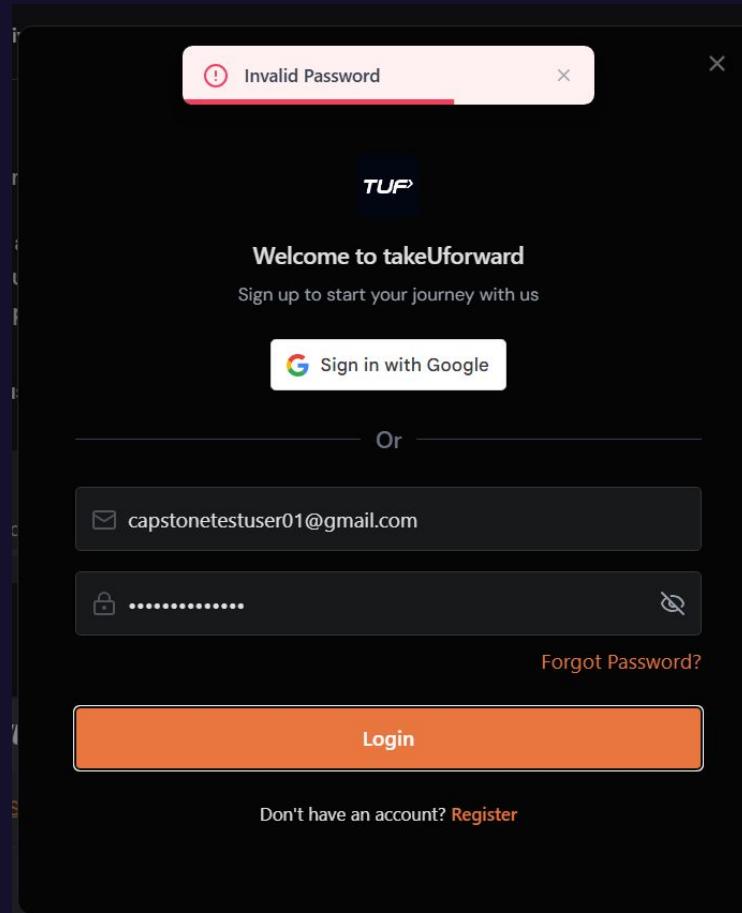
1. Automated key user workflows on takeuforward.org
2. Validated login (valid/invalid users)
3. Edited and verified user profile fields
4. Implemented search & difficulty filter testing
5. Used TestNG, Page Object Model, log4j, Extent Reports, screenshots and also used Apache-poi Data Driven

Demo video Drive Link:

https://drive.google.com/file/d/12rv6VtvZ0xqmwl3dHPskHfi2nsLURHTML/view?usp=drive_link

Expected Results

TC_01 : Invalid Login Test



TC_02: Valid Login Test

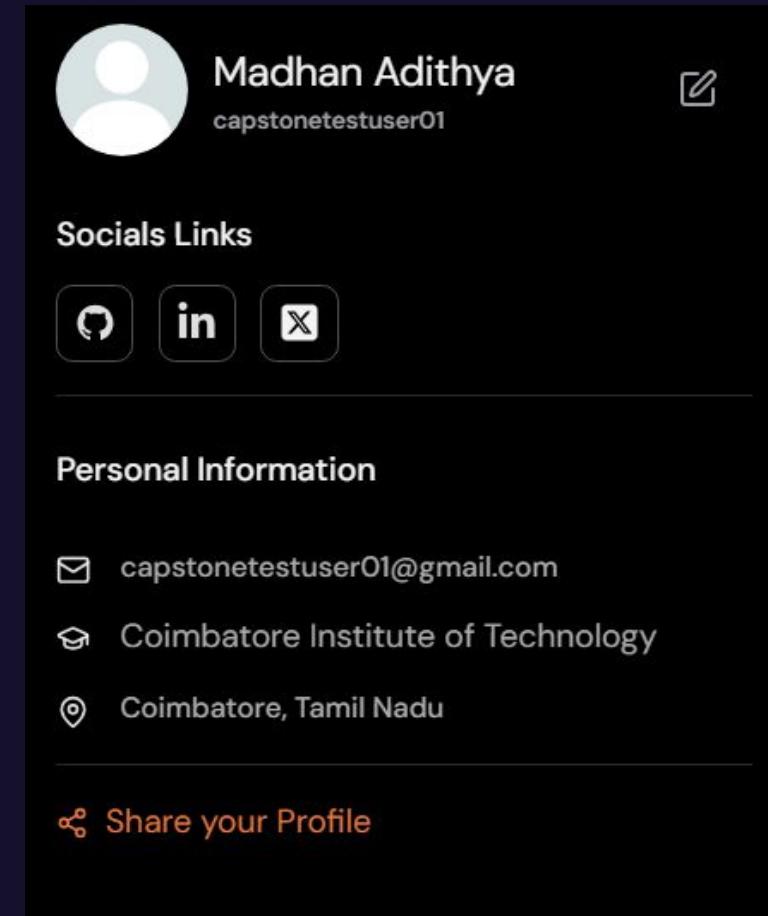


Expected Results

TC_03 : Edit Profile Test

A screenshot of a web application's profile editing interface. At the top, there is a green success message box containing the text "Personal Info Updated". Below this, the user's name is displayed as "Madhan Adithya" and their email as "capstonetestuser01@gmail.com". Underneath these fields are sections for "Location" (Coimbatore, Tamil Nadu) and "College" (Coimbatore Institute of Technology). A "Save Changes" button is located at the bottom left of the form.

TC_04: Verify Updated Profile Test



ASCENDION

Expected Results

TC_05 : Search Test

The screenshot shows the ASCENDION platform interface. At the top, there are two tabs: "All Problems" (which is selected) and "Revision". Below the tabs is a search bar containing the text "The celebrity Problem" with a clear button (X). To the right of the search bar are buttons for "Difficulty" (with a dropdown arrow) and "Pick Random". The main area displays a table with columns: Status, Problem, TUF (with a logo), Resource (Plus), Resource (Free), Practice, Note, Revision, and Difficulty. A single row is visible for "The Celebrity Problem", which is marked as solved ("Solve" button is greyed out). It has a difficulty level of "Hard". There are also icons for a plus sign, a star, and a minus sign.

