American International University-Bangladesh (AIUB)

Department of Computer Science Faculty of Science &Technology (FST) Summer 23-24

Section: A

Software Quality Assurance and Testing

Automation Testing with Selenium

A Report submitted

By

SN	Student Name	Student ID
1	Md Sahadul Haque	21-45417-3
2	Mahamoda Akter	20-44269-3
3	Nokibul Arfin Siam	21-44793-1
4	Jahir islam	20-43882-2

INDEX

Section	Pages
1. Install Selenium Web-Driver and Mocha	2 - 5
2. Test NodeJS Project	6
3. Test Cases and Results	
3.1. Home Page	6
3.2. Signup	6
3.2.1. Signup: Pass	7 - 8
3.2.2. Signup: Fail	9 - 12
3.3. Login	13
3.3.1. Login: Pass	13-14
3.3.2. Login: Fail	15-16
3.4. Forget Password	17
3.4.1. Forget Password: Pass	17 – 18
3.4.2. Forget Password: Fail	19 - 20
3.5. Subscribe	21
3.5.1. Subscribe: Pass	21 – 22
3.5.2. Subscribe: Fail	23 – 24
3.6. Profile	25
3.6.1. Profile: Pass	25 – 26
3.6.2. Profile: Fail	27 – 28
3.7. Logout	29
4. Acknowledgement	30

1. Install Selenium WebDriver and Mocha:

Selenium and Mocha are tools designed for testing purposes:

- **Selenium:** A web testing framework used for automating web browsers. It is commonly used for testing web applications and automating web tasks.
- **Mocha:** A JavaScript test framework that runs on Node.js, designed for testing asynchronous code. It is typically used for unit testing in software development.

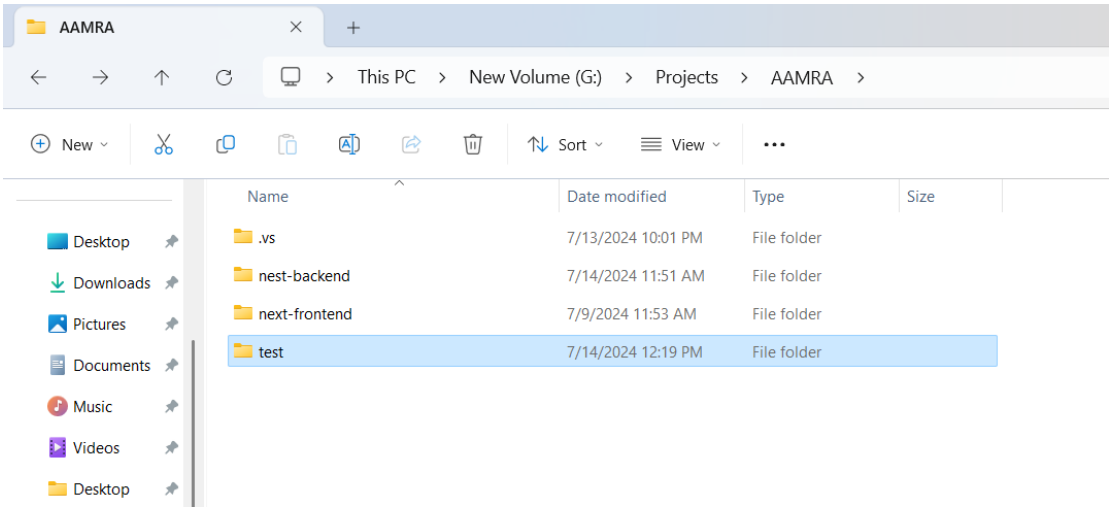

Figure1: create folder called "test"

Step1: Create a new folder in any drive. (the test folder is created inside the project directory)

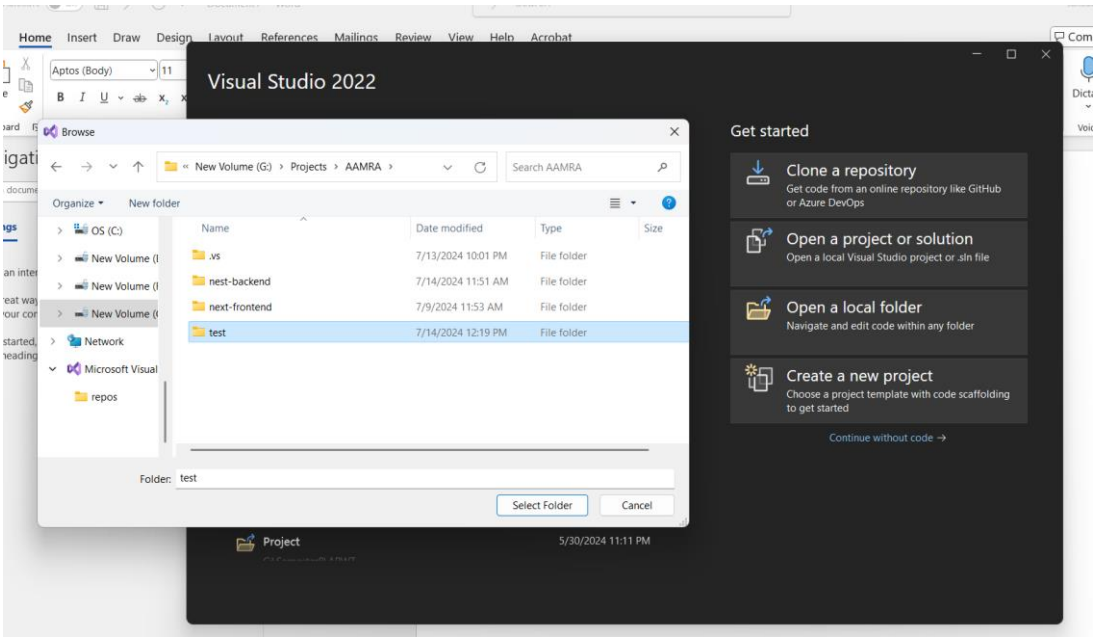

Figure2: test folder is opened in VS code

Step2: Open the folder in VS code

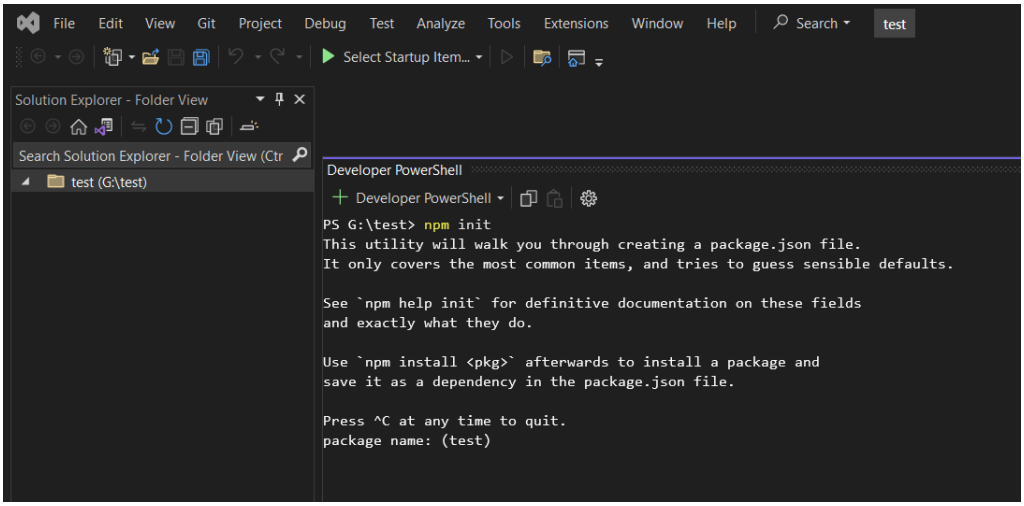

Figure3: initializing the test folder

Step3: Open the folder in the terminal and run ***"npm init"*** command to initialize the program. There will be some information about authentication and dependencies which it is optional to be filled in. A description and git repository can be assigned there.

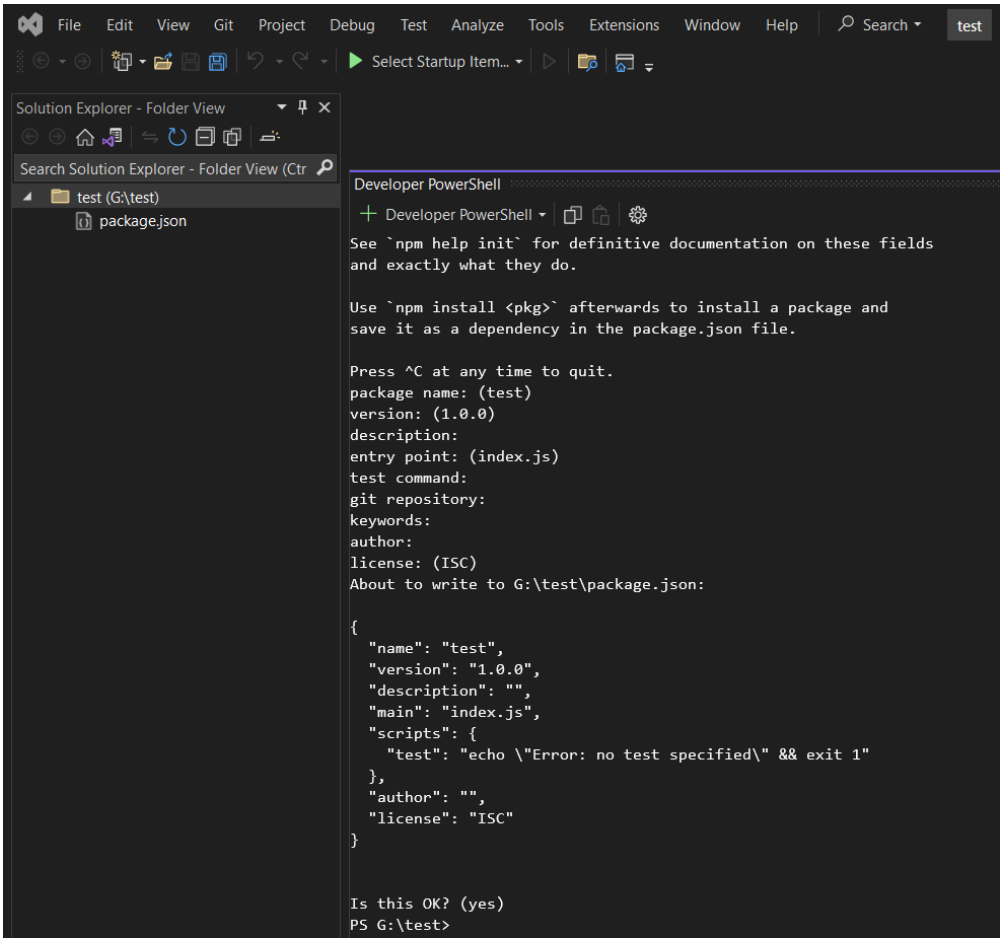

Figure4: package.json file is created

It will create a JSON file after confirming the initialization.

