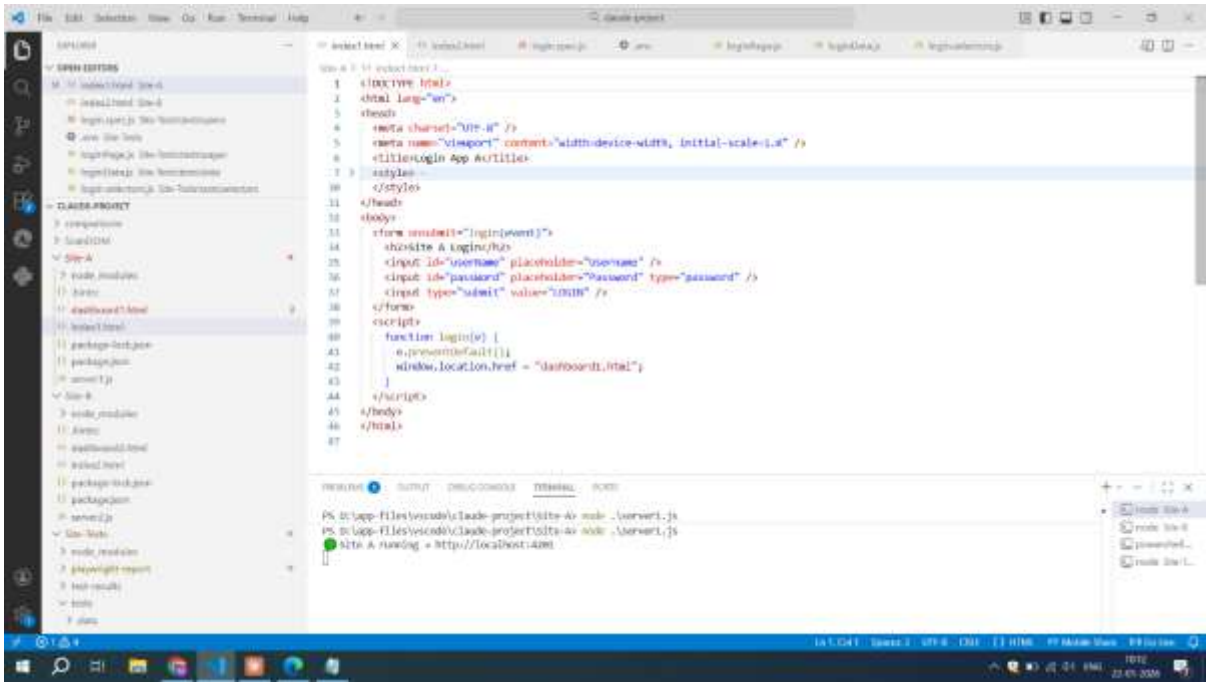
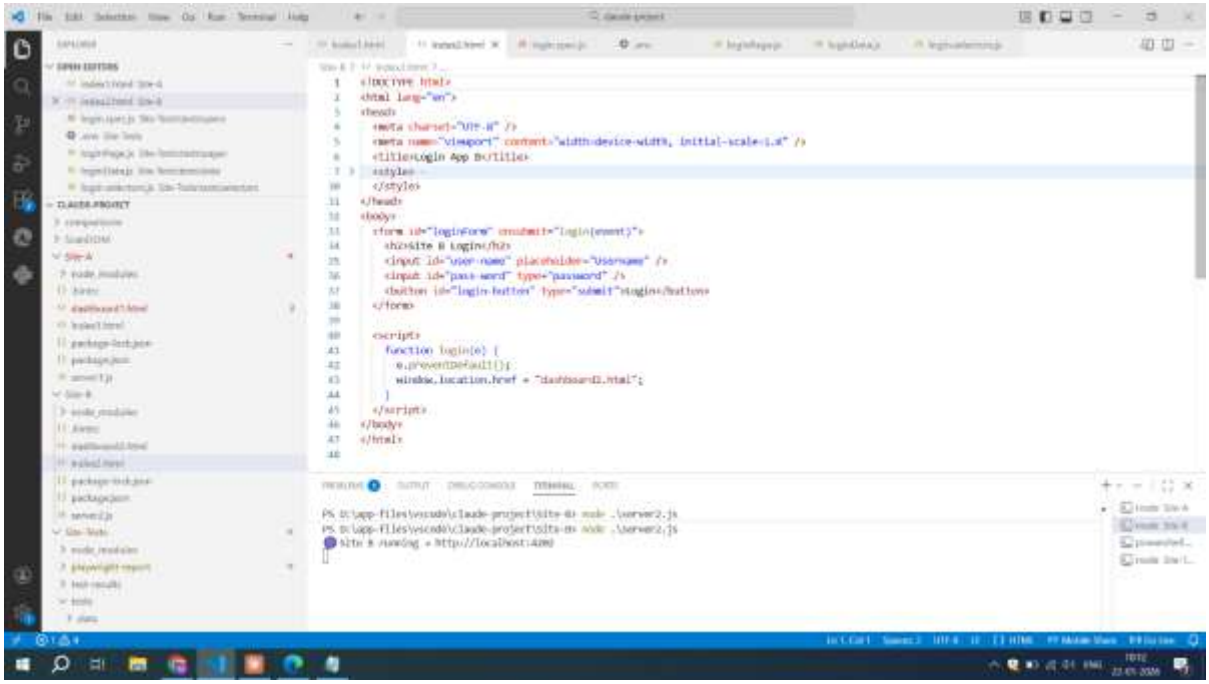
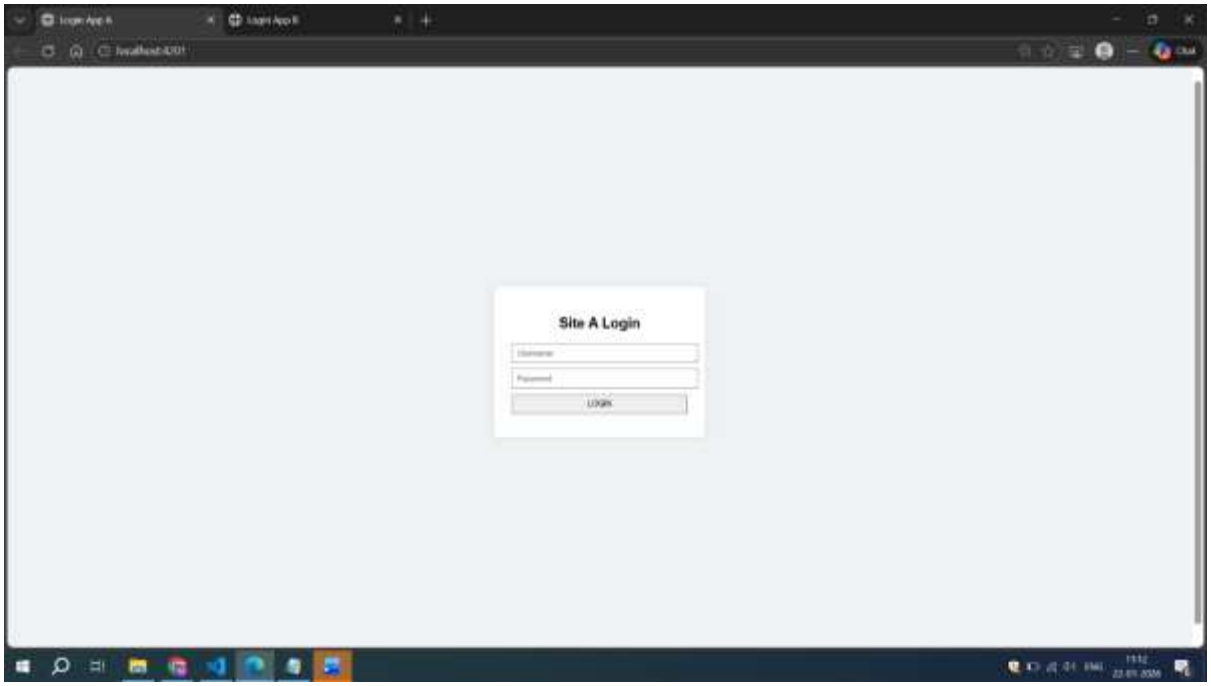
Documentation of the Project