TC_06: Filter Test

The screenshot shows the ASCENDION platform interface with a different filter applied. The search bar now contains "Sort" and the difficulty filter is set to "Hard". The table below shows two rows of problems: "Median of 2 sorted arrays" and "Merge M sorted Lists". Both problems are marked as solved ("Solve" button is greyed out). They both have a difficulty level of "Hard". The "Resource (Plus)" and "Resource (Free)" columns show minus signs (-) indicating no resources are available. There are icons for a plus sign, a star, and a minus sign in the "Revision" column.

ASCENDION

Excel Data for Data Driven Approach

sheet : Login

	A	B	C	D
1	testCaseld	email	password	expectedToast
2	TC_01	capstonetestuser01@gmail.com	dummyPassword	Invalid Password
3	TC_02	capstonetestuser01@gmail.com	password@1234	Login Successful
4				
5				

Sheet: Pofile

	A	B	C	D	E	F
1	testCaseld	name	location	college	github	linkedin
2	TC_03	Madhan Adithya	Coimbatore, Tamil Nadu	Coimbatore Institute of Technology	https://github.com/madhanadithya/	https://www.linkedin.com/in/madhanadithya/
3						

Sheet: Search

	A	B	C	D
1	testCaseld	keyword1	keyword2	difficulty
2	TC_05_06	The Celebrity Problem	Sort	Hard
3				

Sample Test Code Snippet from my Project

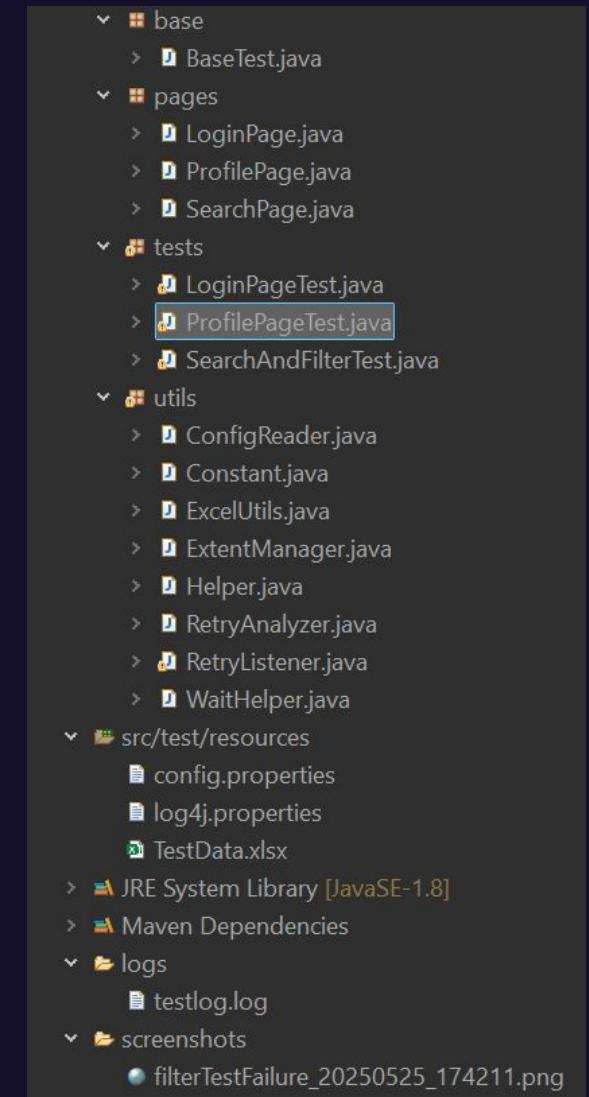
ProfilePageTest.java

```

106 //===== test 4 : verifying the updates made on the profile page =====
107
108 @Test(priority = 4, dependsOnMethods = {"editProfileTest"}, groups = {"profile", "verification", "positive"})
109 public void verifyProfileLinksTest() {
110
111     test = extent.createTest("TC_04 : Verify Profile Social Links Open Correct URLs");
112
113     try {
114         log.info("===== Verifying Display Name =====");
115         test.log(Status.INFO, "Verifying GitHub and LinkedIn URLs");
116
117         ProfilePage profilePage = new ProfilePage(getDriver());
118
119         Map<String, String> data = ExcelUtils.getTestData("Profile", "TC_03"); //with the help of ExcelUtils.java loading data form Excel.
120
121         //===== verify display name =====
122
123         String extractedName = profilePage.getDisplayedUserFullName(); // calling ProfilePage.java class to get the User name displayed in profile
124         log.info("Name displayed in profile: " + extractedName);
125         test.log(Status.INFO, "Name displayed in profile:" + extractedName);
126
127     //     Assert.assertEquals(extractedName, Constant.profile_name); //----> Using Constant.java file
128     //     Assert.assertEquals(extractedName, data.get("name")); //----> Using Excel sheets - data driven approach
129
130         log.info("Profile name matches with data used to update!");
131         test.log(Status.PASS, "Profile name matches with data used to update!");
132
133         //===== verify linked-in =====
134
135         log.info("===== Verifying social links URLs =====");
136         String linkedinUrl = profilePage.verifyAndGetlinkedinUrl();
137
138         log.info("LinkedIn URL opened: " + linkedinUrl); //----> using log4j logging the information.
139         test.log(Status.INFO, "LinkedIn URL opened: " + linkedinUrl); //---> using extends-report logging the info in html report.
140
141     //     Assert.assertTrue(linkedinUrl.contains("linkedin.com"), "LinkedIn URL validation failed");
142     //     Assert.assertTrue(linkedinUrl.contains("linkedin.com"));
143
144         //===== verify github =====
145
146         String githubUrl = profilePage.verifyAndGetgithubUrl();
147
148         log.info("GitHub URL opened: " + githubUrl);
149         test.log(Status.INFO, "GitHub URL opened: " + githubUrl);
150
151     //     Assert.assertTrue(githubUrl.contains(Constant.github_link), "GitHub URL validation failed");
152     //     Assert.assertTrue(githubUrl.contains(data.get("github")));
153
154         test.log(Status.PASS, "Social links verified successfully!");
155         log.info("Social links verified successfully, As expected test passed!");
156
157     } catch (Exception e) {
158         Helper.captureScreenshot(getDriver(), "verifyProfileLinksTestFailure");
159         test.log(Status.FAIL, "Verification test failed: " + e.getMessage());
160         log.error("Verification test failed", e);
161         Assert.fail("Verification test crashed");
162     }
163 }