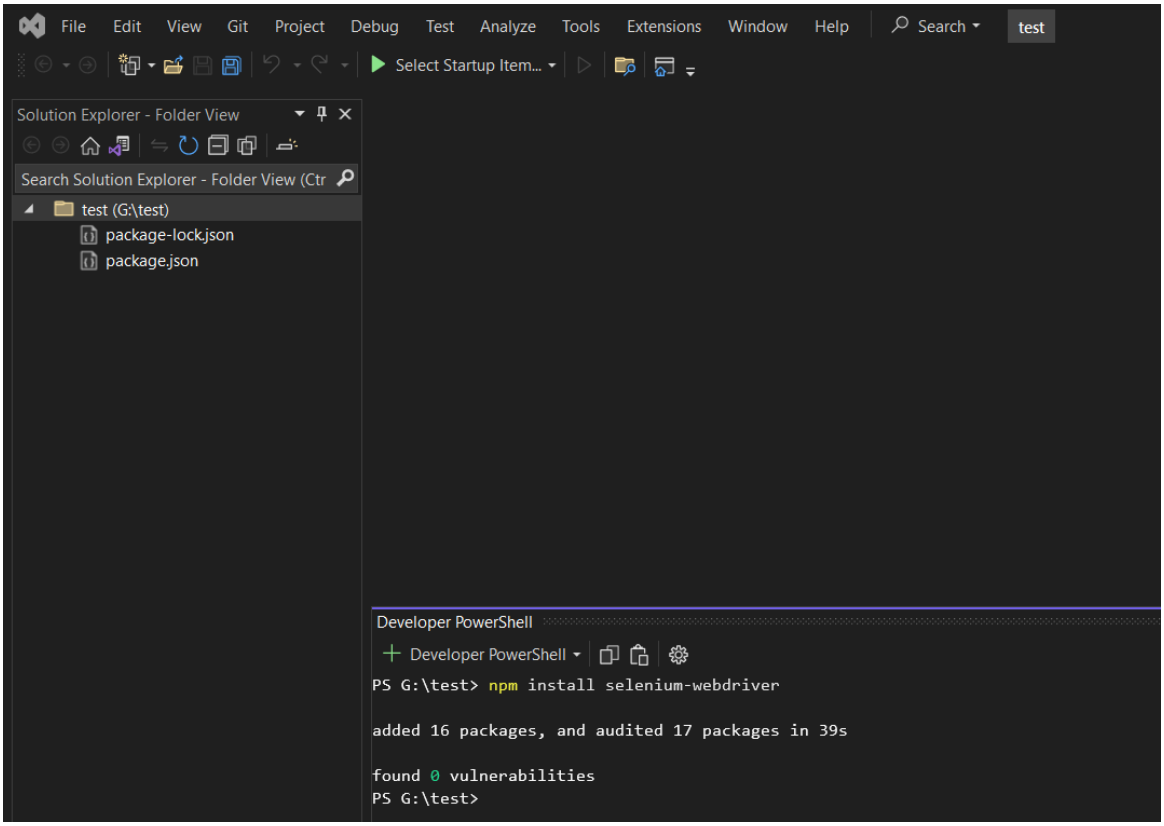

Figure5: installation of selenium web-driver

Step4: Install selenium-webdriver by the command ***"npm install selenium-webdriver"***. A detailed documentation for installation can be found at: selenium-webdriver - npm (npmjs.com).

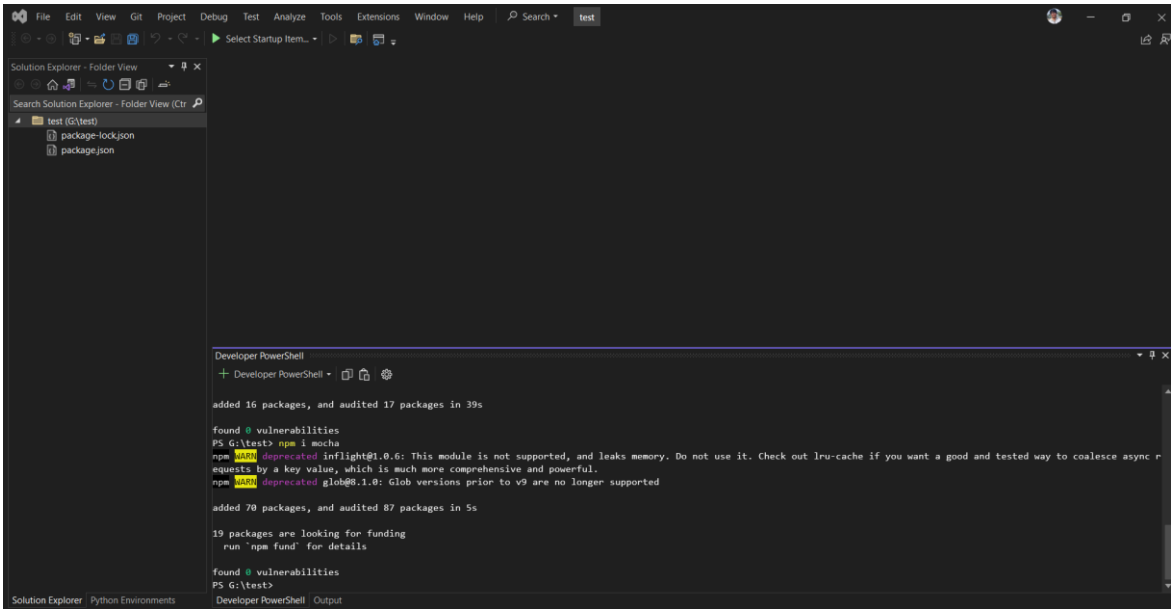

Figure6: installation of mocha

Step5: Install a node package mocha by the command ***"npm install mocha"***. A detailed documentation can be found at: mocha - npm (npmjs.com).

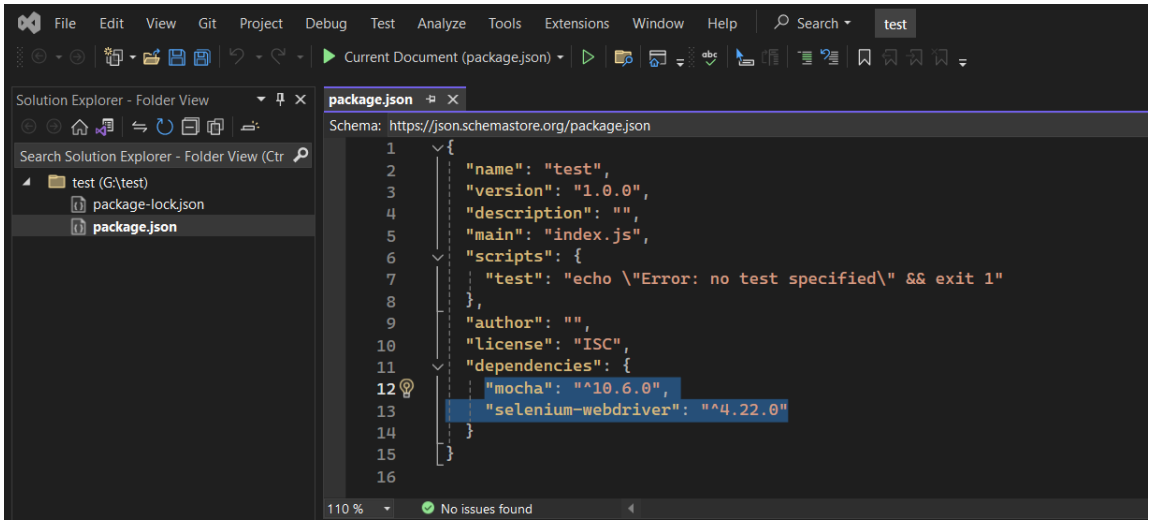

Figure7: dependencies are added after installing

To confirm the installation, check the JSON file and find the dependencies. If the dependencies are added, then both selenium web-driver and mocha are installed successfully.

2. Test NodeJS project:

Step1: select a NodeJS project and start the project.

It can be done by opening the backend and frontend folder in the terminal. In the backend folder terminal write command "***npm run start:dev***" and in the frontend folder write command "***npm run dev***" to start the project.

Step2: Identify all input, button, and other action tag elements by their ids to test.

It can be done by opening the local host in the browser and selecting the action elements in inspect mode.

Step3: Write the code for the elements to be tested.

Step4: Open the test folder in the terminal and run command "***npx mocha filename***" or "***mocha filename***"

Write the ***filename*** i.e. ***home.spec.js*** individually to test the functionality.

It is to notify that mocha can test any website in different browsers and the process of testing the input and action elements are extremely fast and flexible. In this case of testing, the whole project is not tested at once, rather the pages are tested individually after creating the test cases for some selected functions (unit testing). However, the project's eternity can be tested by combining the code in a single file and running it. As this testing task is divided among the group members, it is simpler to write the code in parts by all the individuals.

