

**ASCENDION**

# Capstone Project



## Personal BACKGROUND

(Name, Past Experience, Qualification, Career Summary)

---

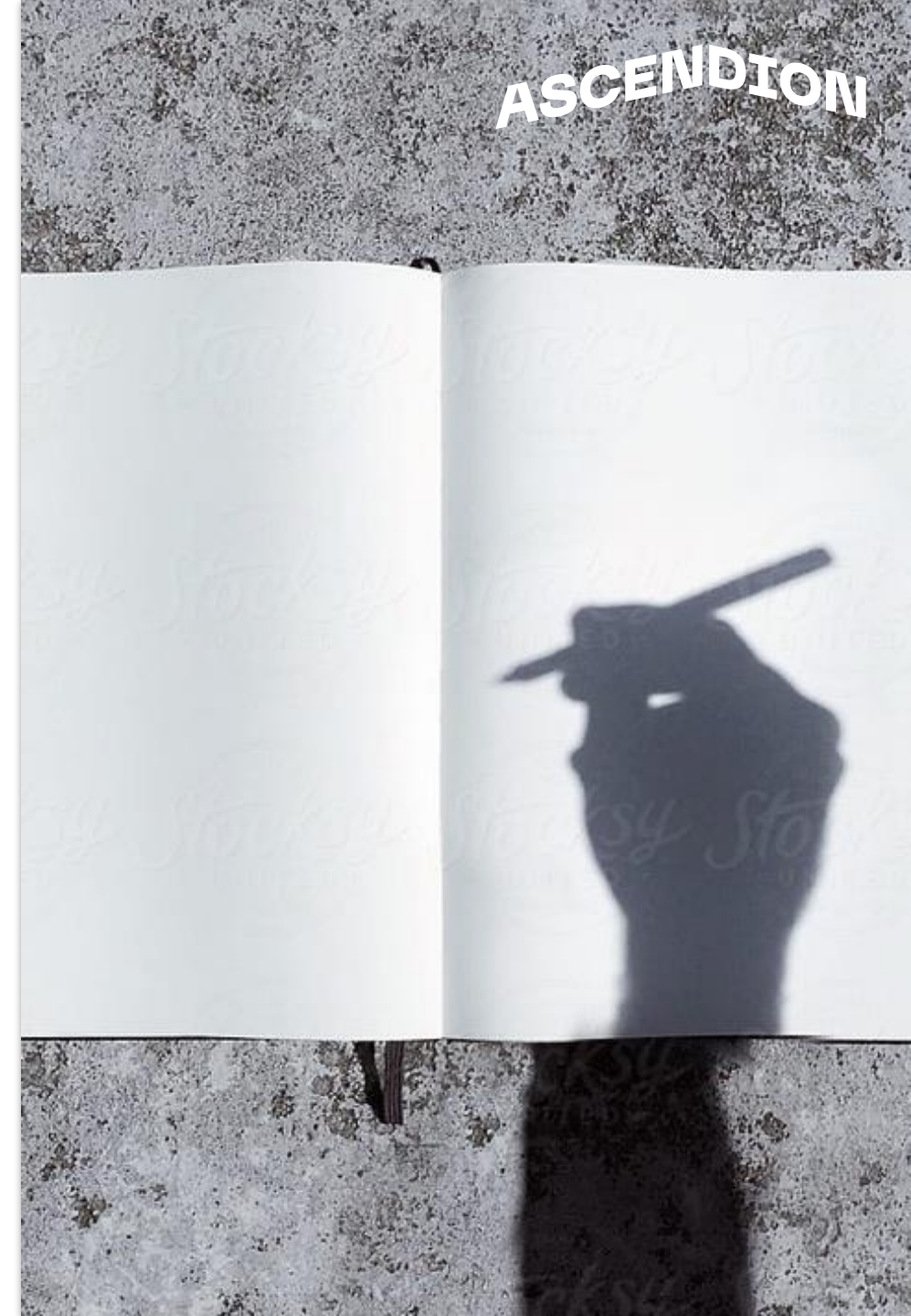
**Name : Prabhu Krishnamoorthy**

**Past Experience : 7.5 years**

**Qualification : Lead Engineer**

**Career Summary : Lead QA Engineer with 7.5 years of expertise across Functional Testing, UI Automation, API Automation, and Mobile Testing, ensuring high-quality, scalable test coverage.**