```

ProfilePageTest.java



ASCENDION

Execution in CLI : Using “mvn test”

```
[INFO] -----  
[INFO] T E S T S  
[INFO] -----  
[INFO] Running TestSuite  
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".  
SLF4J: Defaulting to no-operation (NOP) logger implementation  
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.  
=====launchApp=====  
May 26, 2025 1:47:17 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch  
WARNING: Unable to find an exact match for CDP version 136, returning the closest version; found: 135; Please update to a Selenium version that supports CDP version 136  
2025-05-26 01:47:24 INFO LoginPageTest:76 - =====login with invalid credentials=====  
2025-05-25T20:17:24.966633Z main ERROR Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger to log to the console...  
2025-05-26 01:47:26 INFO LoginPageTest:40 - Opening login page...  
2025-05-26 01:47:28 INFO LoginPageTest:44 - invalid credentials submitted.  
2025-05-26 01:47:29 INFO LoginPageTest:48 - Captured Toast Message: Invalid Password  
2025-05-26 01:47:29 INFO LoginPageTest:63 - As expected test passed!  
2025-05-26 01:47:30 INFO LoginPageTest:90 - =====login with valid credentials=====  
2025-05-26 01:47:30 INFO LoginPageTest:40 - Opening login page...  
2025-05-26 01:47:34 INFO LoginPageTest:44 - valid credentials submitted.  
2025-05-26 01:47:35 INFO LoginPageTest:48 - Captured Toast Message: Login Successful  
2025-05-26 01:47:39 INFO LoginPageTest:56 - Captured greeting Message: Hi, CapstoneTestUser01  
2025-05-26 01:47:39 INFO LoginPageTest:59 - Verified greeting message. Login successful!  
2025-05-26 01:47:39 INFO LoginPageTest:63 - As expected test passed!  
2025-05-26 01:47:39 INFO ProfilePageTest:41 - ===== Editing profile details =====  
2025-05-26 01:47:39 INFO ProfilePageTest:47 - Updating personal data: name, location, college  
2025-05-26 01:47:47 INFO ProfilePageTest:81 - Updating social links: GitHub, LinkedIn, Twitter, Other  
2025-05-26 01:47:57 INFO ProfilePageTest:92 - Profile updated successfully, As expected test passed!!  
2025-05-26 01:47:57 INFO ProfilePageTest:114 - ===== Verifying Display Name =====  
2025-05-26 01:47:57 INFO ProfilePageTest:124 - Name displayed in profile: Madhan Adithya  
2025-05-26 01:47:57 INFO ProfilePageTest:130 - Profile name matches with data used to update!  
2025-05-26 01:47:57 INFO ProfilePageTest:135 - ===== Verifying social links URLs =====  
2025-05-26 01:48:04 INFO ProfilePageTest:138 - LinkedIn URL opened: https://www.linkedin.com/authwall?trk=gf&trkInfo=AQEhcTTvmIM69QAAZcJGFyoLrtAK-rxMifyf-4I3b11Zgzk8BFRer=&sessionRedirect=https%3A%2F%2Fwww.linkedin.com%2Fin%2Fmadhanadithya%2F  
2025-05-26 01:48:13 INFO ProfilePageTest:148 - GitHub URL opened: https://github.com/madhanadithya/  
2025-05-26 01:48:13 INFO ProfilePageTest:155 - Social links verified successfully, As expected test passed!  
2025-05-26 01:48:13 INFO SearchAndFilterTest:33 - ===== search and filter feature testing =====  
2025-05-26 01:48:13 INFO SearchAndFilterTest:34 - Navigating to Striver's DSA sheet  
2025-05-26 01:48:19 INFO SearchAndFilterTest:43 - ===== search test =====  
2025-05-26 01:48:19 INFO SearchAndFilterTest:49 - Performing search test  
2025-05-26 01:48:29 INFO SearchAndFilterTest:66 - Search test completed successfully, As expected test passed!  
2025-05-26 01:48:29 INFO SearchAndFilterTest:80 - ===== filter test =====  
2025-05-26 01:48:29 INFO SearchAndFilterTest:100 - Applying difficulty filter  
2025-05-26 01:48:36 INFO SearchAndFilterTest:106 - Filter result: Hard  
2025-05-26 01:48:36 INFO SearchAndFilterTest:110 - Filter test passed, As expected test passed!  
=====closeApp=====  
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 98.57 s -- in TestSuite  
[INFO] -----  
[INFO] Results:  
[INFO] -----  
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 01:49 min  
[INFO] Finished at: 2025-05-26T01:48:51+05:30  
[INFO]
```