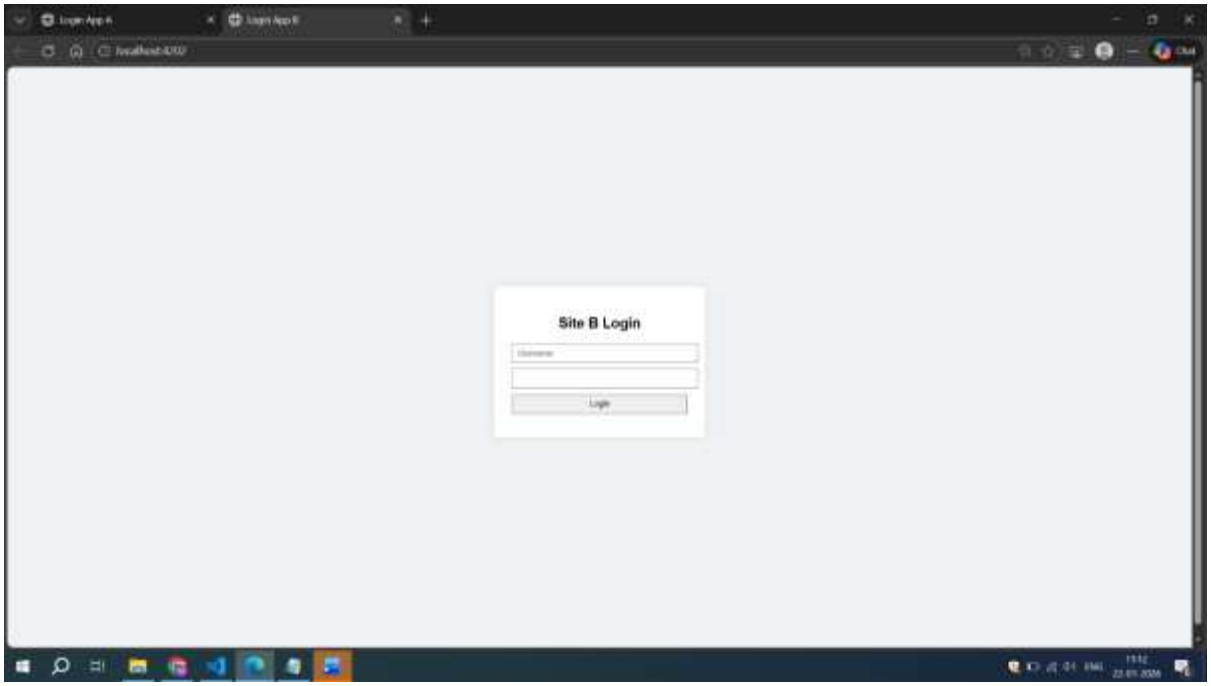
Agentic UI testing

1 the websites – Baseline (SiteA) and Variant (SiteB)