ASCENSION



## Key Takeaways/Learnings from the Program (HTD)

---

- **Strong foundation in building scalable automation frameworks**
  - Using Page Object Model (POM), reusable utilities, and clean coding practices.
- **Efficient UI automation with Selenium WebDriver**
  - Mastered locators (XPath, CSS, ID), waits, browser handling, and debugging dynamic elements.
- **Complete test execution workflow**
  - Implemented data-driven testing, assertions (JUnit/MSTest), and integrated tests with CI/CD.





## Problem Statement of the Capstone Project

---

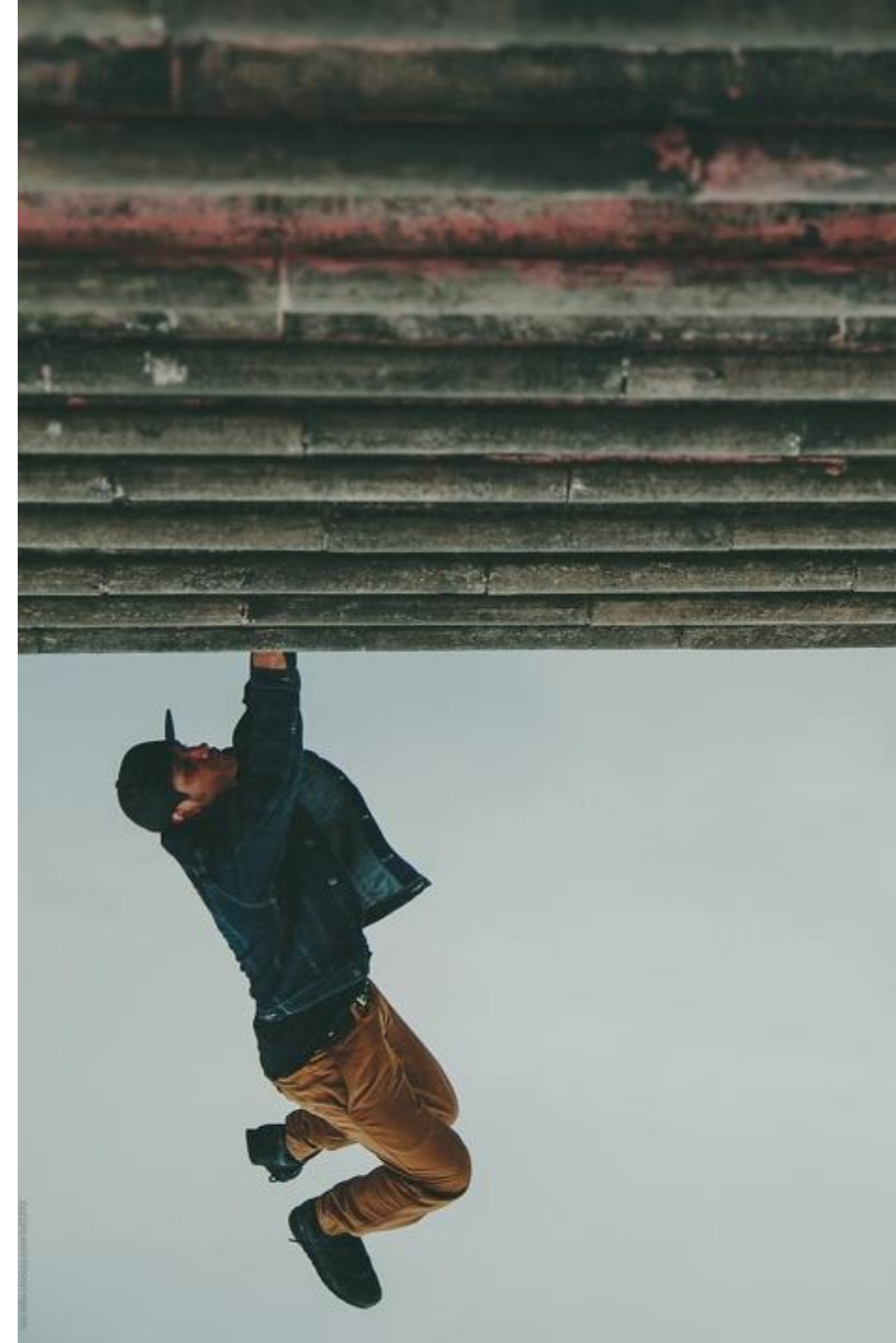
**Maintaining stable UI tests is challenging due to frequent application changes.**