3. Test Cases and results:

3.1. Testing the project home page:

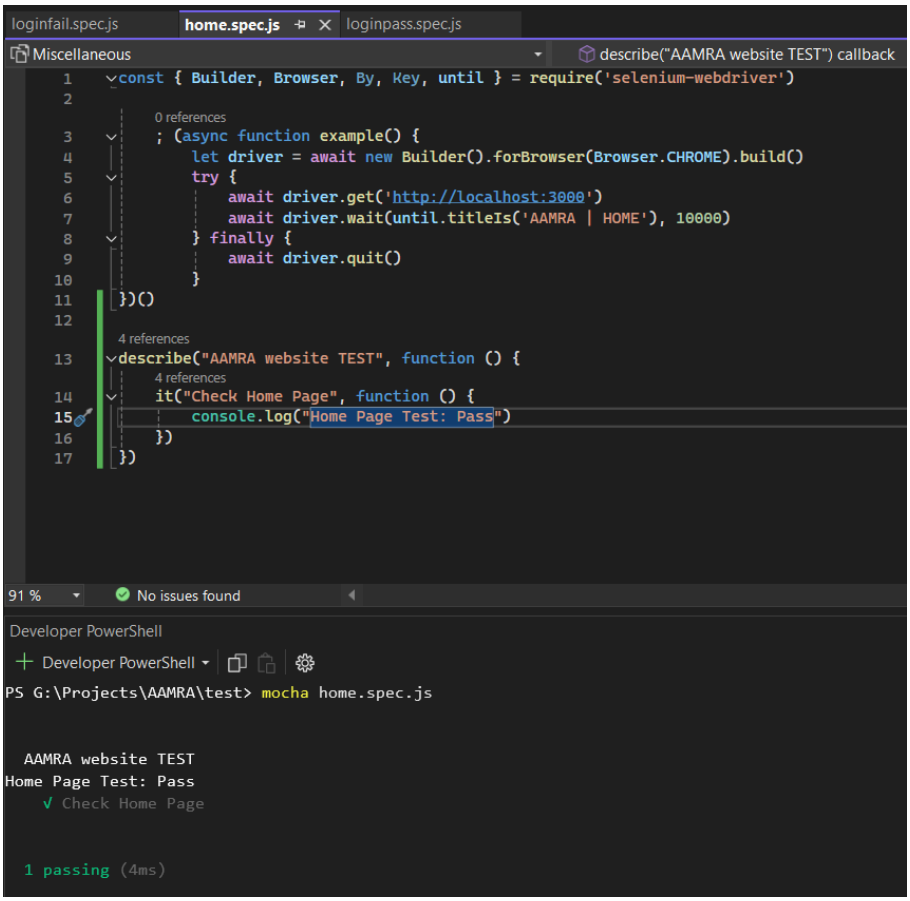

Figure8: code and result for home page

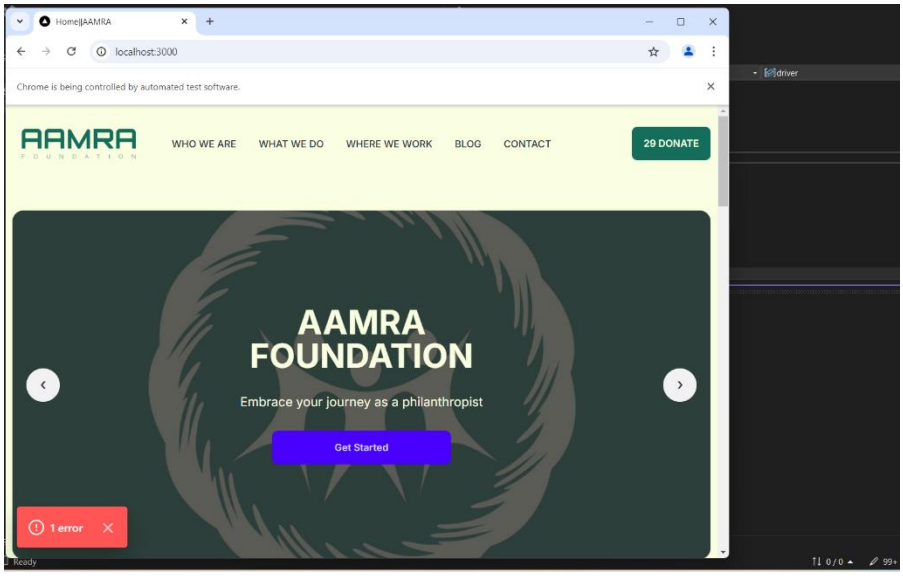

Figure9: output of the test

3.2. Signup

3.2.1. Signup: Pass

Passed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_1		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): High		**Test Executed by:** Nokibul Arfin Siam		
Module Name: User Registration		**Test Execution date:** 10/07/24		
Test Title: User Registration- Successful Signup with valid credentials				
Description: To verify users after successful registration				
Precondition: Must not registered before with an existing email address				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Enter name, age, gender, department, address, email, password, confirm password **3.** Click Signup	Name: Test Demo Age: 30 Gender: male Department: CSE Address: Sample Address Email: abc@gmail.com Password: Password123 Confirm Password: Password123	Registration Successful	Registration Successful	Pass
Post Condition: User can login and a profile will be created for the user				

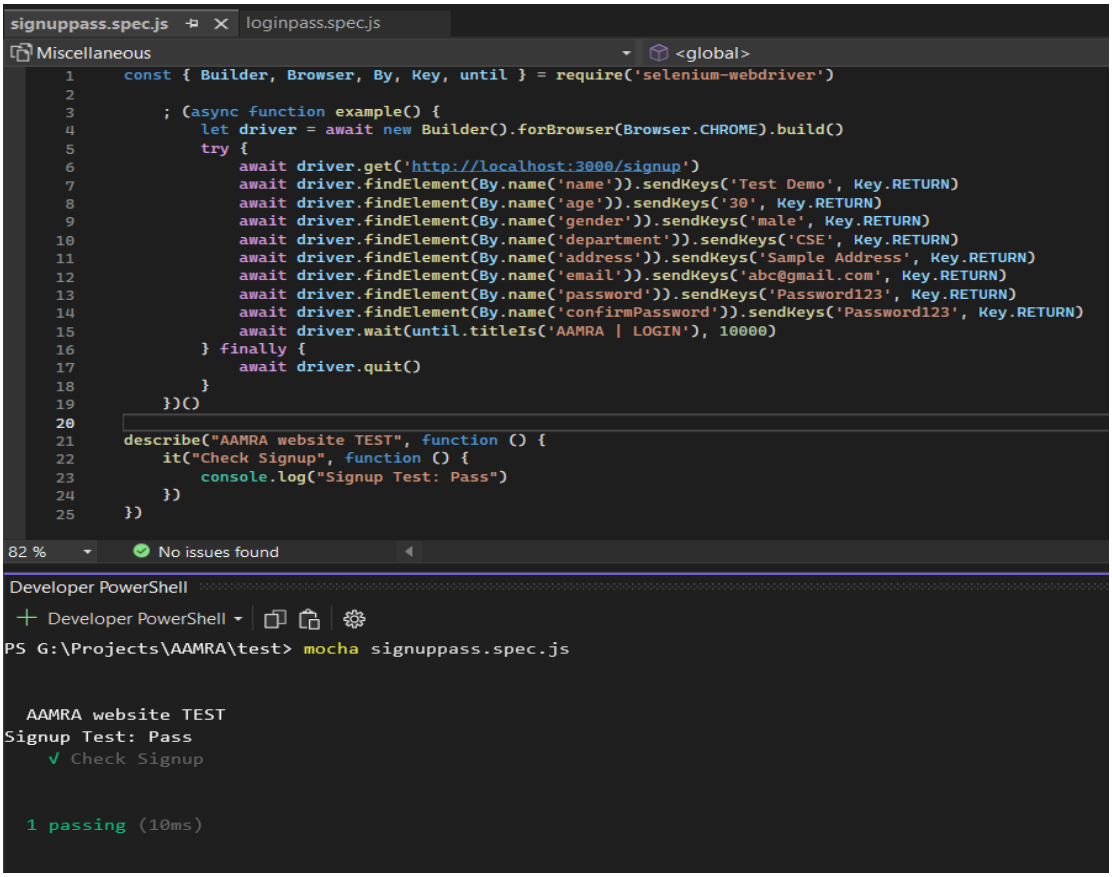

Figure10: code and result for Signup page: Passed

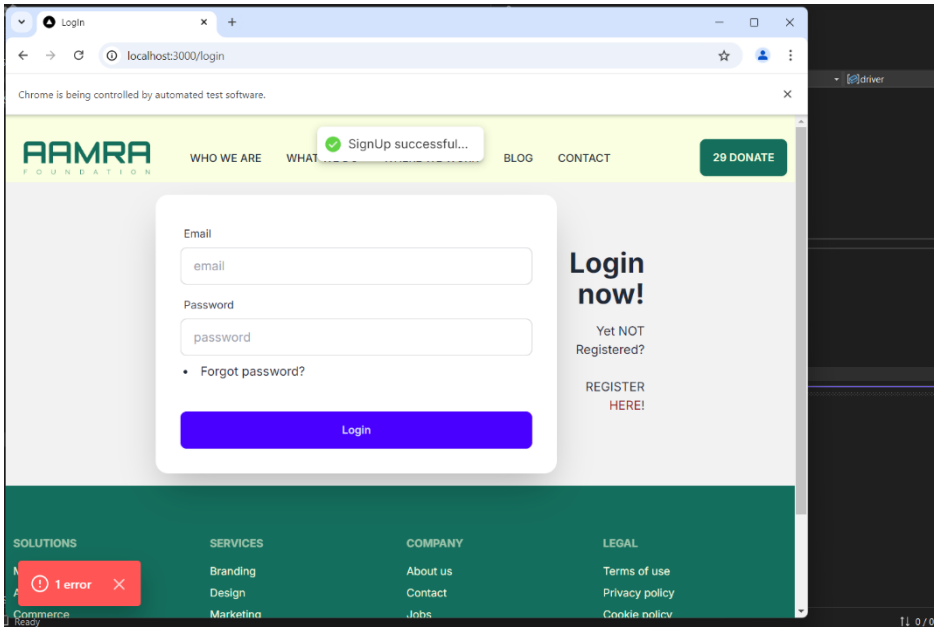

Figure11: output of the test

3.2.2. Signup: Fail

Failed test case (Validation Error):

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_2		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): High		**Test Executed by:** Nokibul Arfin Siam		
Module Name: User Registration		**Test Execution date:** 10/07/24		
Test Title: User Registration- Successful Signup with valid credentials				
Description: To verify users after successful registration				
Precondition: Must not registered before with an existing email address				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Enter name, age, gender, department, address, email, password, confirm password 3. Click Signup	Name: Test Demo Age: -30 Gender: male Department: CSE Address: Sample Address Email: abc2@gmail.com Password: Password123 Confirm Password: Password123	Registration Unsuccessful	Registration Unsuccessful	Fail
Post Condition: Users need to correct the input value and Signup again				

loginfail.spec.js | signupfail.spec.js | signuppass.spec.js | loginpass.spec.js

Miscellaneous | describe("AAMRA website TEST") callback

```
const { Builder, Browser, By, Key, until } = require('selenium-webdriver')

0 references
; (async function example() {
    let driver = await new Builder().forBrowser(Browser.CHROME).build()
    try {
        await driver.get('http://localhost:3000/signup')
        await driver.findElement(By.name('name')).sendKeys('Test Demo', Key.RETURN)
        await driver.findElement(By.name('age')).sendKeys('-30', Key.RETURN)
        await driver.findElement(By.name('gender')).sendKeys('male', Key.RETURN)
        await driver.findElement(By.name('department')).sendKeys('CSE', Key.RETURN)
        await driver.findElement(By.name('address')).sendKeys('Sample Address', Key.RETURN)
        await driver.findElement(By.name('email')).sendKeys('abc@gmail.com', Key.RETURN)
        await driver.findElement(By.name('password')).sendKeys('Password123', Key.RETURN)
        await driver.findElement(By.name('confirmPassword')).sendKeys('Password123', Key.RETURN)
        await driver.wait(until.titleIs('AAMRA | LOGIN'), 10000)
    } finally {
        await driver.quit()
    }
})()

4 references
describe("AAMRA website TEST", function () {
    4 references
    it("Check Signup", function () {
        console.log("Signup Test: Fail")
        throw Error("Test Failed: age can not be negative value!")
    })
})
```

74 % | No issues found

Developer PowerShell

Developer PowerShell

```
1 failing

1) AAMRA website TEST
     Check Signup:
   Error: Test Failed: age can not be negative value!
    at Context.<anonymous> (signupfail.spec.js:24:15)
    at process.processImmediate (node:internal/timers:478:21)
```

Figure12: code and result for Signup page: Failed (age validation)

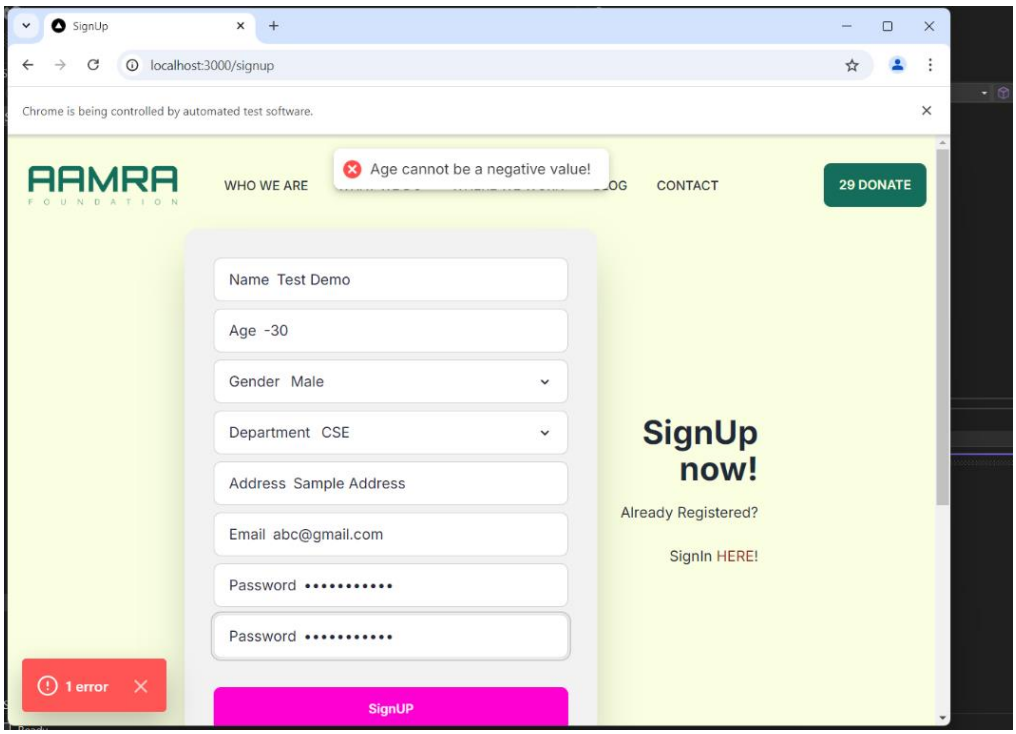

Figure13: output of the test

Failed test case (Already Registered Error):