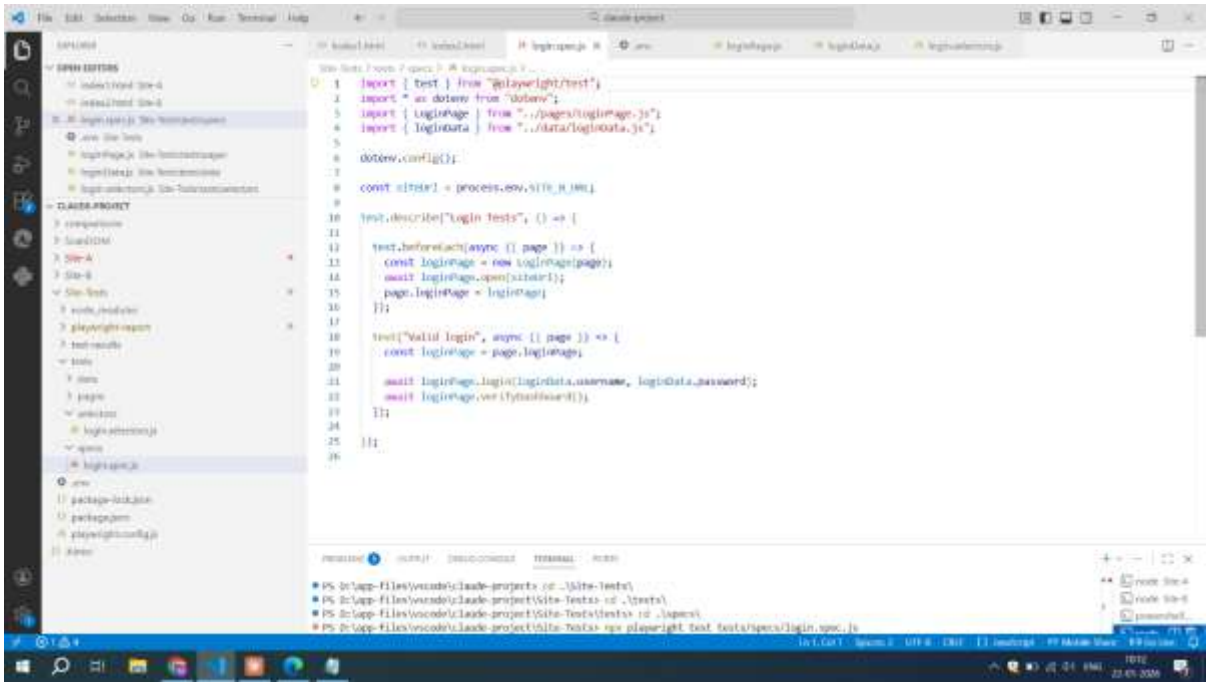




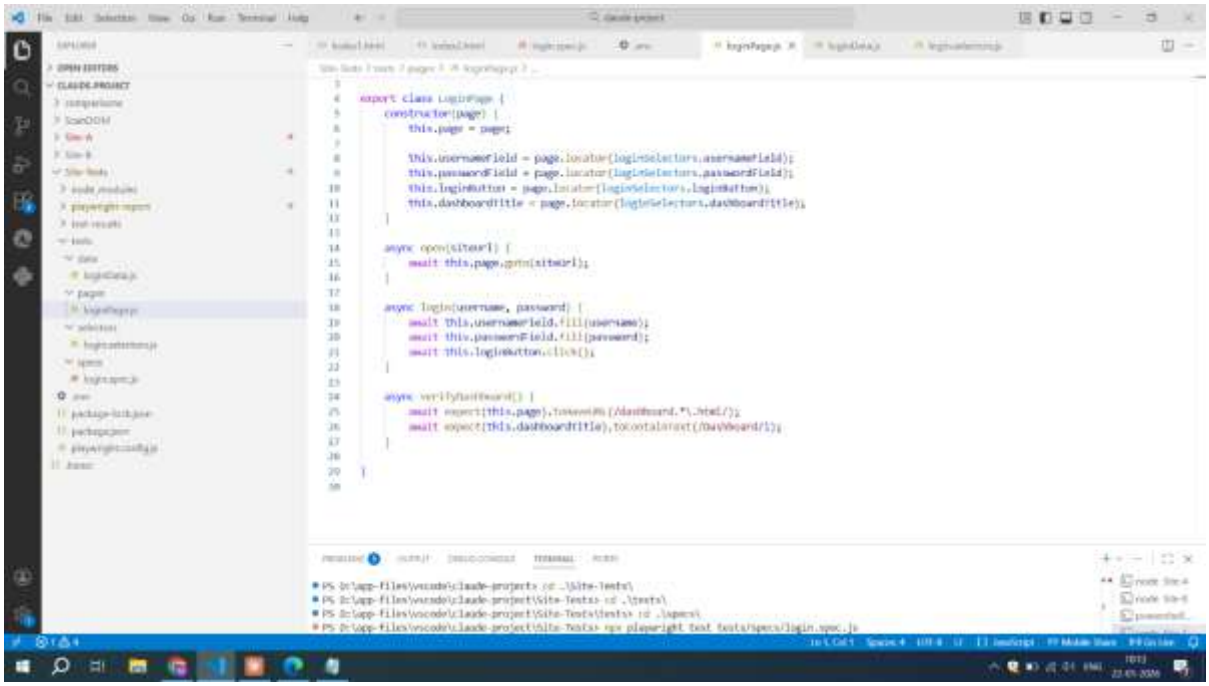

Site A Login
LOGIN


Site B Login
Login

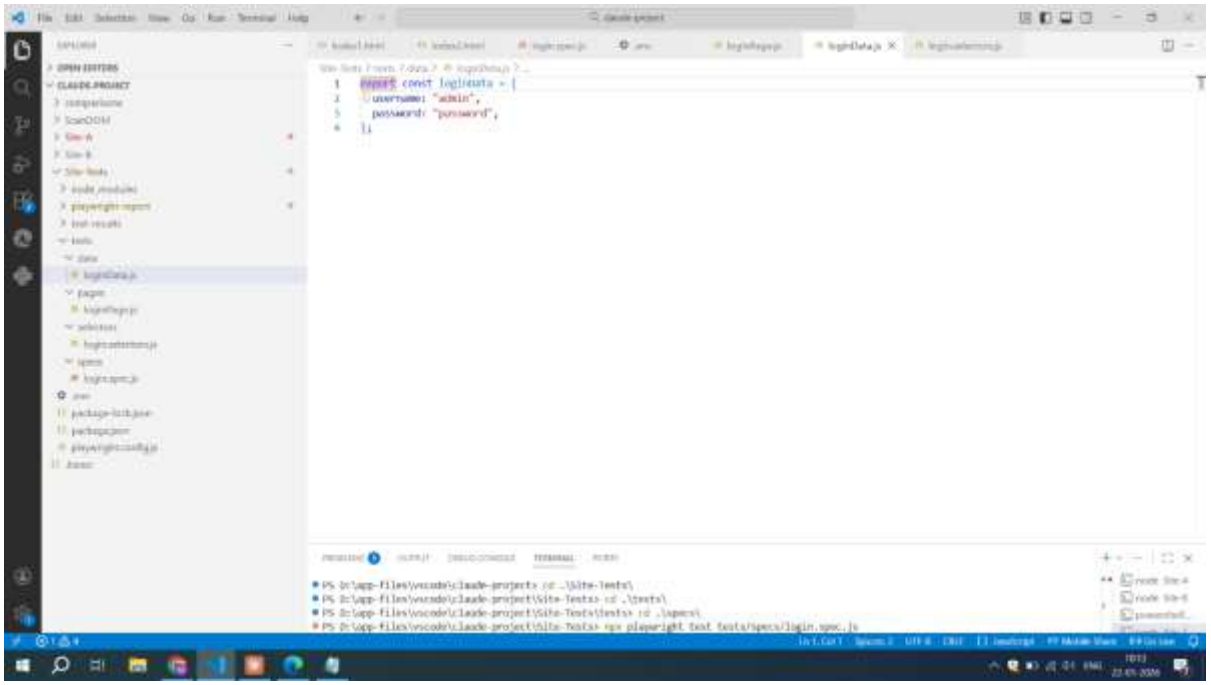
2 test files

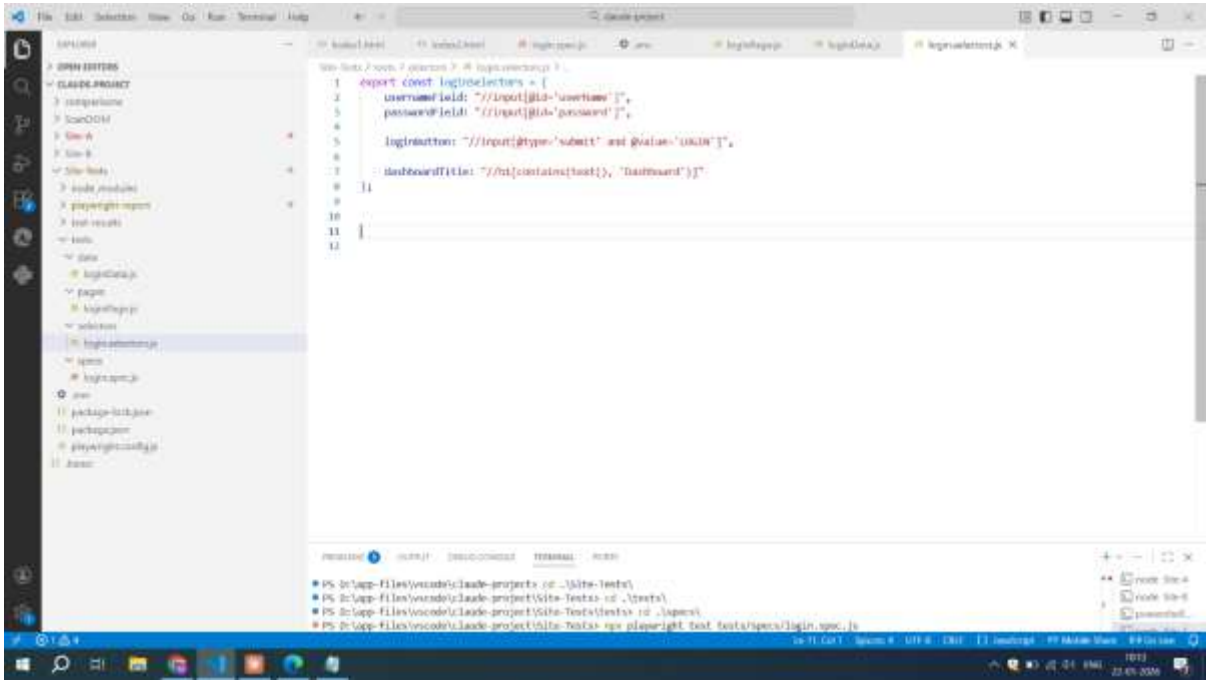


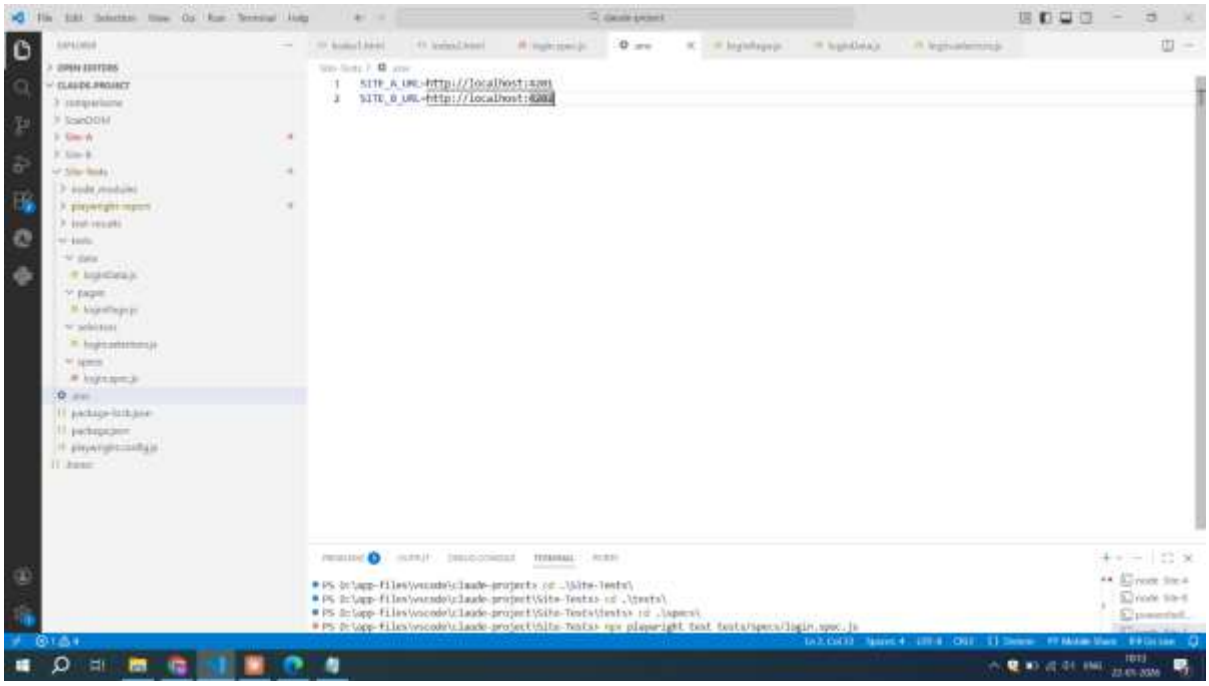
```
import { test } from "@playwright/test";
import * as dotenv from "dotenv";
import { LoginPage } from "../pages/loginPage.js";
import { loginData } from "../data/loginData.js";

dotenv.config();

const siteUrl = process.env.SITE_B_URL;

test.describe("Login tests", () => {

  test.beforeEach(async ({ page }) => {
    const loginPage = new LoginPage(page);
    await loginPage.open(siteUrl);
    page.loginPage = loginPage;
  });

  test("Valid login", async ({ page }) => {
    const loginPage = page.loginPage;

    await loginPage.login(loginData.username, loginData.password);
    await loginPage.verifyDashboard();
  });

});
```