This project overcomes the issue by using a strong Page Object Model (POM) with:

- Centralized locators
- Reusable action methods
- Explicit waits for dynamic elements

Basic automation projects often lack structure, reporting, and reusability. This project solves these gaps by building a scalable, industry-level framework using:

- **Driver Factory** for consistent browser setup
- **Excel-based Data-Driven Testing**
- **Extent Reports** for detailed execution insights
- **Modular, extensible design** for easy maintenance and growth



# ASCENDION

CreateUserTests.cs UpdatePutUserTests.cs GetUsersTests.cs DeleteUserTests.cs UpdatePatchUserTests.cs

CartAPI\_Testing

```
1 using System;
2 using Newtonsoft.Json;
3 using NUnit.Framework;
4 using RestSharp;
5 using CartAPI_Testing.Models;
6
7 namespace CartAPI_Testing.Tests
8 {
9     0 references
10    public class CreateUserTests : BaseTest
11    {
12        private const string BaseUrl = "https://fakestoreapi.com";
13
14        [Test]
15        0 references
16        public void POST_User_Should_Create()
17        {
18            var client = new RestClient(BaseUrl);
19            var request = new RestRequest("/users", Method.Post);
20
21            request.AddJsonBody(new
22            {
23                email = "test@example.com",
24                username = "testuser",
25                password = "testpass",
26                name = new { firstname = "John", lastname = "Doe" },
27                address = new
28                {
29                    city = "Delhi",
30                    street = "Main Road",
31                    number = 10,
32                    zipcode = "12345",
33                    geolocation = new { lat = "40.0", longi = "20.0" }
34                }
35            },
36            "application/json");
37        }
38    }
39 }
```

100% 0 2

Ln: 9, Ch: 34 (10 chars, 1 lines) SPC CRLF UTF-8 with BOM

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'CartAPI\_Testing' (1 of 1 project)

CartAPI\_Testing

Dependencies

Base

C# Base.cs

Models

C# User.cs

Reports

Reports

2025-12-09

ExtentReport.html

Tests

C# CreateUserTests.cs

C# DeleteUserTests.cs

C# GetUsersTests.cs

C# UpdatePatchUserTests.cs

C# UpdatePutUserTests.cs

Utils

Reporter

C# Configuration.cs

Test Explorer

5

5

0

Search (Ctrl+I)

Test run finished: 5 Tests (5 Passed, 0 Failed, 0 Skipped) run in 5.1 sec

0 Warnings 0 Errors

Test

▶ Run

▶ Debug

▶ Profile

▶ 

✓

 CartAPI\_Testing (5)

▶ 

✓

 CartAPI\_Testing.Tests (5)

▶ 

✓

 CreateUserTests (1)

▶ 

✓

 DeleteUserTests (1)

▶ 

✓

 GetUsersTests (1)

▶ 

✓

 UpdatePatchUserTests (1)

▶ 

✓

 UpdatePutUserTests (1)