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_3		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): High		**Test Executed by:** Nokibul Arfin Siam		
Module Name: User Registration		**Test Execution date:** 10/07/24		
Test Title: User Registration- Successful Signup with valid credentials				
Description: To verify users after successful registration				
Precondition: Must not registered before with an existing email address				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Enter name, age, gender, department, address, email, password, confirm password **3.** Click Signup	Name: Test Demo Age: 30 Gender: male Department: CSE Address: Sample Address Email: abc@gmail.com Password: Password123 Confirm Password: Password123	Registration Unsuccessful	Registration Unsuccessful	Fail
Post Condition: Users need to provide an unregistered email and signup again				

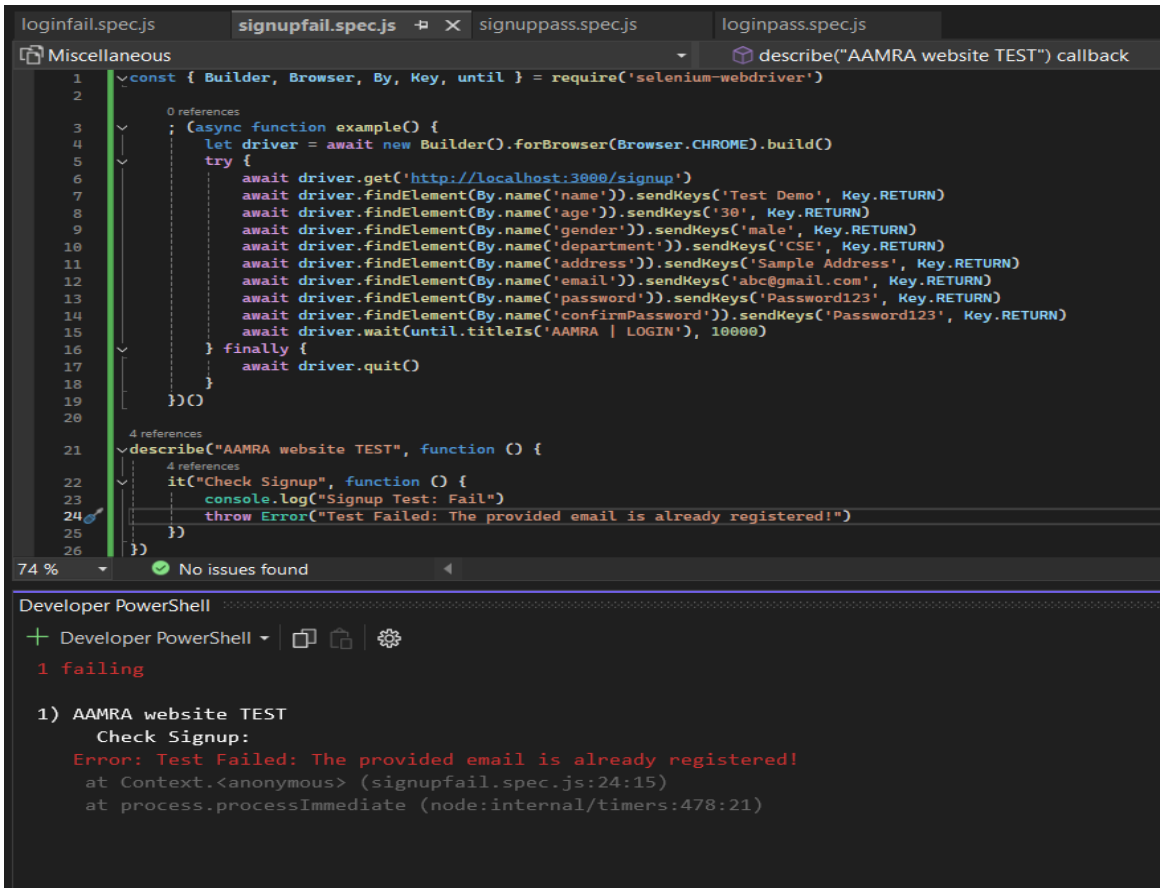

Figure14: code and result for Signup page: Failed (existing mail)

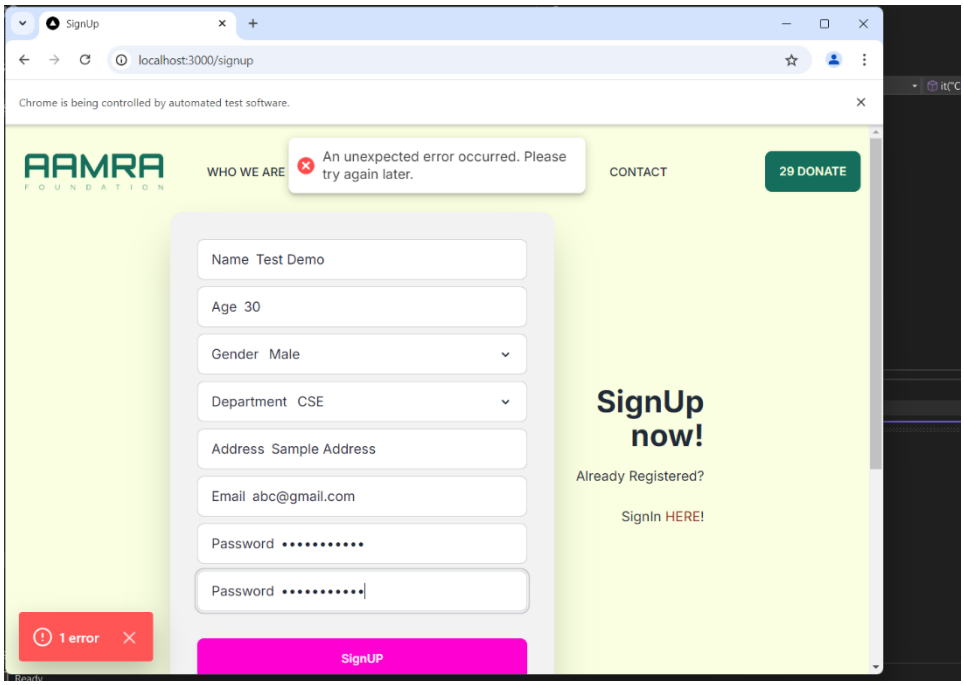

Figure15: output of the test

3.3. Login

3.3.1. Login: Pass

Passed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_4		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Medium		**Test Executed by:** Mahamoda Akter		
Module Name: User Login		**Test Execution date:** 12/07/24		
Test Title: Successful user login				
Description: Users should login to the site if the users are valid				
Precondition: Must Register to the site				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Emter email and password 3. Click Login	Email: sahadul80@gmail.com Password: Password123	Login successful	Login successful	Pass
Post Condition: Users should be able to view their profile dashboard				

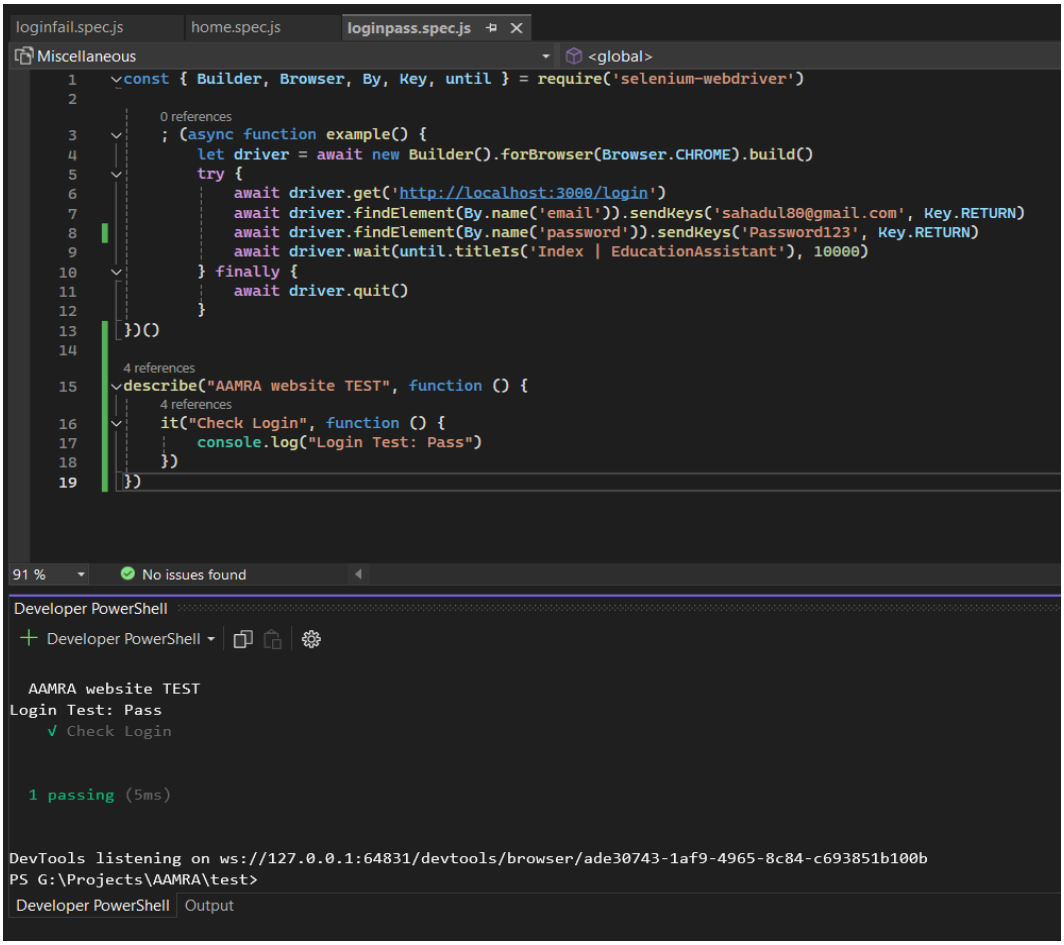

Figure16: code and result for login page: Passed

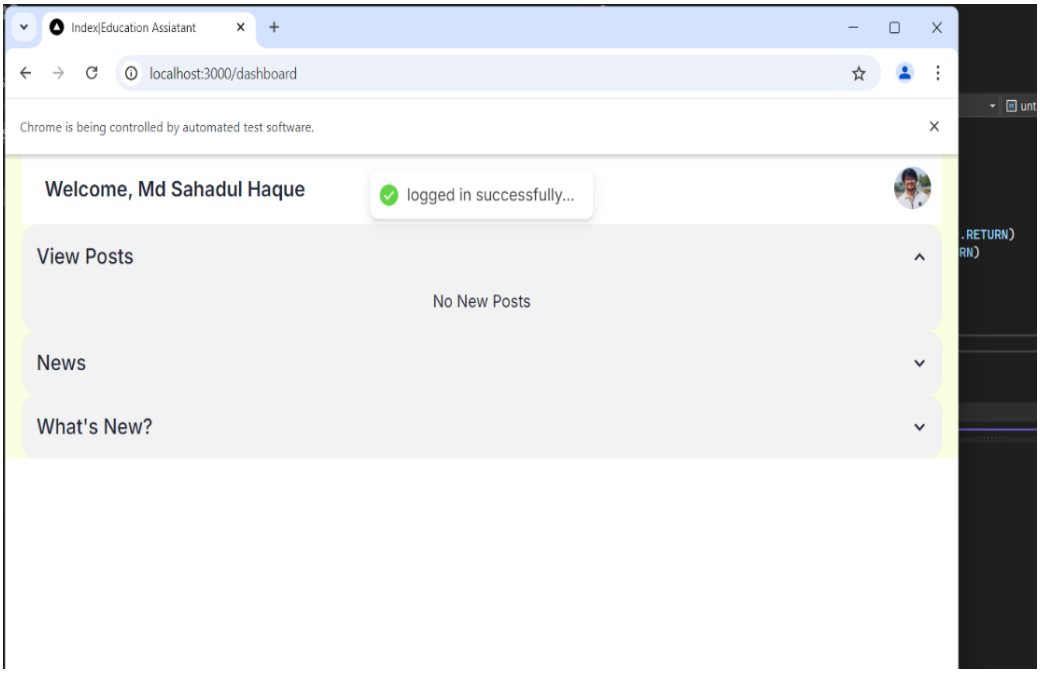

Figure17: output of the test

3.3.2. Login: Fail

Failed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_5		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Medium		**Test Executed by:** Mahamoda Akter		
Module Name: User Login		**Test Execution date:** 12/07/24		
Test Title: Successful user login				
Description: Users should login to the site if the users are valid				
Precondition: Must Register to the site				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Emter email and password **3.** Click Login	Email: sahadul80@gmail.com Password: Password125	Login unsuccessful	Login unsuccessful	Fail
Post Condition: Users should not be able to procced and need to login with valid credentials or forget password				