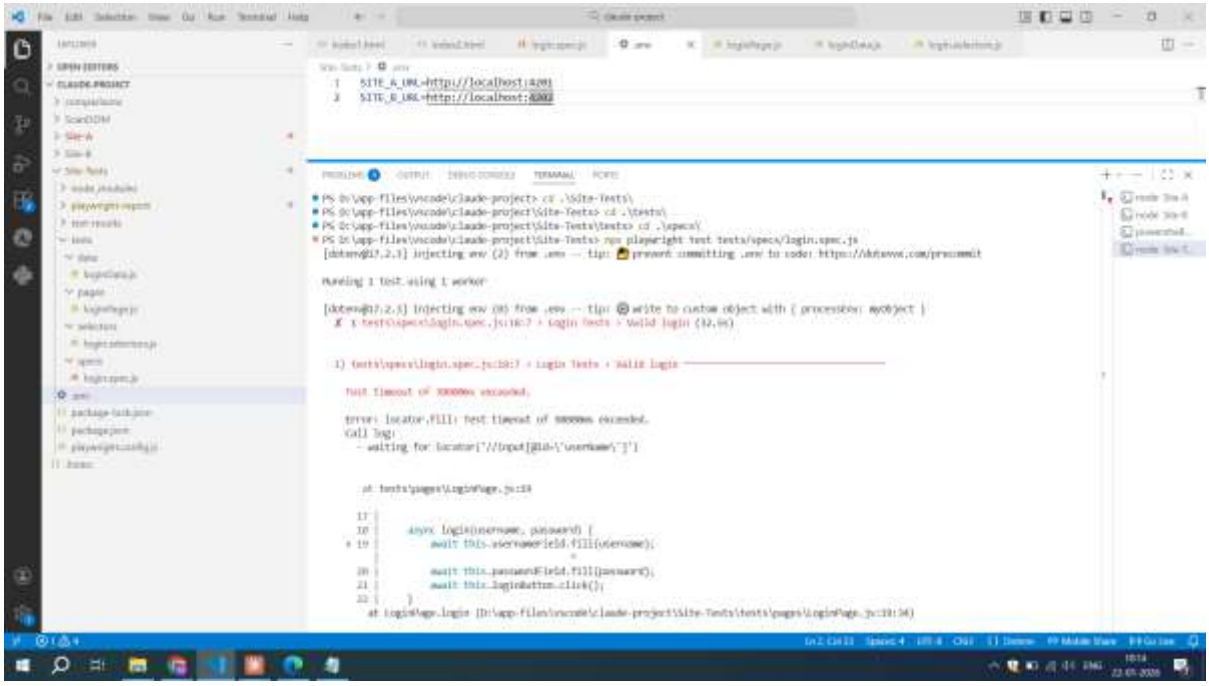
```
export class LoginPage {
    constructor(page) {
        this.page = page;

        this.usernameField = page.locator(loginSelectors.usernameField);
        this.passwordField = page.locator(loginSelectors.passwordField);
        this.loginButton = page.locator(loginSelectors.loginButton);
        this.dashboardTitle = page.locator(loginSelectors.dashboardTitle);
    }

    async open(siteUrl) {
        await this.page.goto(siteUrl);
    }

    async login(username, password) {
        await this.usernameField.fill(username);
        await this.passwordField.fill(password);
        await this.loginButton.click();
    }

    async verifyDashboard() {
        await expect(this.page).toHaveURL(/dashboard.*\.html/);
        await expect(this.dashboardTitle).toContainText(/Dashboard/i);
    }

}
```


export const loginData = {
username: "admin",
password: "password",
};

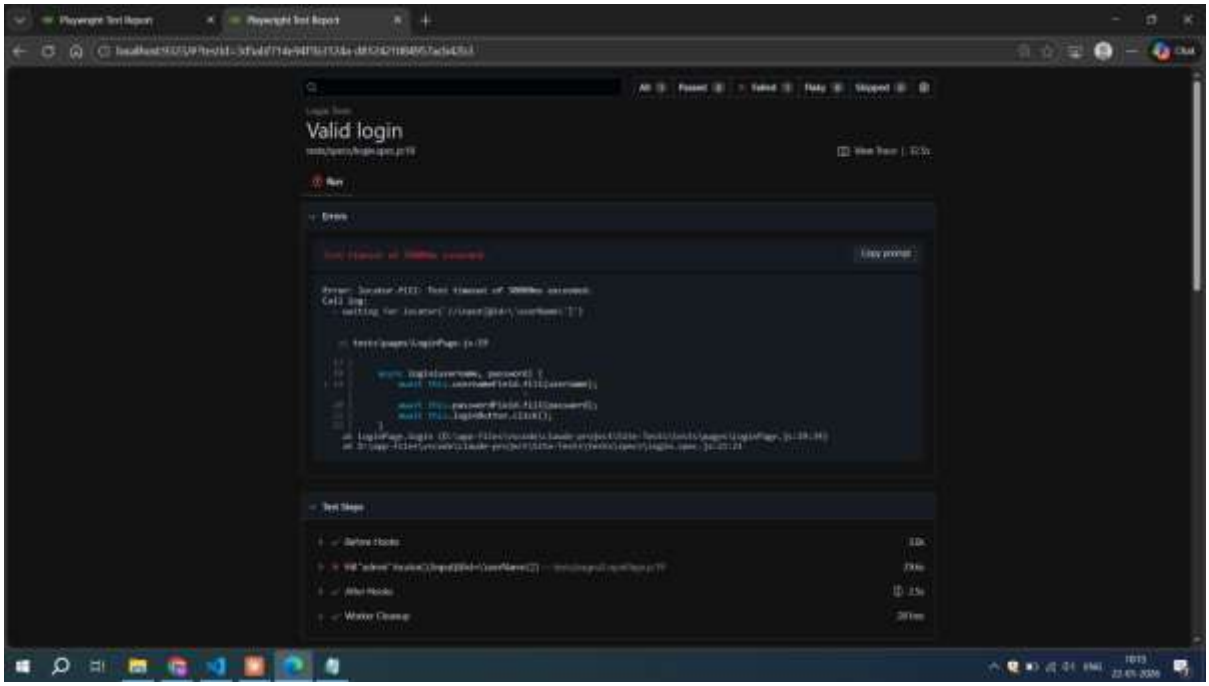

export const loginSelectors = {
usernameField: "//input[@id='username']",
passwordField: "//input[@id='password']",
loginButton: "//input[@type='submit' and @value='LOGIN']",
dashboardTitle: "//h1[contains(text(), 'Dashboard')]"
};



3) test result in terminal (npx playwright test)

