

## Capstone Project

### Note:

Please attach 2-3 screenshots as proof of concept in the presentation deck. These visuals should clearly demonstrate the key functionality or results of the concept being presented.

### Capstone Project Instructions

Please clearly distinguish between mandatory and optional exercises as part of the Capstone project. Follow the guidelines below:

#### ✔ Mandatory Exercises

These are required components that must be completed to successfully pass the Capstone project. Ensure that all mandatory tasks are fully implemented and documented. Provide 2-3 screenshots as proof of concept in the final presentation deck.

#### ⚙ Optional Exercises

These are additional tasks that can enhance your project but are not required for completion. Completing optional tasks may add extra value and demonstrate advanced understanding.

### \* JIRA dashboard (mandatory)

Sprint  
Backlog  
test case execution ---- Zypher/X-ray  
execution test result screenshot  
Dashboard

### 1.) Selenium Web UI Test Automation (Java + TestNG): (mandatory)

Create basic tests for login, form submission, or search functionality -  
design more than 2 functional test case (choose test scenario)  
Use Page Object Model (POM) for maintainability.  
Add assertions and logging (log4j).  
Capture Screenshot if test failed.  
Execute test from TestNG.xml/POM.xml/maven goals --- mvn test/ mvn clean install  
design code for headless or headed mode ---- value fetch from configfile

\*\*\*\* optional

Helper.java  
customMethod  
  
ClickOnElementUsingJS(ele)  
CaptsureScreenshot(driver, fileName)

design code for crossBrowserTest ---- Edge/chrome browser  
Action Example  
IFrameExample  
windowsExample  
AlertExample

### Reference SampleApp:

https://www.hyrtutorials.com/p/add-padding-to-containers.html  
https://accounts.lambdatest.com/register  
https://ecommerce-playground.lambdatest.io/index.php?route=account/login  
https://opensource-demo.orangehrmlive.com/web/index.php/auth/login  
https://www.amazon.in/  
https://admin-demo.nopcommerce.com/login?ReturnUrl=%2Fadmin%2F

-----  
2.) Appium Mobile Automation on BrowserStack: (\*\* optional)

-----  
Set up Appium tests for Android/iOS apps (or mobile web).  
Integrate with BrowserStack using access credentials.  
Validate a mobile login flow or navigation.  
Configure desired capabilities dynamically via test config/JavaClass file.

VirtualDevice  
-----

Calculator/DemoApp (APK file)

ios device/Android/Tab/MACOS

Install the DemoApp  
get the appPackage and appActivity using ADB shell commands

- 1.) Identify element using UIAutomator/Appium Inspector
- 2.) e.g. Verify Add test on calculator
- 3.) BaseTest --- TestCase
- 4.) Execute test from testNG.xml/execute using maven goals

-----  
3.) API Automation with Postman and Newman: (Mandatory)  
-----

Design Postman collection to test APIs (GET/POST/PUT/PATCH/DELETE) ---- 2 API TEST Method  
Export collection.  
Use Newman CLI for local (newman run Collectionfilename) and Jenkins execution (optional):

validation  
-----

- 1.) StatusCode
- 2.) Response Body:  
ex: Apple iPhone 12 Mini, 256GB, Blue
- 3.) ResponseTime  
ex: 2000msec
- 4.) Response Header  
ex: Content-Type ---- JSON

execute test from command Prompt using NEWMAN

-----  
newman run fileName

extra:

newman run fileName -r htmlextra

SampleAPI:  
-----

https://reqres.in/  
https://restful-api.dev/  
https://demoqa.com/BookStore/v1/Books  
https://fakestoreapi.com/docs  
https://dummyjson.com/docs/users#users-all  
https://restful-api.dev/  
https://jsonplaceholder.typicode.com/  
https://restful-booker.herokuapp.com/booking

-----  
4.) Performance Testing with JMeter: (\*\*\*\* optional)  
-----

performanceTesting on API

performance testing on WebApp

testScenario:

user: 10  
Ramp-up: 1sec  
Delay: 10 sec  
Duration: 120 sec

Create a .jmx script simulating concurrent users for API or web.  
Use CLI for execution:  
Run JMeter test from command Prompt and generate HTML reports

ex:

CMD ---- > JMeter.bat -n -t "Jmx file" -l "CreateLogFile" -e -o "HTMLReports"

---

5. GitHub Integration: (mandatory)

---

Push entire project to a GitHub repository.  
Use branches for feature development and main for CI builds.

GitHUB Account  
Create a gitHUB Repo

ex: CAPSTONE\_COGNIXIA\_WEB\_API\_MAY2025

Selenium and API

---

6. Jenkins Pipeline Integration: (optional)

---

Configure Jenkins to:  
Pull code from GitHub.  
Execute Selenium tests via Maven/Gradle.  
Run Appium tests on BrowserStack.  
Trigger Postman tests using Newman.  
Run JMeter script and collect HTML reports.

Sceanrio

JOB: SeleniumWeb (generic project or Maven project)

plugin: git integration

- 1.) Integrate with GitHub account
- 2.) Trigger  
poll SCM : H/2 \* \* \* \*
- 3.) Build Step:  
execute test using mvn test or test

=====