Logs using log4j

```
testlog.log x
1 2025-05-26 00:44:04,292 INFO tests.LoginPageTest - =====login with invalid credentials=====
2 2025-05-26 00:44:07,938 INFO tests.LoginPageTest - Opening login page...
3 2025-05-26 00:44:12,245 INFO tests.LoginPageTest - invalid credentials submitted.
4 2025-05-26 00:44:14,631 INFO tests.LoginPageTest - Captured Toast Message: Invalid Password
5 2025-05-26 00:44:14,642 INFO tests.LoginPageTest - As expected test passed!
6 2025-05-26 00:44:16,304 INFO tests.LoginPageTest - =====login with valid credentials=====
7 2025-05-26 00:44:16,507 INFO tests.LoginPageTest - Opening login page...
8 2025-05-26 00:44:20,414 INFO tests.LoginPageTest - valid credentials submitted.
9 2025-05-26 00:44:23,306 INFO tests.LoginPageTest - Captured Toast Message: Login Successful
10 2025-05-26 00:44:28,277 INFO tests.LoginPageTest - Captured greeting Message: Hi, CapstoneTestUser01
11 2025-05-26 00:44:28,279 INFO tests.LoginPageTest - Verified greeting message. Login successful!
12 2025-05-26 00:44:28,281 INFO tests.LoginPageTest - As expected test passed!
13 2025-05-26 00:44:28,528 INFO tests.ProfilePageTest - ===== Editing profile details =====
14 2025-05-26 00:44:28,552 INFO tests.ProfilePageTest - Updating personal data: name, location, college
15 2025-05-26 00:44:42,676 INFO tests.ProfilePageTest - Updating social links: GitHub, LinkedIn, Twitter, Other
16 2025-05-26 00:44:54,707 INFO tests.ProfilePageTest - Profile updated successfully, As expected test passed!!
17 2025-05-26 00:44:54,889 INFO tests.ProfilePageTest - ===== Verifying Display Name =====
18 2025-05-26 00:44:55,069 INFO tests.ProfilePageTest - Name displayed in profile: Madhan Adithya
19 2025-05-26 00:44:55,070 INFO tests.ProfilePageTest - Profile name matches with data used to update!
20 2025-05-26 00:44:55,071 INFO tests.ProfilePageTest - ===== Verifying social links URLs =====
21 2025-05-26 00:45:02,859 INFO tests.ProfilePageTest - LinkedIn URL opened: https://www.linkedin.com/authwall?trk=gf&trkInfo=
22 2025-05-26 00:45:15,340 INFO tests.ProfilePageTest - GitHub URL opened: https://github.com/madhanadithya/
23 2025-05-26 00:45:15,343 INFO tests.ProfilePageTest - Social links verified successfully, As expected test passed!
24 2025-05-26 00:45:15,565 INFO tests.SearchAndFilterTest - ===== search and filter feature testing =====
25 2025-05-26 00:45:15,566 INFO tests.SearchAndFilterTest - Navigating to Striver's DSA sheet
26 2025-05-26 00:45:23,277 INFO tests.SearchAndFilterTest - ===== search test =====
27 2025-05-26 00:45:23,278 INFO tests.SearchAndFilterTest - Performing search test
28 2025-05-26 00:45:34,234 INFO tests.SearchAndFilterTest - Search test completed successfully, As expected test passed!
29 2025-05-26 00:45:34,427 INFO tests.SearchAndFilterTest - ===== filter test =====
30 2025-05-26 00:45:34,558 INFO tests.SearchAndFilterTest - Applying difficulty filter
31 2025-05-26 00:45:41,698 INFO tests.SearchAndFilterTest - Filter result: Hard
32 2025-05-26 00:45:41,700 INFO tests.SearchAndFilterTest - Filter test passed, As expected test passed!
```

Extent Report

May 26, 2025 01:47:24 AM

Tests		
TC_01 : Login Test - Invalid Credentials 1:47:26 AM / 00:00:03:157	Pass	
TC_02 : Login Test - Valid Credentials 1:47:30 AM / 00:00:08:796	Pass	
TC_03 : Edit Profile Test - Update Personal and Social Details 1:47:39 AM / 00:00:17:687	Pass	
TC_04 : Verify Profile Social Links Open Correct URLs 1:47:57 AM / 00:00:15:732	Pass	
TC_05 : Search Functionality Test 1:48:19 AM / 00:00:09:967	Pass	
TC_06 : Difficulty Filter Test 1:48:29 AM / 00:00:06:604	Pass	

TC_04 : Verify Profile Social Links Open Correct URLs

05.26.2025 1:47:57 AM 05.26.2025 1:48:13 AM 00:00:15:732 · #test-id=4

STATUS	TIMESTAMP	DETAILS
Info	1:47:57 AM	Verifying GitHub and LinkedIn URLs
Info	1:47:57 AM	Name displayed in profile:Madhan Adithya
Pass	1:47:57 AM	Profile name matches with data used to update!
Info	1:48:04 AM	LinkedIn URL opened: https://www.linkedin.com/authwall?trk=gf&trkInfo=AQEhcTTvmlM69QAAZcJGFyoLrtAK-rxMifyf-4l3b11Zgzk8BFRG3L-1ySjpfRaz9z_ZSrKx2xCF1ZUCn80gehBrEcW4D9VVUwljAv0_2cL8ZkPDAk2v4UYLemwYuY98yzzX4k=&original_referer=&sessionRedirect=https%3A%2F%2Fwww.linkedin.com%2Fin%2Fmadhanadithya%2F
Info	1:48:13 AM	GitHub URL opened: https://github.com/madhanadithya/
Pass	1:48:13 AM	Social links verified successfully!

ASCENDION

Extent Report

May 26, 2025 01:47:24 AM

Started May 26, 2025 01:47:24 AM

Ended May 26, 2025 01:48:36 AM

Tests Passed 6

Tests Failed 0

Tests

Pass

6 tests passed
0 tests failed, 0 skipped, 0 others

Log events

Pass Info

10 events passed
0 events failed, 17 others

Timeline

0 10 20 30 40 50 60 70

System/Environment

NAME	VALUE
Tester	Automation Hero ?
OS	Windows 11

Surefire-Reports

All suites

Suite

Info

- C:\Users\madhan.adithya\eclipse-workspace\TakeUforward\testng.xml
- 1 test
- 7 groups**
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 6 methods, 6 passed
- Passed methods (hide)
 - editProfileTest
 - filterTest
 - invalidLoginTest
 - searchTest
 - validLoginTest
 - verifyProfileLinksTest

Groups for Suite

filter

filterTest

login

invalidLoginTest
validLoginTest

negative

invalidLoginTest

positive

editProfileTest
filterTest
searchTest
validLoginTest
verifyProfileLinksTest

profile

editProfileTest
verifyProfileLinksTest

search

searchTest

verification

verifyProfileLinksTest

All suites

Suite

Info

- C:\Users\madhan.adithya\eclipse-workspace\TakeUforward\testng.xml
- 1 test
- 7 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 6 methods, 6 passed
- Passed methods (hide)
 - editProfileTest
 - filterTest
 - invalidLoginTest
 - searchTest
 - validLoginTest
 - verifyProfileLinksTest

tests.LoginPageTest

invalidLoginTest
validLoginTest

tests.ProfilePageTest

editProfileTest
verifyProfileLinksTest

tests.SearchAndFilterTest

filterTest
searchTest

Tags/Groups

Passed Tests - All cases passed

Bug Found:

Bug Report: Incomplete URL Validation for Social Links

URL: [Edit Profile Page](#)

Issue:

The social links fields (LinkedIn, GitHub, Twitter) only validate for protocol and domain, not the presence of a username.