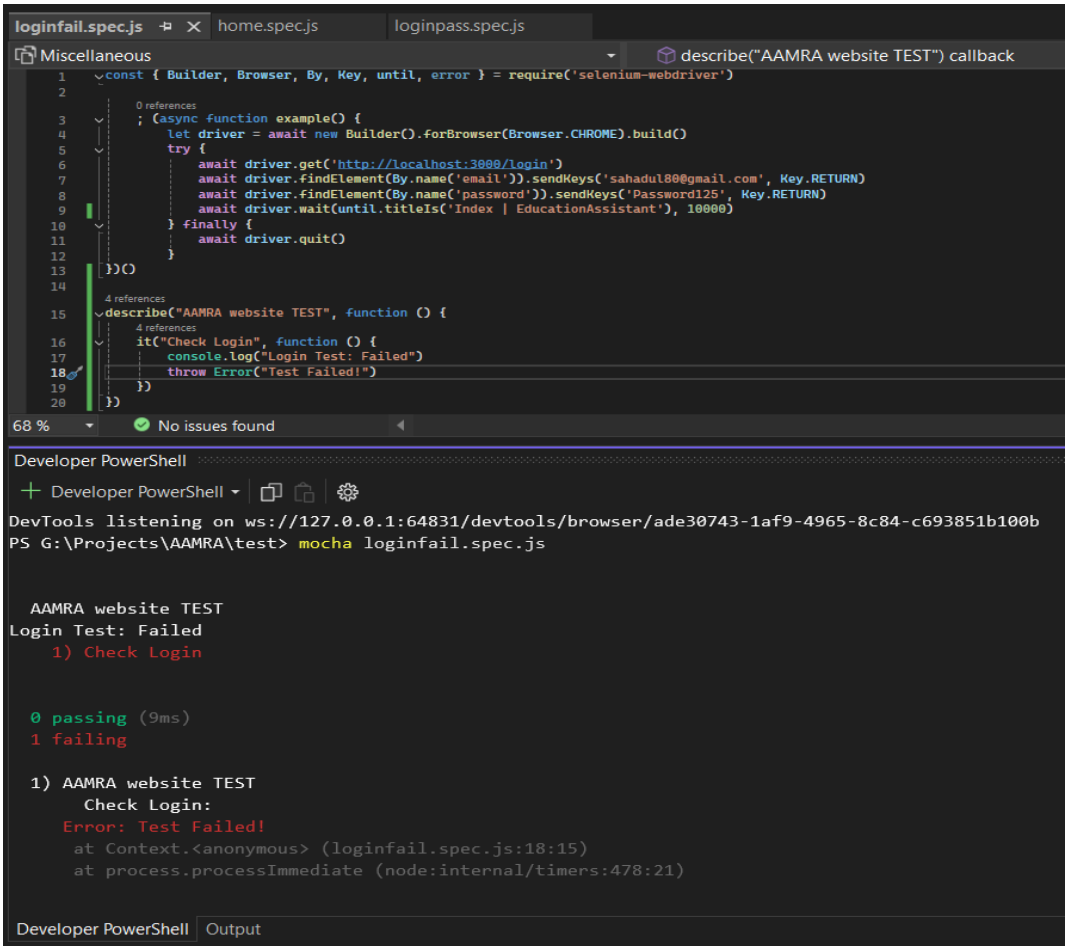

Figure18: code and result for login page: Failed

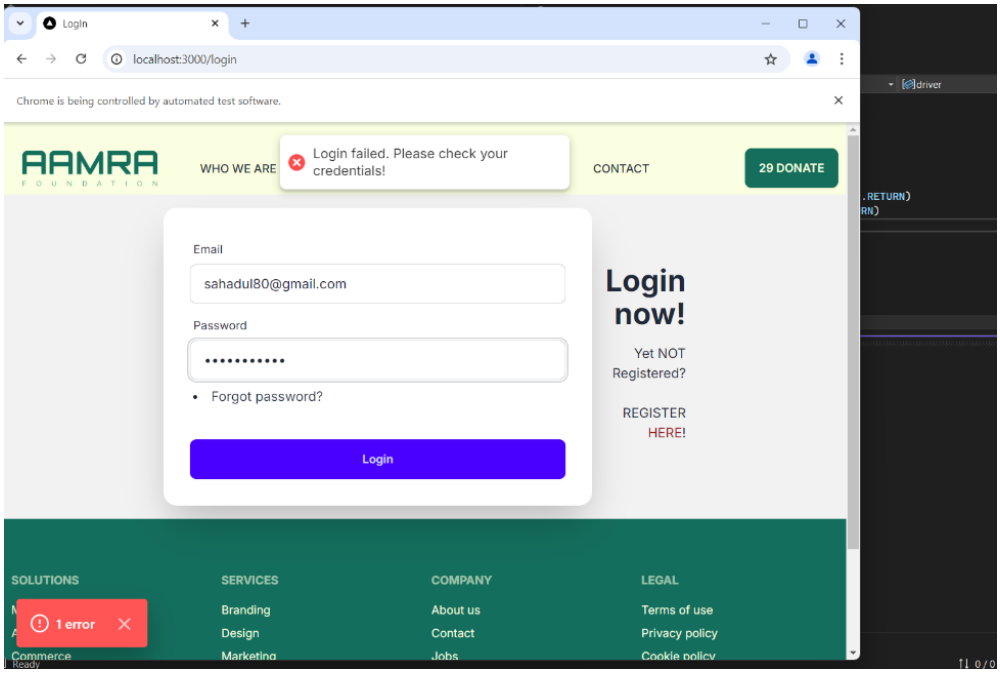

Figure19: output of the test

3.4. Forget Password

3.4.1. Forget Password: Pass

Passed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_6		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Medium		**Test Executed by:** Jahir Islam		
Module Name: Forget password		**Test Execution date:** 21/07/2024		
Test Title: recover the user profile				
Description: Request to login after forgetting the password				
Precondition: must be an existing user of the site				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Enter valid email address 3. Click submit	Email: sahadul80@gmail.com	An email is sent to reset the password.	Email is sent	pass
Post Condition: A reset code with a link will be sent the email to change the old password				