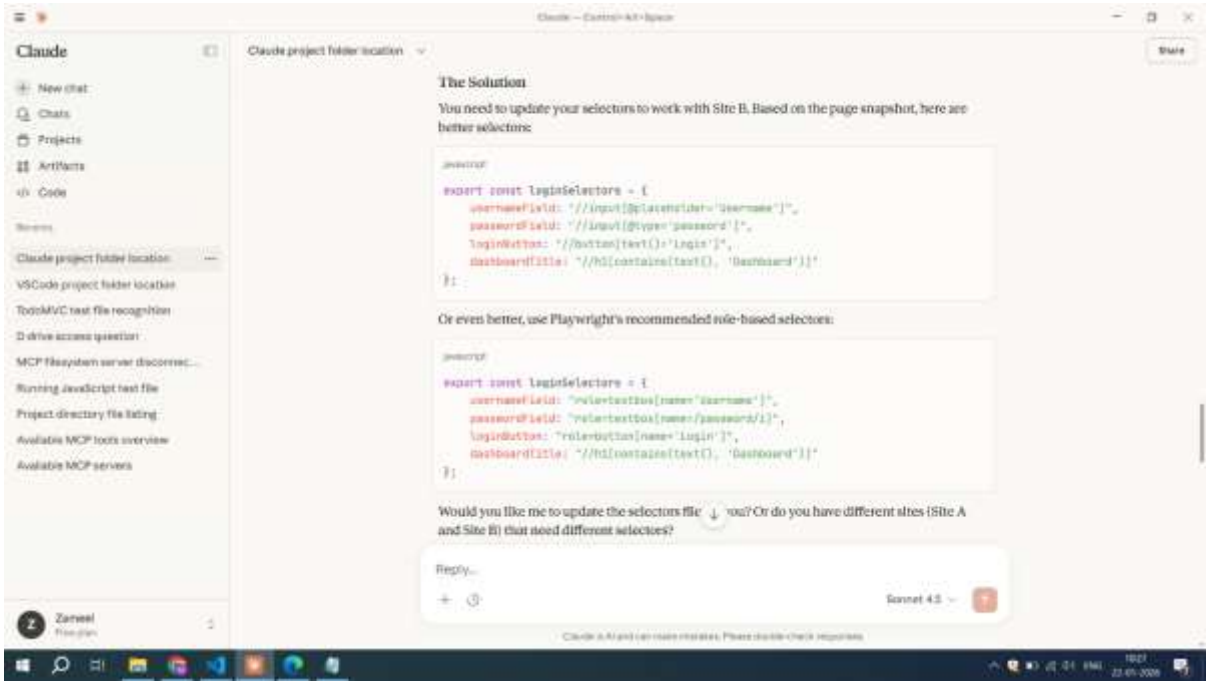


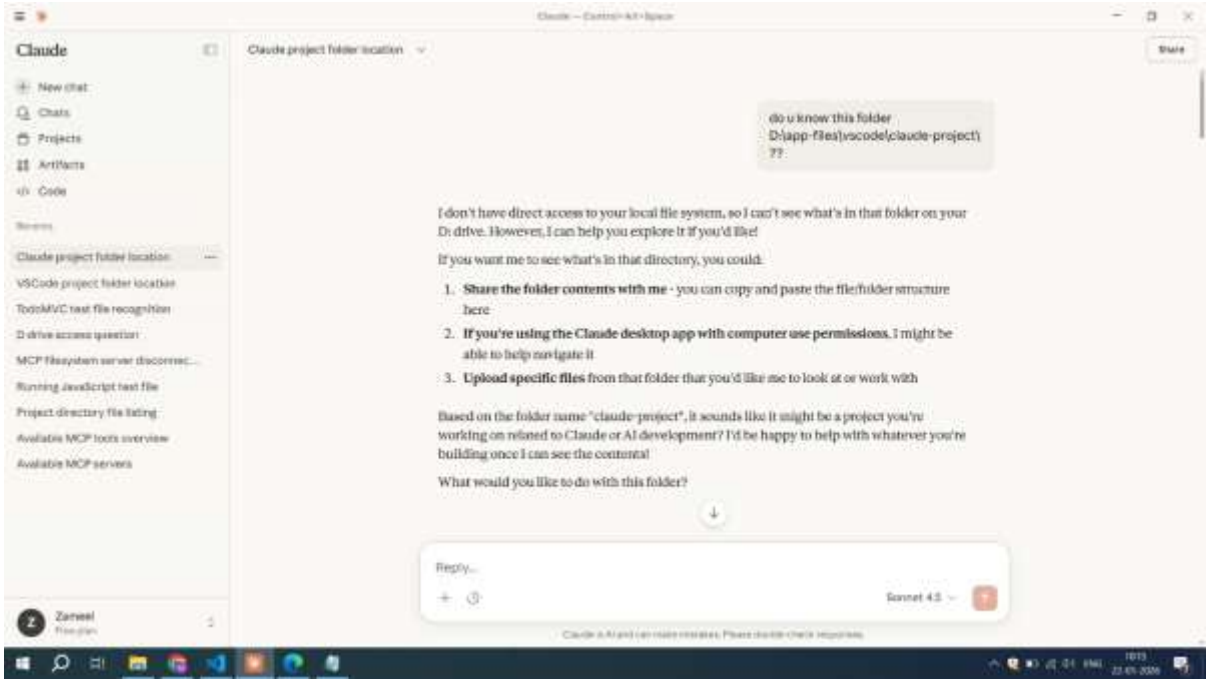
4) playwright report