Steps to Reproduce:

1. Go to the Edit Profile page.
2. Enter a social link like `https://github.com/` (without a username).
3. Save the changes.

Expected Behavior:

The input should be rejected if it lacks a username or valid path after the domain.

Actual Behavior:

The form accepts the input even if the username is missing.

Example:

Accepted input: `https://github.com/`

Expected: Should only accept URLs like `https://github.com/username`

Severity: Medium

I think they haven't fixed this bug yet because it might not be a current business priority them.

API Testing Using Postman And Newman

- 1) Created db.json with endpoints: users, products, orders
- 2) Performed full CRUD operations via Postman for all 3 Endpoints.
- 3) Validated the requests using Scripts.
- 4) Executed via Newman CLI with reports

Share Collection via API :

https://api.postman.com/collections/37293848-1fc82365-385e-4862-a884-48ed842cc49f?access_key=PMAT-01JW4Q5KN194XJKJRY72CPBNZ

ASCENDION

API Testing Using Postman

The screenshot shows the left sidebar of the Postman application. It displays a hierarchical tree structure of API endpoints:

- Capstone API Testing - json-server
 - Users
 - POST POST Create users
 - GET GET All users
 - GET GET Single users
 - PUT PUT Update users
 - PATCH PATCH Update users
 - DEL DELETE users
 - Products
 - POST POST Create products
 - GET GET All products
 - GET GET Single products
 - PUT PUT Update products
 - PATCH PATCH Update products
 - DEL DELETE products
 - Orders
 - POST POST Create orders
 - GET GET All orders
 - GET GET Single orders
 - PUT PUT Update orders
 - PATCH PATCH Update orders
 - DEL DELETE orders

The screenshot shows a single API request configuration in Postman:

- Method:** GET
- URL:** {{base_url}} /orders/{{orderId}}
- Headers:** (7)
- Body:** (empty)
- Scripts:** (selected)
- Pre-request:**

```
1 pm.test("Status code is 200", () => pm.response.to.have.status(200));
```
- Post-response:**

```
1 pm.test("Response time is under 2000ms", () => pm.expect(pm.response.elapsedTime).to.be.below(2000));
```
- Body:** (selected)
- JSON** preview:

```
1 {  
2   "id": "5",  
3   "userId": 6,  
4   "productId": 5,  
5   "quantity": 2,  
6   "orderDate": "2025-05-24",  
7   "status": "Shipped"  
8 }
```

Sample Get

The screenshot shows a second API request configuration in Postman:

- Method:** GET
- URL:** {{base_url}} /orders
- Headers:** (7)
- Body:** (empty)
- Scripts:** (selected)
- Pre-request:**

```
1 pm.test("Status code is 200", () => pm.response.to.have.status(200));
```
- Post-response:**

```
1 pm.test("Response time is under 2000ms", () => pm.expect(pm.response.elapsedTime).to.be.below(2000));
```
- Body:** (selected)
- JSON** preview:

```
1 [  
2   {  
3     "id": "1",  
4     "userId": 1,  
5     "productId": 1,  
6     "quantity": 2,  
7     "orderDate": "2025-05-23",  
8     "status": "Delivered"  
9   },  
10   {  
11     "id": "2",  
12     "userId": 2,  
13     "productId": 3,  
14     "quantity": 1,  
15     "orderDate": "2025-05-22",  
16     "status": "Pending"  
17   },  
18   {  
19     "id": "3",  
20     "userId": 4,  
21     "productId": 3,  
22     "quantity": 2,  
23     "orderDate": "2025-05-23",  
24     "status": "Delivered"  
25   },  
26 ]
```

Sample Get All

File Structure

ASCENDION

API Testing Using Postman

HTTP Capstone API Testing - json-server / Orders / POST Create orders

POST `{{base_url}} /orders`

Params Authorization Headers (10) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary

```
1 {  
2   "id": "5",  
3   "userId": 6,  
4   "productId": 5,  
5   "quantity": 2,  
6   "orderDate": "2025-05-24",  
7   "status": "Shipped"  
8 }
```

Body Cookies Headers (9) Test Results (9/9)

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

```
1 {  
2   "id": "5",  
3   "userId": 6,  
4   "productId": 5,  
5   "quantity": 2,  
6   "orderDate": "2025-05-24",  
7   "status": "Shipped"  
8 }
```

HTTP Capstone API Testing - json-server / Orders / PATCH Update orders

PATCH `{{base_url}} /orders/ {{orderId}}`

Params Authorization Headers (10) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   // "id": "5",  
3   // "userId": 6,  
4   // "productId": 5,  
5   // "quantity": 2,  
6   // "orderDate": "2025-05-24",  
7   "status": "Delivered"  
8 }
```

Body Cookies Headers (9) Test Results (5/5)

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

```
1 {  
2   "id": "5",  
3   "userId": 6,  
4   "productId": 5,  
5   "quantity": 2,  
6   "orderDate": "2025-05-24",  
7   "status": "Delivered"  
8 }
```

DELETE `{{base_url}} /orders/ {{orderId}}`

Params Authorization Headers (7) Body Scripts Settings

Pre-request

```
1 pm.test("Status code is 200", () => pm.response.assertions.add({  
2   status: 200  
3 })  
4 pm.test("Response time is under 2000ms", () => pm.response.assertions.add({  
5   duration: 2000  
6 }))  
7 pm.test("Content-Type is application/json", () => pm.response.assertions.add({  
8   type: "application/json"  
9 }))
```

Post-response

Body Cookies Headers (9) Test Results (6/6)

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

```
1 {  
2   "id": "5",  
3   "userId": 6,  
4   "productId": 5,  
5   "quantity": 2,  
6   "orderDate": "2025-05-24",  
7   "status": "Shipped"  
8 }
```

Sample Post

Sample Patch

Sample Delete

API Testing Using Postman