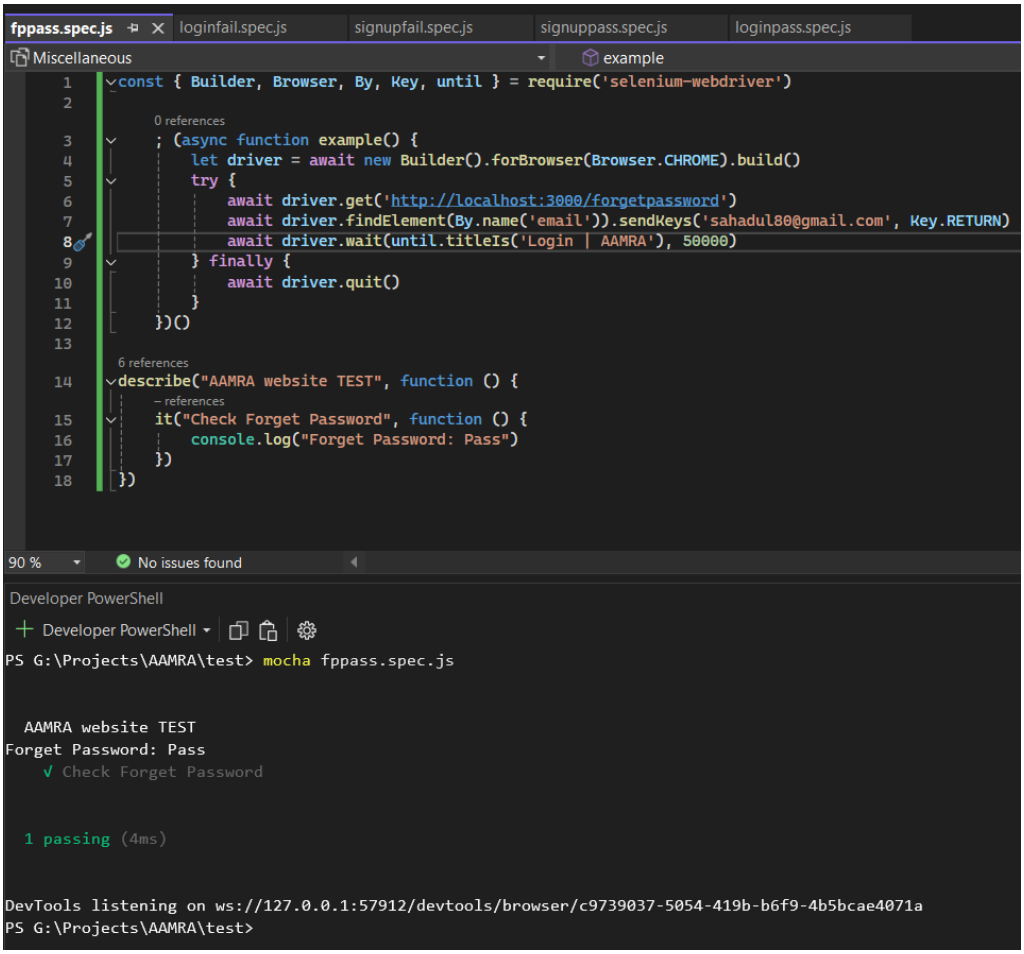

Figure20: code and result for forget password: Pass

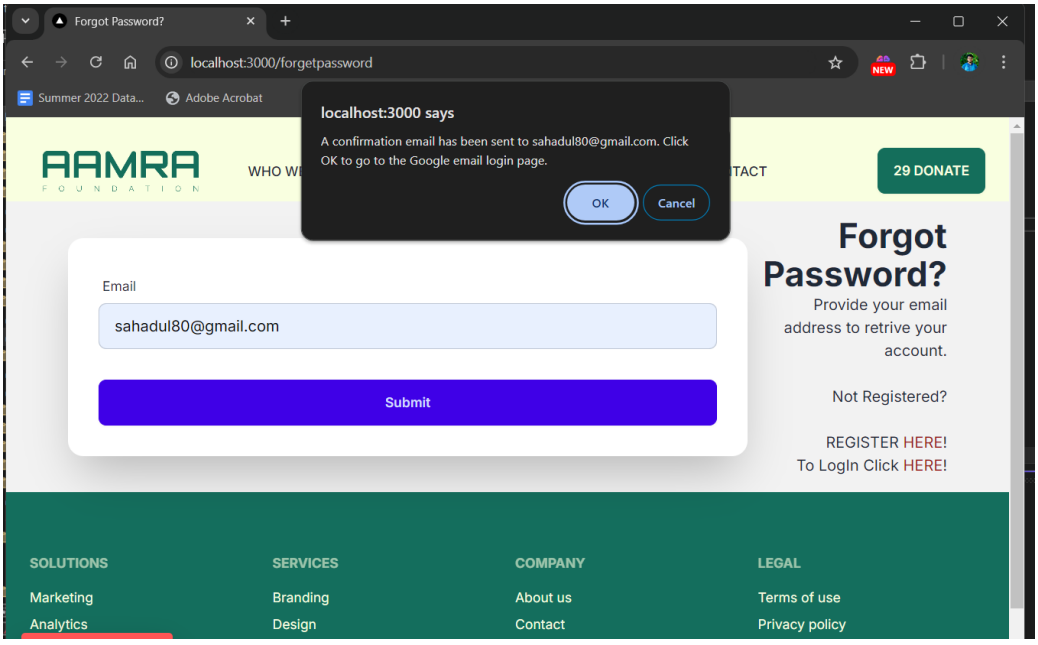

Figure21: output of the test

3.4.2. Forget Password: Fail

Failed test case

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_7		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Medium		**Test Executed by:** Jahir Islam		
Module Name: Forget password		**Test Execution date:** 21/07/2024		
Test Title: recover the user profile				
Description: Request to login after forgetting the password				
Precondition: must be an existing user of the site				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Enter valid email address 3. Click submit	Email: sahadul@gmail.com	Email is not found at the server	Email sending failure	fail
Post Condition: Asking to enter email valid email address				

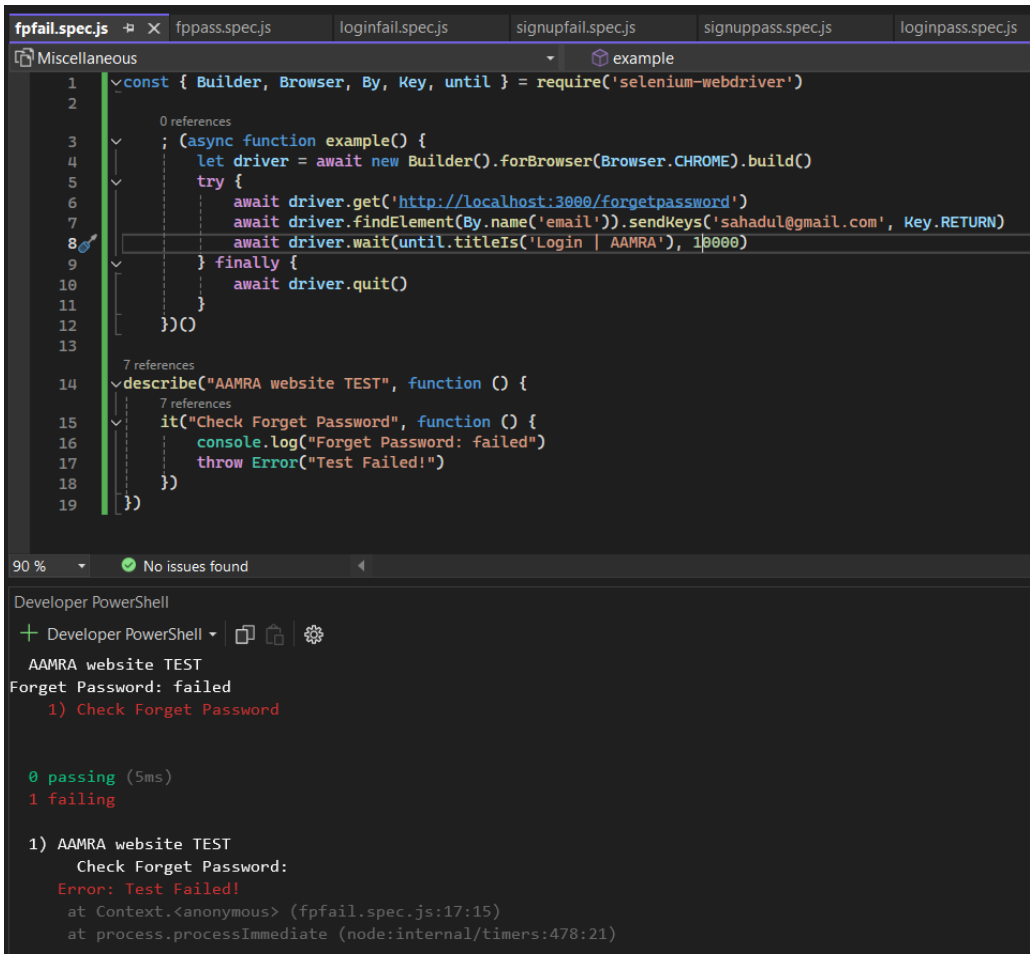

Figure22: code and result for forget password: Failed

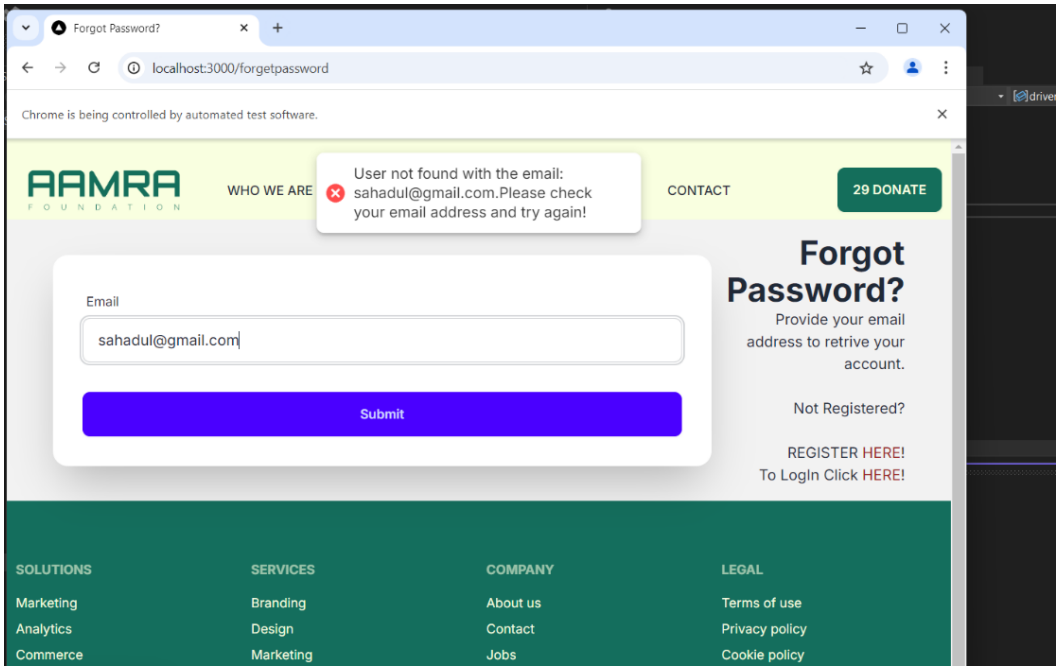

Figure23: output of the test

3.5. Subscribe

3.5.1. Subscribe: Pass

Passed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_8		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Low		**Test Executed by:** Md Sahadul Haque		
Module Name: Subscribe		**Test Execution date:** 21/07/2024		
Test Title: subscribe to the site				
Description: user should get push updates from the site				
Precondition: visit the site but not a user				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Enter valid email address 3. Click submit	Email: sahadul80@gmail.com	An email is sent to confirm subscription	Email is sent	pass
Post Condition: A welcome message will be sent the email to change the old password				