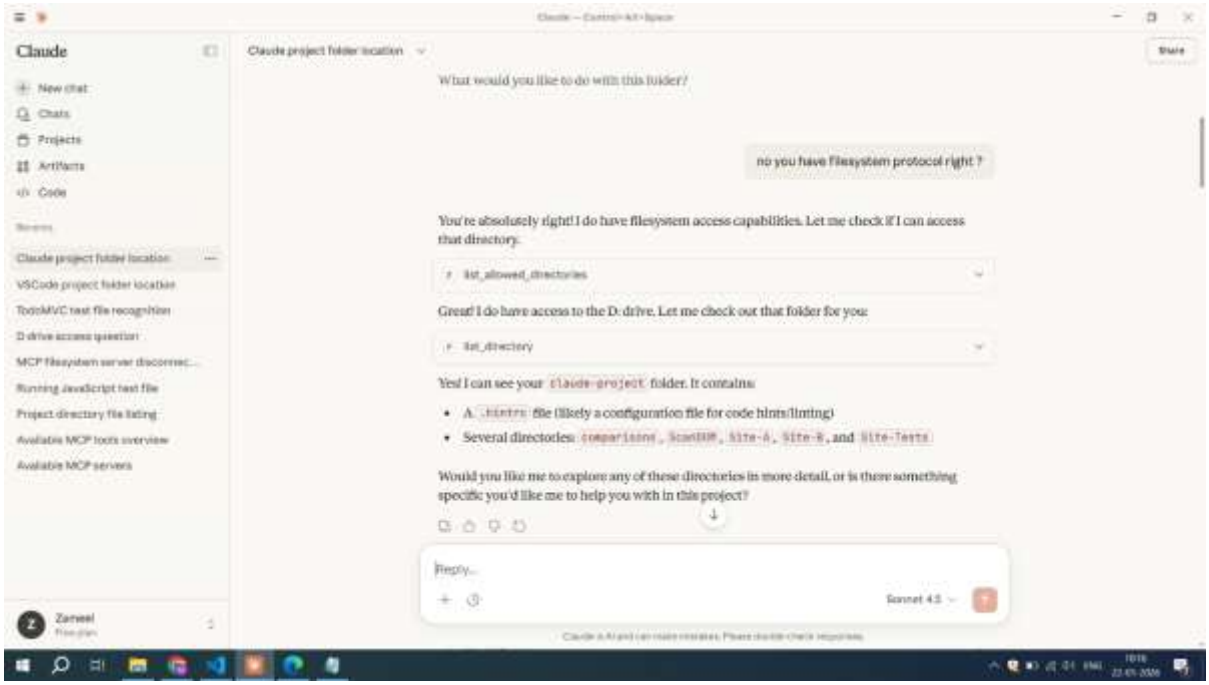


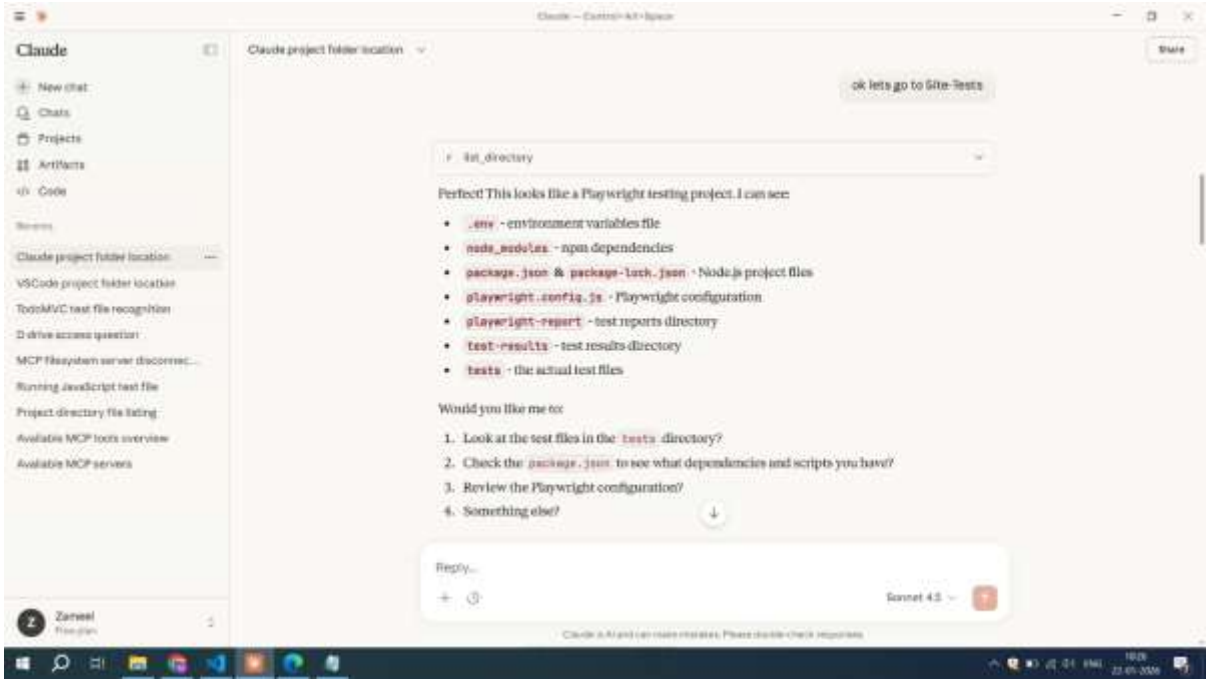
5)agentic ai interface(Claude)