```

PUT {{base_url}} /orders/{{orderId}}
Params Authorization Headers (10) Body Scripts • Settings
Pre-request
1 pm.test("Status code is 200", () => pm.response.to.have.status(200));
2
3 pm.test("Response time is under 2000ms", () => pm.expect(pm.response.responseTime).to.be.below(2000));
4
5 pm.test("Content-Type is application/json", () => pm.expect(pm.response.headers.get("Content-Type")).to.include("application/json"));
6
7 requestBody = pm.request.body.raw ? JSON.parse(pm.request.body.raw) : {};
8 responseBody = pm.response.json();
9
10 pm.test("Response matches updated order data", () => {
11   pm.expect(responseBody).to.deep.equal(requestBody);
12 });
13
14 pm.test("Order ID is consistent with environment variable", () => {
15   pm.expect(responseBody.id).to.eql(pm.environment.get("orderId"));
16 });
17
18 pm.test("Status is valid", () => {
19   const validStatuses = ["Pending", "Confirmed", "Shipped", "Delivered", "Cancelled"];
20   pm.expect(validStatuses).to.include(responseBody.status);
21 });
22
23 pm.test("Negative - Order ID is not null or empty", () => {
24   pm.expect(responseBody.id).to.not.be.oneOf([null, ""]);
25 });
26
27 pm.test("Negative - Status is not an invalid value", () => {
28   const invalidStatuses = ["Unknown", "Processing", "X"];
29   pm.expect(invalidStatuses).to.not.include(responseBody.status);
30 });
31
32 pm.test("Negative - Quantity is not zero or negative", () => {
33   pm.expect(responseBody.quantity).to.not.be.below(1);
34 });
35

```

Sample Validation Script for PUT

Body	Cookies	Headers (9)	Test Results (9/9)	🕒
Filter Results ▾				
PASSED	Status code is 200			
PASSED	Response time is under 2000ms			
PASSED	Content-Type is application/json			
PASSED	Response matches updated order data			
PASSED	Order ID is consistent with environment variable			
PASSED	Status is valid			
PASSED	Negative - Order ID is not null or empty			
PASSED	Negative - Status is not an invalid value			
PASSED	Negative - Quantity is not zero or negative			

Sample Test Result

ASCENDION

API Testing Using Newman

```
└ Users
  └ POST Create users
    POST http://localhost:3000/users [201 Created, 441B, 42ms]
    ✓ Status code is 201
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response body matches request body
    ✓ Response contains all required user fields
    ✓ Email format is valid
    ✓ Phone is numeric and 10 digits

  └ GET All users
    GET http://localhost:3000/users [200 OK, 1.25kB, 12ms]
    ✓ Status code is 200
    ✓ Response time is less than 2000ms
    ✓ Content-Type is application/json
    ✓ Response is an array
    ✓ Each user object contains expected fields
    ✓ Sample user exists: madhan with email madhan@gmail.com
    ✓ All emails are valid format

  └ GET Single users
    GET http://localhost:3000/users/7 [200 OK, 436B, 5ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response contains all required user fields
    ✓ Returned user ID matches requested userId
    ✓ Email format is valid
    ✓ Phone is numeric and 10 digits

  └ PUT Update users
    PUT http://localhost:3000/users/7 [200 OK, 442B, 36ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response body matches request body
    ✓ Response contains all required user fields
    ✓ Response contains no extra fields
    ✓ Name field is not empty
    ✓ ID is a positive integer string
    ✓ Phone number does not start with 0

  └ PATCH Update users
    PATCH http://localhost:3000/users/7 [200 OK, 448B, 33ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response contains all required fields
    ✓ Returned user ID matches environment user1d
    ✓ Phone is numeric and 10 digits
    ✓ Name field is non-empty string

  └ DELETE users
    DELETE http://localhost:3000/users/7 [200 OK, 448B, 41ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
```

```
└ Products
  └ POST Create products
    POST http://localhost:3000/products [201 Created, 423B, 36ms]
    ✓ Status code is 201
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response body matches request body
    ✓ Product fields are correct

  └ GET All products
    GET http://localhost:3000/products [200 OK, 996B, 6ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response is an array
    ✓ Each product has required fields
    ✓ Product IDs are unique
    ✓ Prices are positive numbers
    ✓ inStock is boolean
    ✓ Category is a non-empty string

  └ GET Single products
    GET http://localhost:3000/products/6 [200 OK, 418B, 4ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Product has all required fields
    ✓ Product ID matches requested ID
    ✓ Price is a positive number
    ✓ inStock is boolean
    ✓ Category and name are valid strings

  └ PUT Update products
    PUT http://localhost:3000/products/6 [200 OK, 419B, 26ms]
    ✓ Updated product matches request body
    ✓ Product fields are valid
    ✓ Product ID remains unchanged

  └ PATCH Update products
    PATCH http://localhost:3000/products/6 [200 OK, 426B, 22ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Price is not zero or negative after patch
    ✓ inStock defaults to boolean true/false if not patched
    ✓ ID remains unchanged after patch

  └ DELETE products
    DELETE http://localhost:3000/products/6 [200 OK, 426B, 29ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Deleted product has all expected fields
    ✓ Product ID in response matches deleted product ID
    GET http://localhost:3000/products/6 [404 Not Found, 387B, 3ms]
    ✓ Negative - Confirm product is deleted - GET returns 404
```

```
└ Orders
  └ POST Create orders
    POST http://localhost:3000/orders [201 Created, 445B, 24ms]
    ✓ Status code is 201
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response matches request body
    ✓ Response has all order fields
    ✓ Order ID is numeric
    ✓ Quantity is a positive number
    ✓ Order date format is valid
    ✓ Status is valid

  └ GET All orders
    GET http://localhost:3000/orders [200 OK, 1kB, 4ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response is an array
    ✓ Each order has all required fields
    ✓ Order quantity is a positive integer
    ✓ Order status is valid
    ✓ Order date format is valid (YYYY-MM-DD)

  └ GET Single orders
    GET http://localhost:3000/orders/5 [200 OK, 440B, 4ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-type is application/json
    ✓ Response has all expected order fields
    ✓ Order ID matches requested ID
    ✓ Order status is among accepted values