Group Summary

CartAPI\_Testing

Tests in group: 5

Total Duration: 4.6 sec

Outcomes

✓

 5 Passed

HTTP Report

09-12-2025 10:53:36

Tests

POST\_User\_Should\_Create

10:53:36 / 1130.63 ms

Pass

DELETE\_User\_Should\_Return\_200

10:53:45 / 682.79 ms

Pass

GET\_All\_Users\_Should\_Return\_List

10:53:45 / 641.79 ms

Pass

PATCH\_User\_Should\_Update\_Username

10:53:46 / 557.73 ms

Pass

PUT\_User\_Should\_Update

10:53:47 / 538.88 ms

Pass

POST\_User\_Should\_Create

09-12-2025 10:53:36

09-12-2025 10:53:37

1130.63 ms

#test-id=1

STATUS	TIMESTAMP	DETAILS
Pass	Dec 09, 2025 10:53:37	Test Passed

# ASCENDION

---

```
CartAPI_Testing/
```

```
|
```

```
|— Models/
```

```
|
```

```
  └─ User.cs
```

```
|
```

```
|— Tests/
```

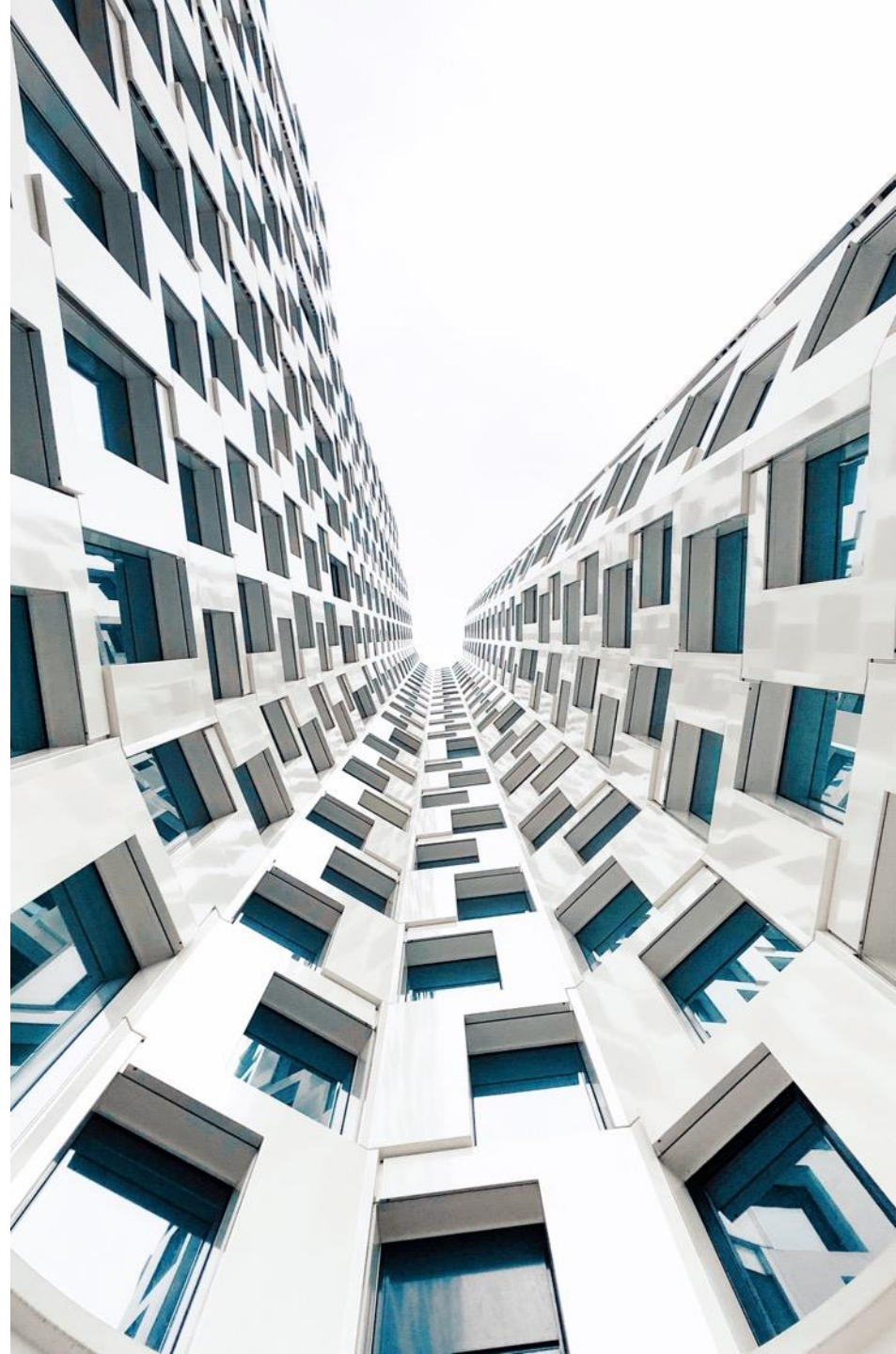
```
  └─ CreateUserTests.cs
```

```
  └─ GetUsersTests.cs
```

```
  └─ UpdatePutUserTests.cs
```

```
  └─ UpdatePatchUserTests.cs
```

```
  └─ DeleteUserTests.cs
```