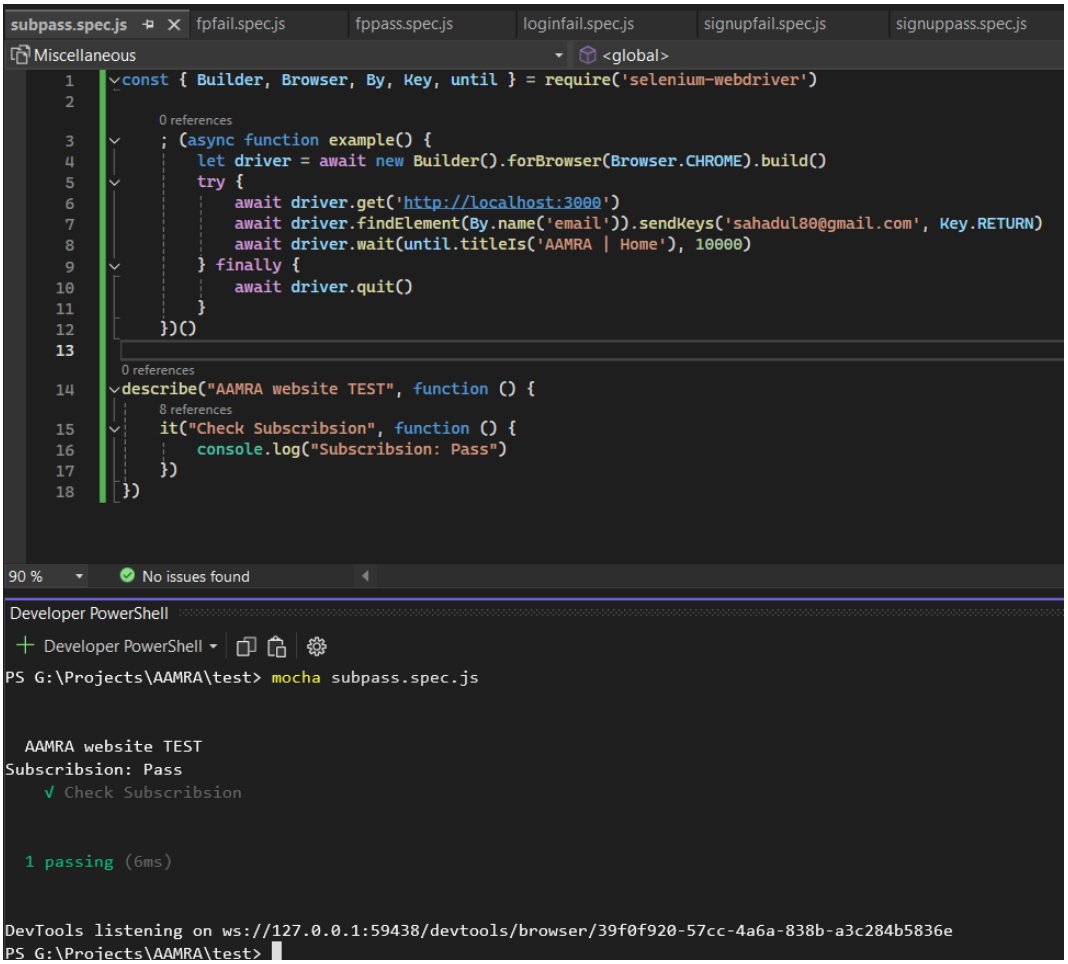

Figure24: code and result for subscribe: Passed

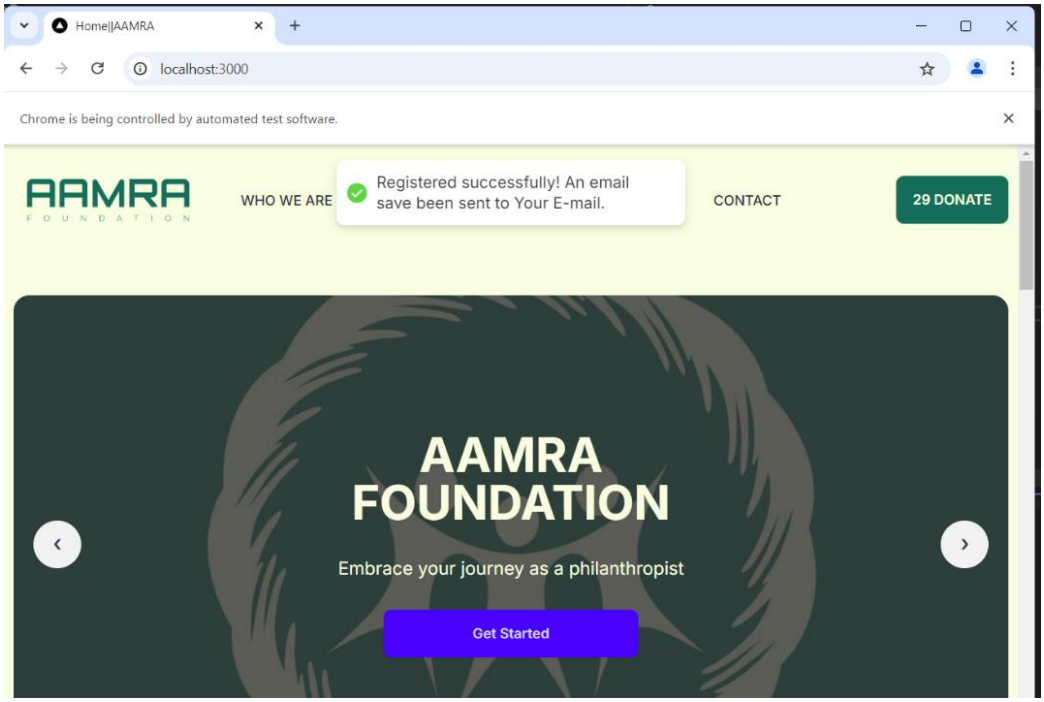

Figure25: output of the test

3.5.2. Subscribe: Fail

Failed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_9		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Low		**Test Executed by:** Md Sahadul Haque		
Module Name: Subscribe		**Test Execution date:** 21/07/2024		
Test Title: subscribe to the site				
Description: user should get push updates from the site				
Precondition: visit the site but not a user				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Go to the site 2. Enter valid email address 3. Click submit	Email: sahadul80@gmail.com	An email is sent to confirm subscription already exists.	Email sending failure	fail
Post Condition: stays at homepage with warning				

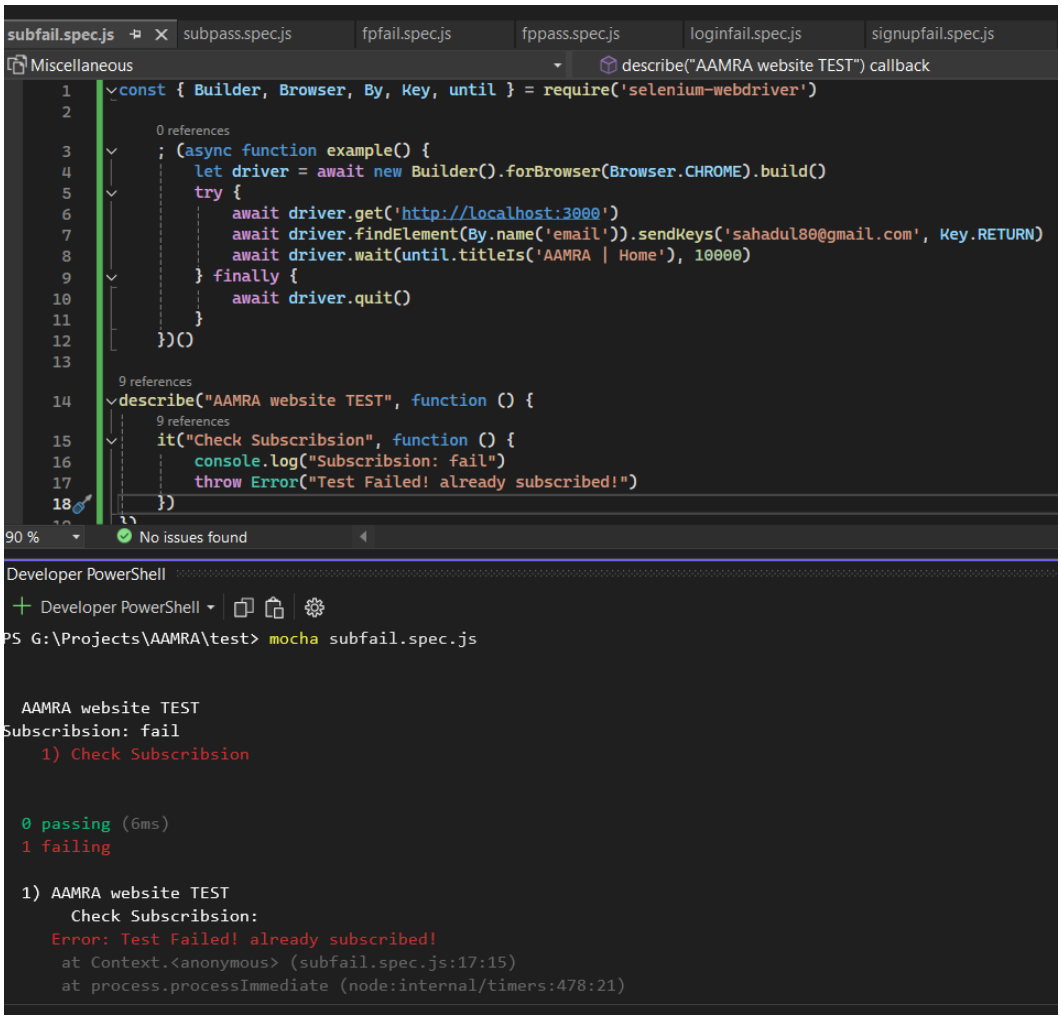

Figure26: code and result for subscribe Failed

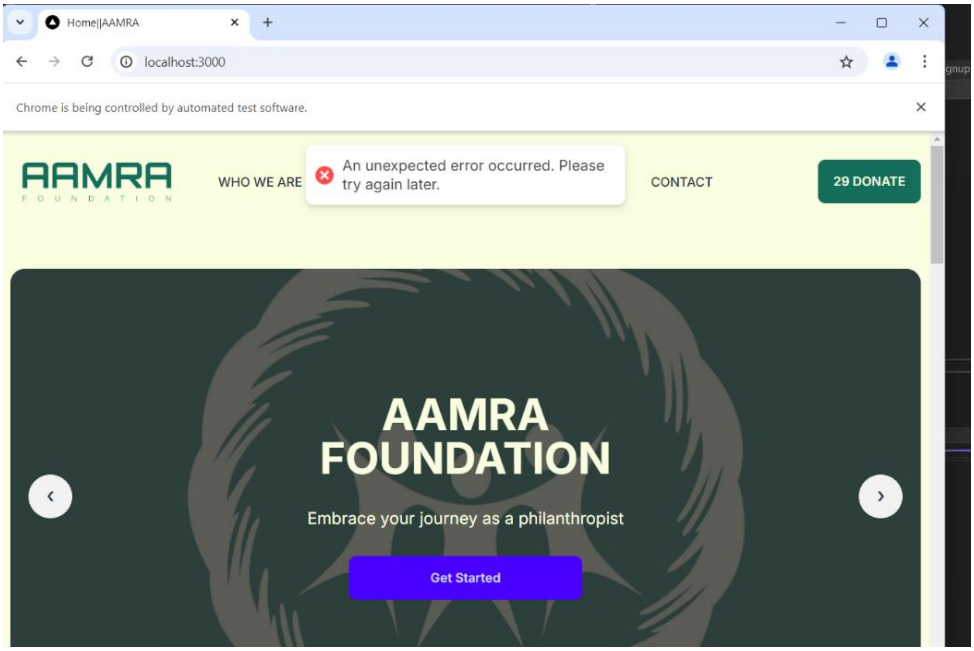

Figure27: output of the test

3.6. Profile

3.6.1. Profile: Pass

Passed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_10		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Low		**Test Executed by:** Md Sahadul Haque		
Module Name: Profile		**Test Execution date:** 21/07/2024		
Test Title: view profile after login				
Description: users should see their information				
Precondition: login to the site				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Login to the site 2. Click menu (icon) 3. Click profile	Email: sahadul80@gmail.com Password: Password123 Click the profile button	User can see the profile page	Profile is shown	pass
Post Condition: A reset code with a link will be sent the email to change the old password				

profilepass.spec.js | subfail.spec.js | subpass.spec.js | fpfail.spec.js | fppass.spec.js | loginfail.sp

Miscellaneous — describe("AAMRA website TEST") callback

```
const { Builder, Browser, By, Key, until } = require('selenium-webdriver')

0 references
; (async function example() {
    let driver = await new Builder().forBrowser(Browser.CHROME).build()
    try {
        await driver.get('http://localhost:3000/login')
        await driver.findElement(By.name('email')).sendKeys('sahadul80@gmail.com', Key.RETURN)
        await driver.findElement(By.name('password')).sendKeys('Password123', Key.RETURN)
        await driver.get('http://localhost:3000/profile')
        await driver.wait(until.titleIs('Profile | AAMRA'), 10000)
    } finally {
        await driver.quit()
    }
})()

0 references
describe("AAMRA website TEST", function () {
    10 references
    it("View Profile", function () {
        console.log("Profile Test: Pass")
    })
})
```

82 % — No issues found

Developer PowerShell

Developer PowerShell

```
PS G:\Projects\AAMRA\test> mocha profilepass.spec.js


  AAMRA website TEST
Profile Test: Pass
    √ View Profile


  1 passing (5ms)


DevTools listening on ws://127.0.0.1:60562/devtools/browser/aec15cbc-e9b8-471f-9201-566a78c41f15
PS G:\Projects\AAMRA\test>
```

Figure28:

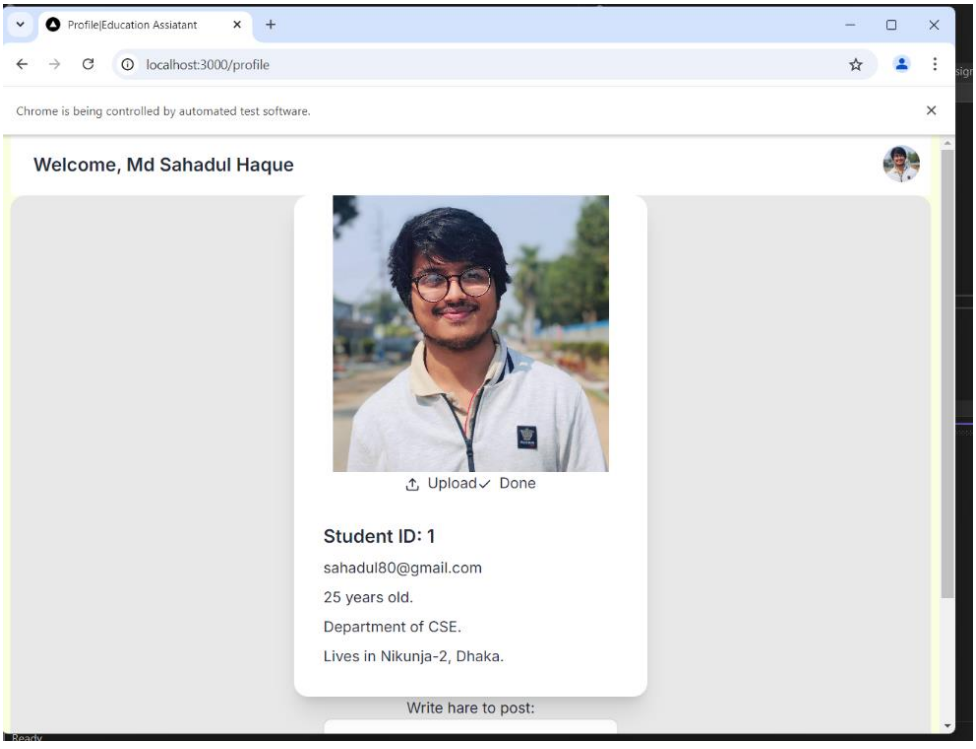

Figure29:

3.6.2. Profile: Fail

Failed test case:

Project Name: AAMRA Foundation		**Test Designed by:** Md Sahadul Haque		
Test Case: FR_10		**Test Design date:** 01/07/2024		
Test Priority (Low, Medium, High): Low		**Test Executed by:** Md Sahadul Haque		
Module Name: Profile		**Test Execution date:** 21/07/2024		
Test Title: view profile after login				
Description: users should see their information				
Precondition: login to the site				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Login to the site 2. Click menu (icon) 3. Click profile	Email: sahadul80@gmail.com Password: Password123 Click the profile button	User can see the profile page	Profile is shown	pass
Post Condition: A reset code with a link will be sent the email to change the old password				

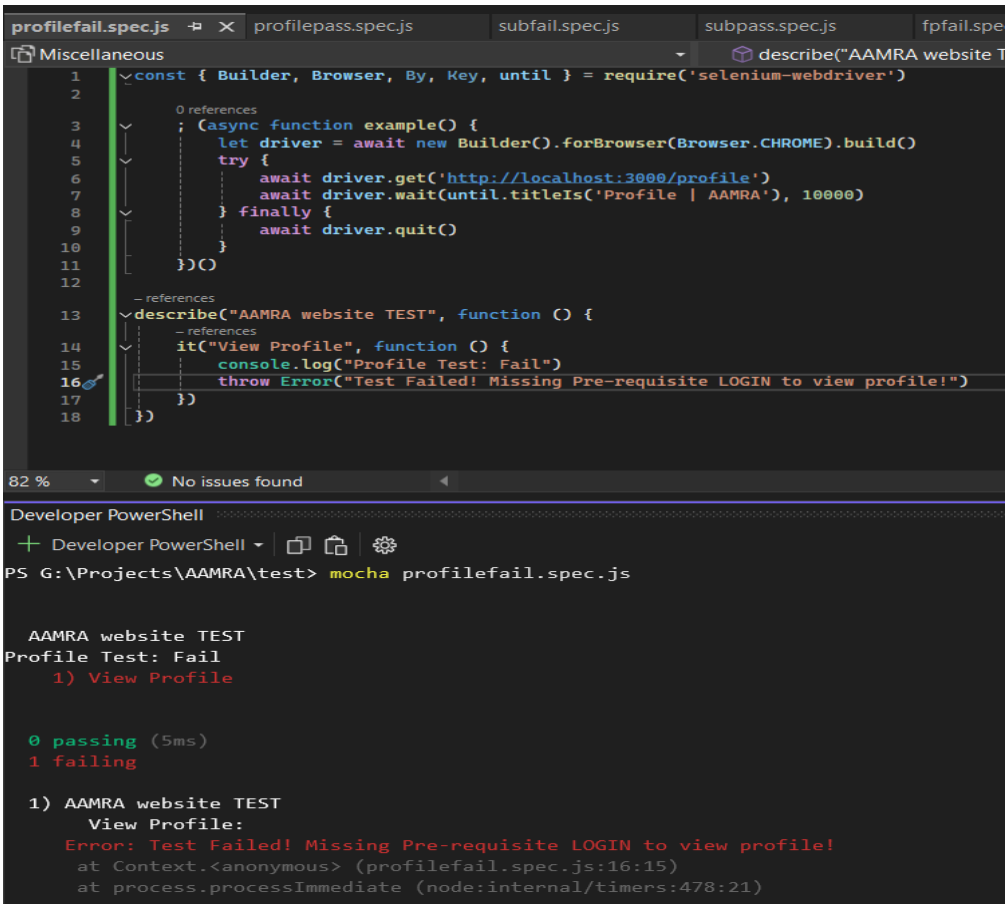

Figure30:

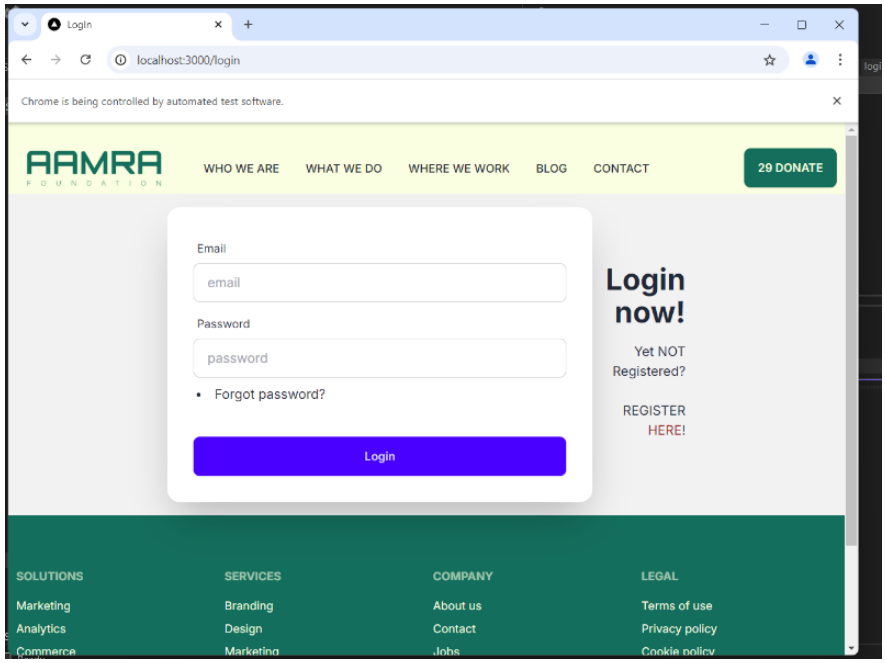

Figure31:

3.7. Logout

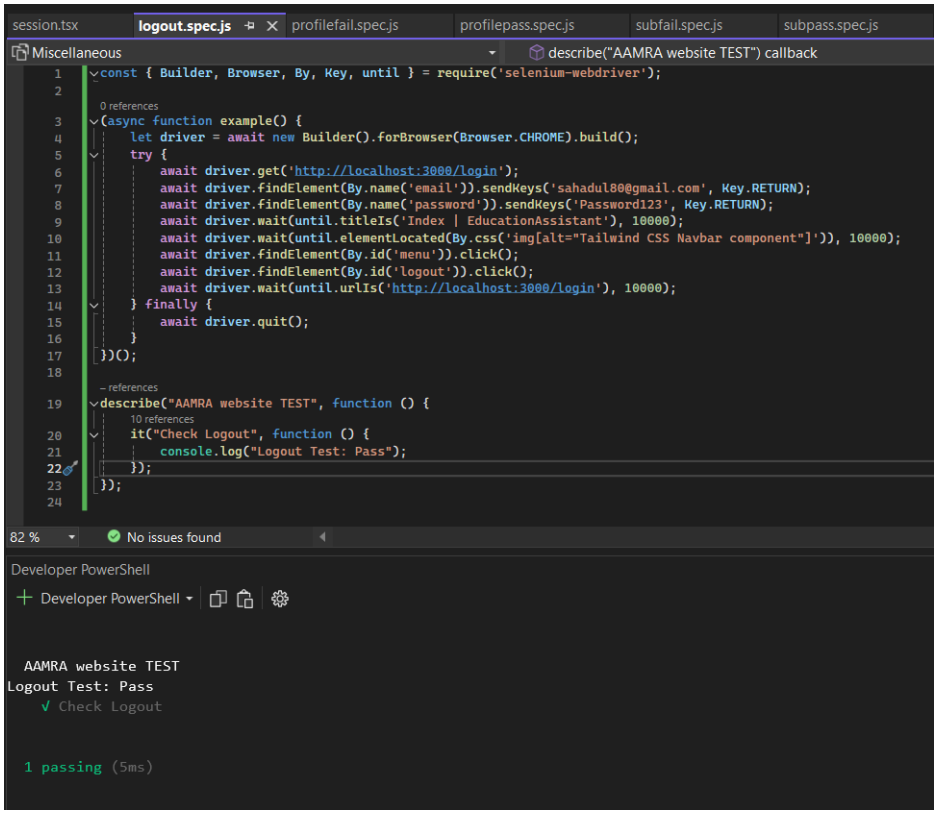

Figure32:

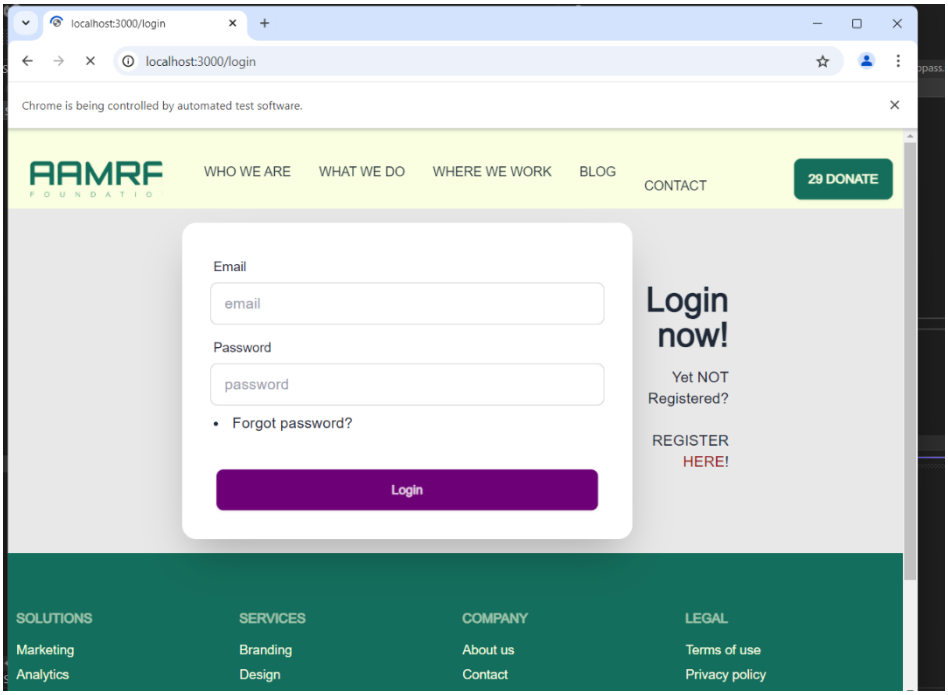

Figure33:

4. Acknowledgement

This is to verify that the tool selenium accompanied with JavaScript framework mocha are only used for testing a portion of the project “AAMRA” which is not deployed yet and still in its developing stages. These testing tools can be used to perform Distributed Denial of Service (DDos) attacks my modifying the codes in any website which is not using DDos prevention systems i.e., Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA). Performing DDoS attacks is illegal and unethical, and it can cause significant harm to websites, servers, and other internet infrastructure. Engaging in such activities can lead to severe legal consequences, including criminal charges. Using these tools for malicious activities violates their intended purpose and is against the law.