  └ PUT Update orders
    PUT http://localhost:3000/orders/5 [200 OK, 440B, 170ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Response matches updated order data
    ✓ Order ID is consistent with environment variable
    ✓ Status is valid
    ✓ Negative - Order ID is not null or empty
    ✓ Negative - Status is not an invalid value
    ✓ Negative - Quantity is not zero or negative

  └ PATCH Update orders
    PATCH http://localhost:3000/orders/5 [200 OK, 442B, 226ms]
    ✓ Status code is 200
    ✓ Response time is under 2000ms
    ✓ Content-Type is application/json
    ✓ Negative - Status is not empty
    ✓ Negative - ID field remains consistent

  └ DELETE orders
    DELETE http://localhost:3000/orders/5 [200 OK, 442B, 106ms]
    ✓ Status code is 200
```

Users Assertions

Products Assertions

Orders Assertions

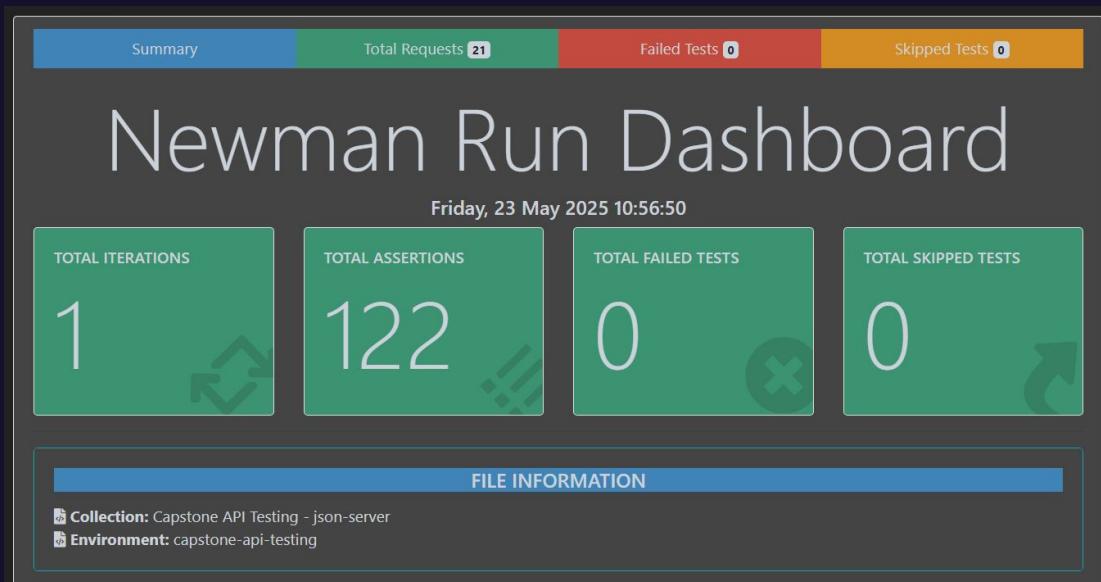
ASCENDION

API Testing Using Newman

	executed	failed
iterations	1	0
requests	21	0
test-scripts	18	0
prerequest-scripts	0	0
assertions	122	0
total run duration:	2.5s	
total data received:	3.99kB (approx)	
average response time:	39ms [min: 1ms, max: 226ms, s.d.: 56ms]	

ASCENDION

API Testing Using Newman - Reports



TIMINGS AND DATA		
SUMMARY ITEM	TOTAL	FAILED
Total run duration: 4.8s		
Total data received: 3.9KB		
Average response time: 144ms		
Requests	21	0
Prerequest Scripts	0	0
Test Scripts	18	0
Assertions	122	0
Skipped Tests	0	-

ASCENDION

JMeter

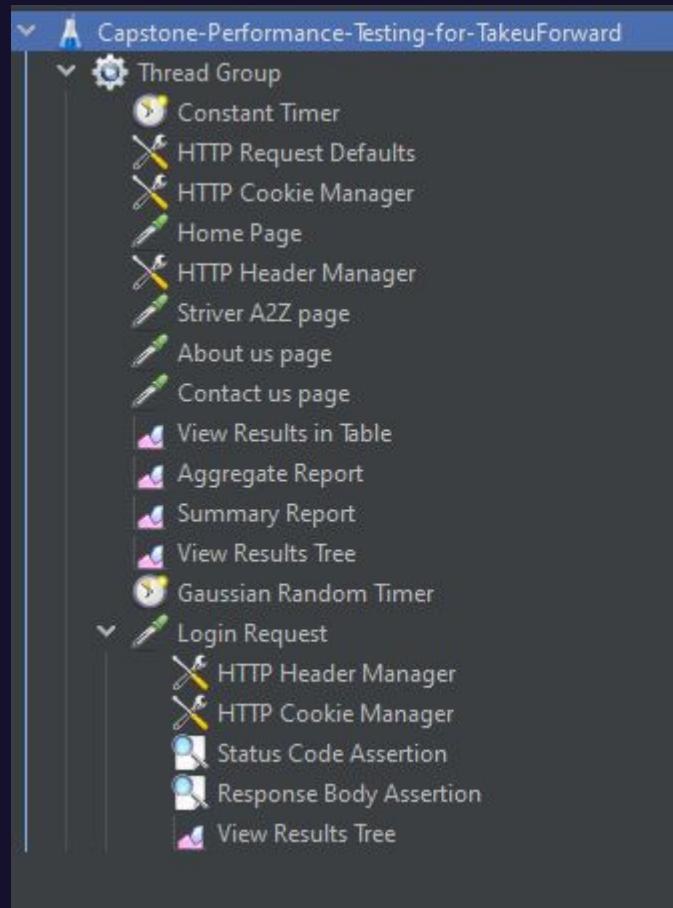
1) Tested 5 API endpoints of takeuforward.org (4 GET, 1 POST).

GET : Home Page, Contact-us, About-us, Striver A2Z sheet Page

POST: Login Page

2) Captured performance metrics and generated HTML reports

JMeter - Sampler



File Structure

This screenshot shows the "HTTP Request" configuration dialog. In the "Basic" tab, the "Name" field is set to "Striver A2Z page". The "Web Server" section has "Protocol [http]:" and "Server Name or IP: takeuforward.org". Under the "HTTP Request" section, the method is set to "GET" and the path is "/strivers-a2z-dsa-course/strivers-a2z-dsa-course-sheet-2/". Several checkboxes are present: "Redirect Automatically" (unchecked), "Follow Redirects" (checked), "Use KeepAlive" (checked), "Use multipart/form-data" (unchecked), and "Browser-compatible headers" (unchecked). The "Parameters" tab is selected, showing a table with two rows: "Name" and "Value". The "Send Parameters With the Request:" checkbox is checked. The "Body Data" and "Files Upload" tabs are also visible.

Sample GET request - to Striver A2Z sheets in TakeuForward

ASCENDION

JMeter - Sampler

HTTP Request

Name: Login Request

Comments:

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: backend.takeuforward.org

HTTP Request

POST Path: /api/auth/login

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