# ASCENDION

---

## 1. GET

Fetches data from the server (read-only).

## 2. POST

Creates a new resource on the server.

## 3. PUT

Updates/replaces an entire resource.

## 4. PATCH

Partially updates a resource.

## 5. DELETE

Removes a resource from the server.





# Test Execution Report

File

C:/Users/prabhu.km/source/repos/Project%20Amazon/Project%20Amazon/bin/Debug/net10.0/Reports/2025-12-08/ExtentRepor...

🔍

🔊

☆

⚙️

☆

👤

...

🌈

🔍

Buy Laptop Flow

08-12-2025 20:06:55

Tests

Buy Iphone Flow

20:06:58 / 28893.46 ms

Pass


Buy Iphone Flow


08-12-2025 20:06:58


08-12-2025 20:07:27


28893.46 ms

#test-id=1

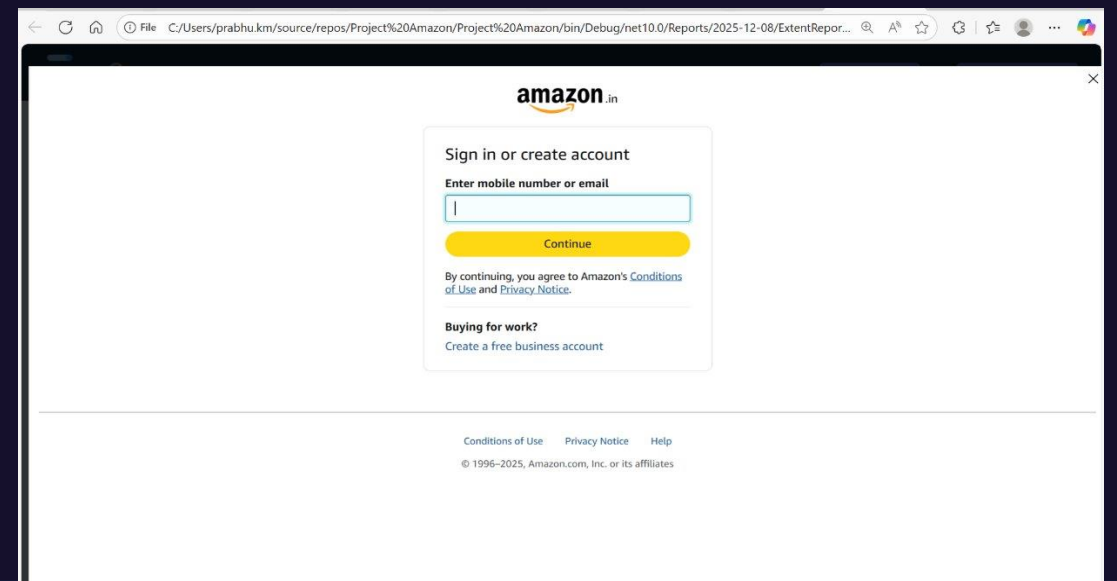
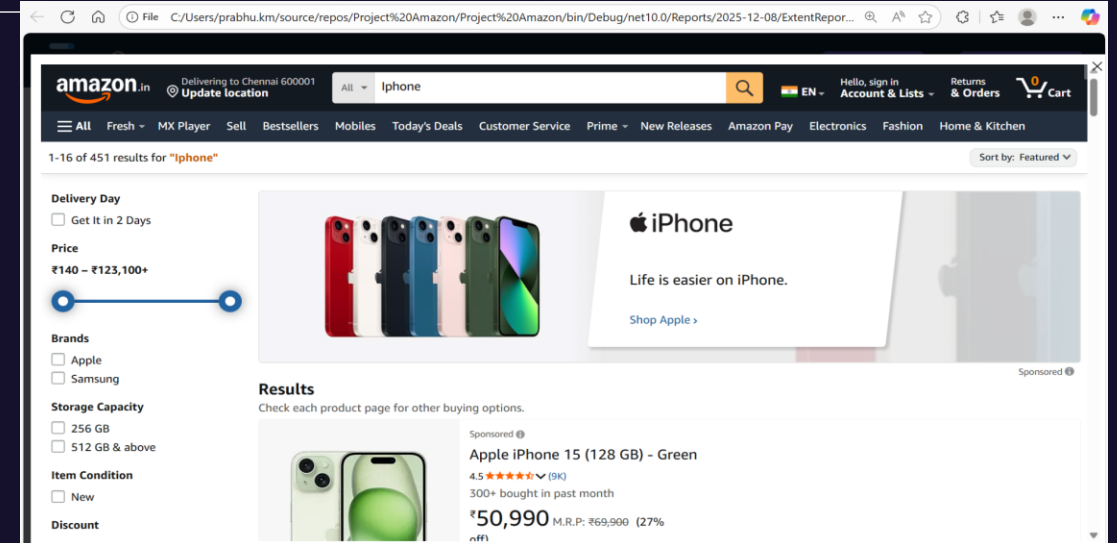
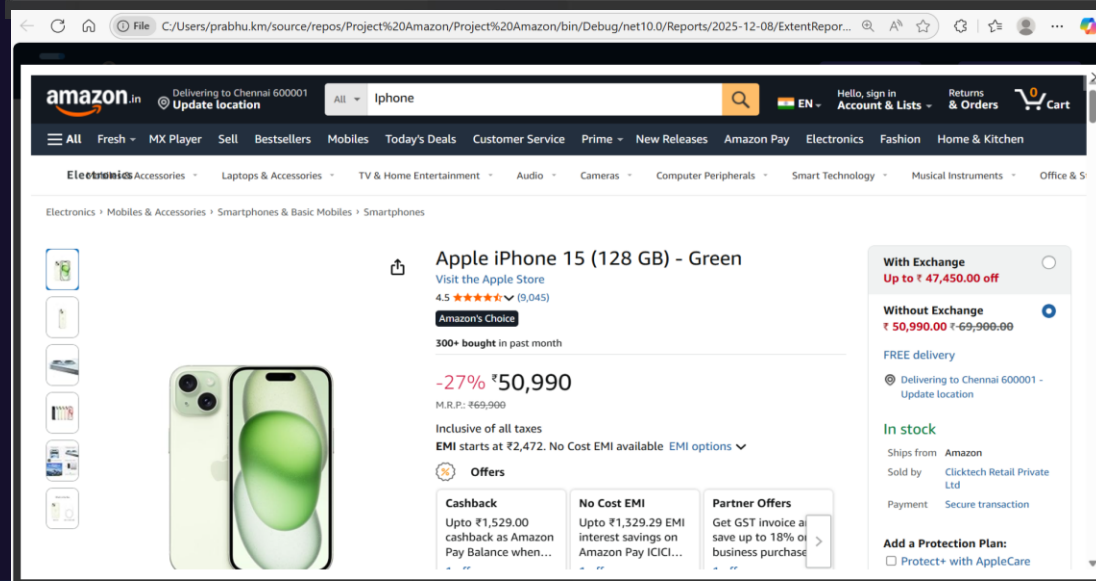
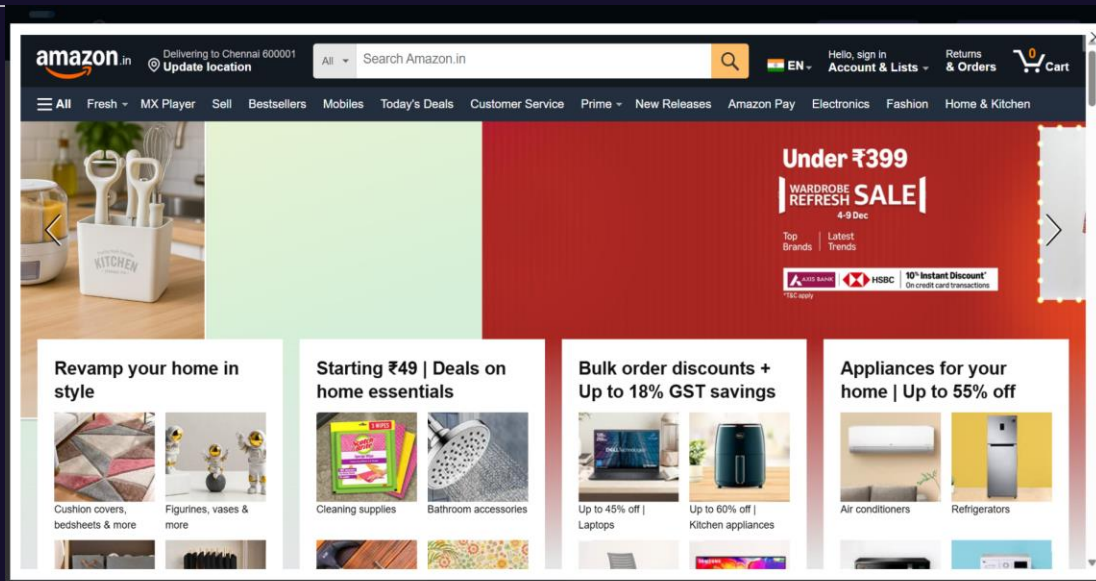