```
1 {
2   "loginCred": {
3     "email": "capstonetestuser01@gmail.com",
4     "password": "password@1234"
5   }
6 }
```

Content-Type	application/json
Origin	https://takeuforward.org
Referer	https://takeuforward.org
Accept	/

Sample Header

Sample POST request - to login in TakeuForward

ASCENDION

JMeter - Listeners

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename:

Search: Case sensitive Regular exp. Search Reset

Text Sampler result Request Response data

Response Body Response headers

```
{"success":true,"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImNhcnN0b25lQVzdHVzZXlODg4MDExfQ.YDJD0H1mNkRZf4QEzW8tKe_3sBn7SHNvMfoFrNpA","email": "capstone-testuser01@gmail.com"}
```

Contact us page
Login Request
Home Page
Striver A2Z page
Striver A2Z page
Striver A2Z page

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Home Page	20	1424	169	21064	4523.15	5.00%	14.8/min	0.88	0.05	3676.7
Striver A2Z page	20	114	57	334	96.24	0.00%	14.7/min	0.89	0.07	3724.4
About us page	20	81	55	143	23.63	0.00%	14.8/min	0.89	0.06	3722.5
Contact us page	19	115	59	1000	208.60	0.00%	15.6/min	0.95	0.06	3725.4
Login Request	11	2068	527	15600	4281.30	0.00%	10.4/min	0.30	0.07	1782.2
TOTAL	90	637	55	21064	2715.29	1.11%	49.1/min	2.78	0.22	3476.2

Summary Report

View Results Tree

JMeter - Reports

Apache JMeter Dashboard

Dashboard

Charts

Customs Graphs

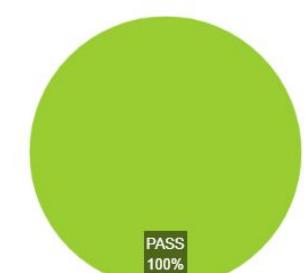
Test and Report information

Source file	"results.jtl"
Start Time	"5/25/25 12:12 AM"
End Time	"5/25/25 12:13 AM"
Filter for display	""

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.916	500 ms	1 sec 500 ms	Total
0.450	500 ms	1 sec 500 ms	Login Request
0.895	500 ms	1 sec 500 ms	Home Page
1.000	500 ms	1 sec 500 ms	Contact us page
1.000	500 ms	1 sec 500 ms	About us page
1.000	500 ms	1 sec 500 ms	Striver A2Z page

Requests Summary



PASS
100%

FAIL
PASS

ASCENDION

JMeter - Reports

ASCENDION

Jenkins

- 1) Automated Selenium test execution via Jenkins
- 2) Configured trigger for scheduled builds

```
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 100.5 s -- in TestSuite
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  01:49 min
[INFO] Finished at: 2025-05-26T02:26:34+05:30
[INFO] -----
Finished: SUCCESS
```

The screenshot shows the Jenkins dashboard for the project 'capstone_may2025'. The top navigation bar includes the Jenkins logo and the project name. Below the navigation is a sidebar with links: Status (highlighted), Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main area is titled 'Builds' and contains a table with three rows of build history. Each row includes a checkmark icon, the build number (#4, #5, or #6), the start time (2:09 AM, 2:19 AM, or 2:24 AM), and a dropdown arrow. A 'Filter' input field is also present.

	Build	Time	Actions
<input checked="" type="checkbox"/>	#6	2:24 AM	
<input checked="" type="checkbox"/>	#5	2:19 AM	
<input checked="" type="checkbox"/>	#4	2:09 AM	

ASCENDION

Github

- 1) Pushed all project code to GitHub
- 2) Used branches for each module/topic for better version control

Github Repository Link:

[madhanadithya/madhanadithya-Capstone-hp-training-may2025](https://github.com/madhanadithya/madhanadithya-Capstone-hp-training-may2025)

The screenshot shows a GitHub repository page for 'madhanadithya-Capstone-hp-training-may2025'. The repository is public and has 3 branches and 0 tags. The 'Code' tab is selected. The 'About' section contains the following text: 'In this repository I have attached codes and files related to my capstone project for HP training, which is held on May 2025.' Below the 'About' section, there are links to 'Readme', 'Activity', '0 stars', and '0 watching'.

About

In this repository I have attached codes and files related to my capstone project for HP training, which is held on May 2025.

Readme
Activity
0 stars
0 watching

Commit	Description	Time Ago
madhanadithya Merge pull request #2 from madhanadithya/features/selenium-java	added the collection and environmental files of postman a...	3 days ago
API-testing-json-server	updated selenium java codes for takeufoward website	16 minutes ago
Selenium-Java/TakeUforward	updated selenium java codes for takeufoward website	16 minutes ago
jmeter	updated selenium java codes for takeufoward website	16 minutes ago
README.md	Initial commit	3 days ago

Challenges Faced

- 1) While using json-server, when we create data using a POST request, the generated id is automatically converted from an integer to a string. This makes establishing relationships between endpoints difficult.

`GET {base_url}/orders/{id}?_expand=user&_expand=product`

- 2) session between the tests to maintain the workflow.

Conclusion

- I have learnt about end-to-end automation using Selenium for web UI, Postman and Newman for API testing, JMeter for performance testing, Jenkins for CI/CD integration, and Jira for managing sprints, test cases, and tracking execution.
- This project helped me understand how to design, execute, and manage real-time test scenarios effectively using industry-standard tools.

ASCENDION

Thank
You