## Amazon Execution Screenshot



Execution Result

			Buy Laptop Flow		08-12-2025 20:06:55	
			STATUS	TIMESTAMP	DETAILS	
<div>Tests</div> <div>Buy Iphone Flow</div> <div>20:06:58 / 28893.46 ms</div> <div>Pass</div>			Info	Dec 08, 2025 20:07:03	Navigated to Amazon	
			Pass	Dec 08, 2025 20:07:05	Navigate to Amazon dashboard completed successfully	
			Info	Dec 08, 2025 20:07:11	Searched for Iphone	
			Pass	Dec 08, 2025 20:07:13	Search Product completed successfully	
			Info	Dec 08, 2025 20:07:14	Clicked first product	
			Pass	Dec 08, 2025 20:07:16	Click First Product of Iphone completed successfully	
			Info	Dec 08, 2025 20:07:17	Switched to product tab	
			Pass	Dec 08, 2025 20:07:19	Click Buy Now option completed successfully	
			Pass	Dec 08, 2025 20:07:21	Click Buy Now option completed successfully	
			Info	Dec 08, 2025 20:07:23	Clicked Buy Now	
			Info	Dec 08, 2025 20:07:23	Enter email address	
			Info	Dec 08, 2025 20:07:25	Clicked the continue Button	
			Pass	Dec 08, 2025 20:07:27	Buy Laptop Flow completed successfully	



## Conclusion

---

- Selenium C# ensures reliable UI automation for Amazon's key user journeys.
- RestSharp provides efficient validation of backend APIs using GET, POST, PUT, PATCH, and DELETE.
- Combining UI and API tests offers stronger coverage and faster defect detection.
- This integrated approach improves test quality, stability, and overall system reliability.



Thank You

ASCENDION