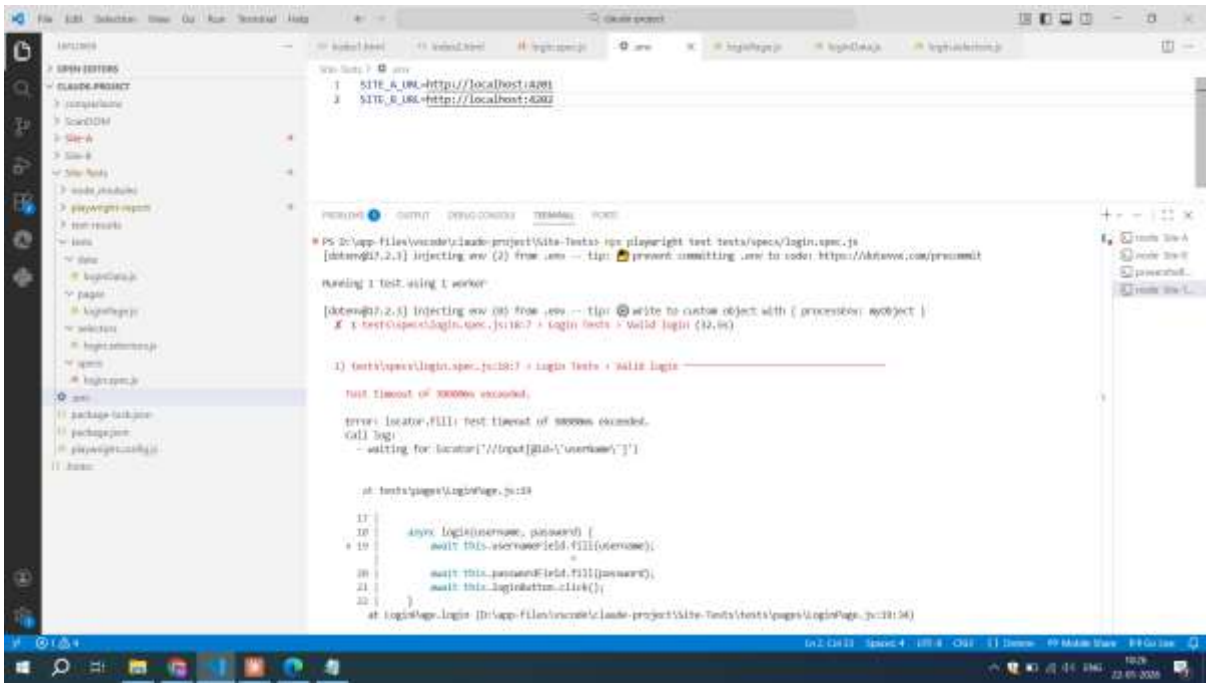
i)analysing the error

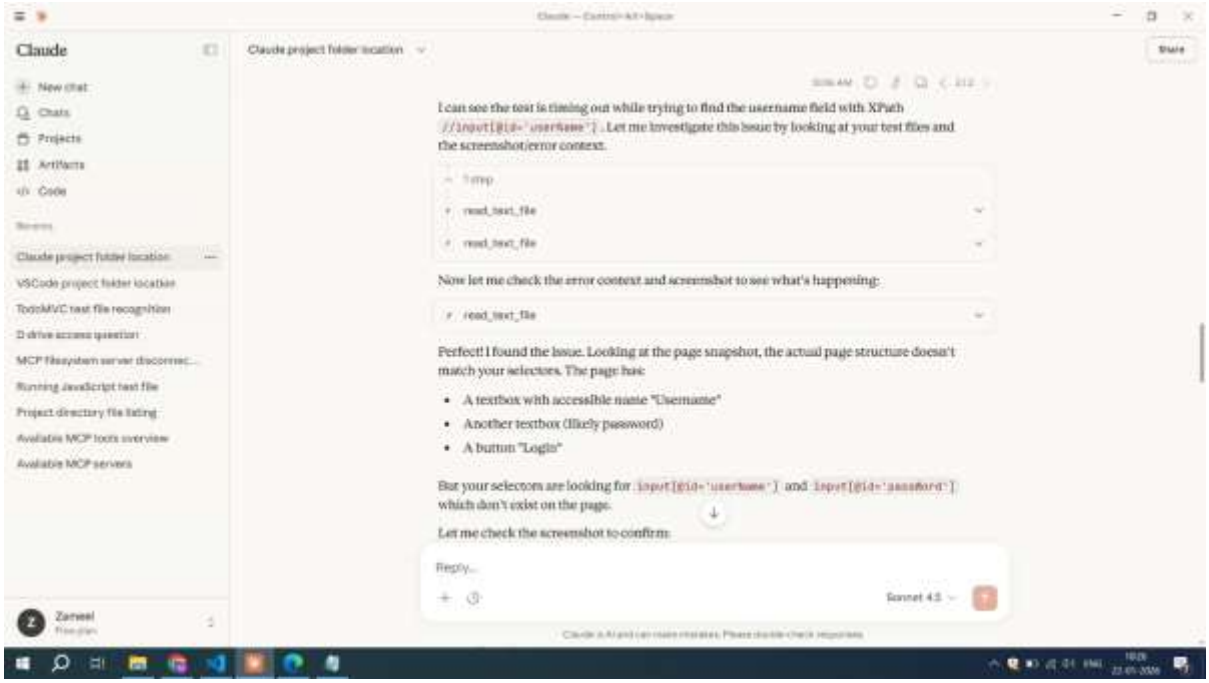


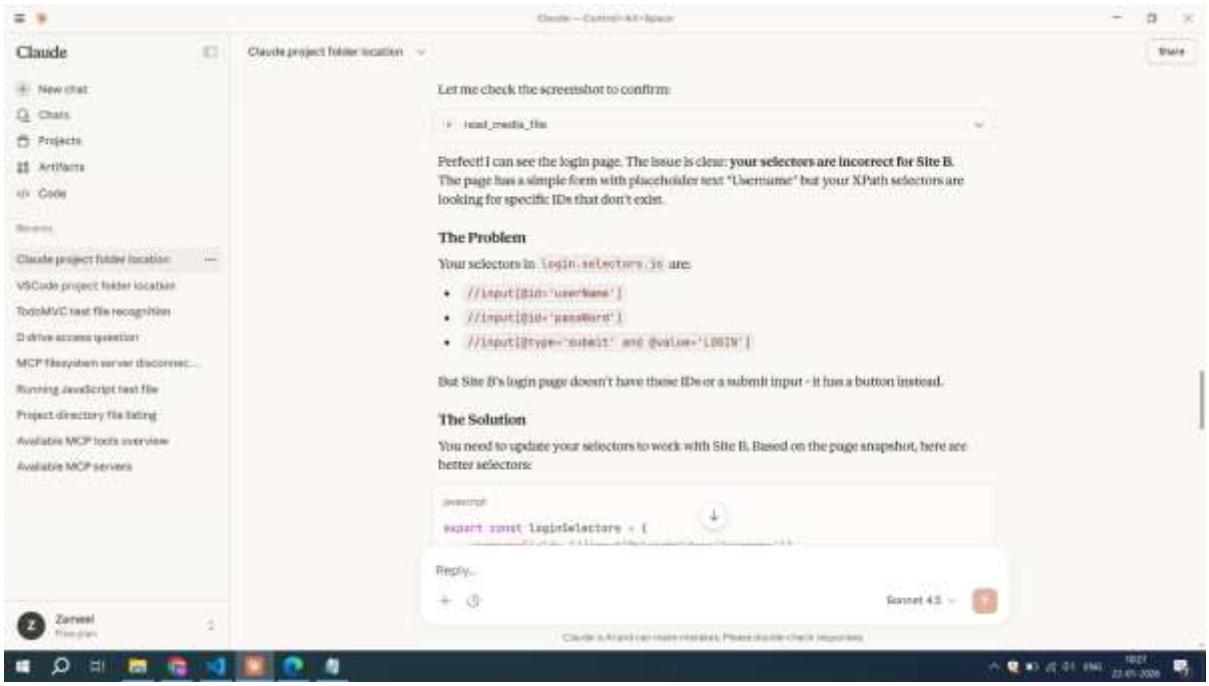

Claude
New chat
Chats
Projects
Artifacts
Code
Recents
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zaneel
Free plan
Claude project folder location
Share
do u know this folder D:\app-files\vscode\claude-project\ ??
I don't have direct access to your local file system, so I can't see what's in that folder on your D: drive. However, I can help you explore it if you'd like!
If you want me to see what's in that directory, you could:
1. Share the folder contents with me - you can copy and paste the file/folder structure here
2. If you're using the Claude desktop app with computer use permissions, I might be able to help navigate it
3. Upload specific files from that folder that you'd like me to look at or work with
Based on the folder name "claude-project", it sounds like it might be a project you're working on related to Claude or AI development? I'd be happy to help with whatever you're building once I can see the contents!
What would you like to do with this folder?
Reply...
Sonnet 4.5
Claude is AI and can make mistakes. Please double-check responses.


Claude
New chat
Chats
Projects
Artifacts
Code
Recents
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zaneel
Free plan
Claude project folder location
Share
What would you like to do with this folder?
no you have filesystem protocol right ?
You're absolutely right! I do have filesystem access capabilities. Let me check if I can access that directory.
list_allowed_directories
Great! I do have access to the D: drive. Let me check out that folder for you:
list_directory
Yes! I can see your claude-project folder. It contains:
A .claude file (likely a configuration file for code hints/linting)
Several directories: comparisons, ScanSSM, Site-A, Site-B, and Site-Tests
Would you like me to explore any of these directories in more detail, or is there something specific you'd like me to help you with in this project?
Reply...
Sonnet 4.5
Claude is AI and can make mistakes. Please double-check responses.

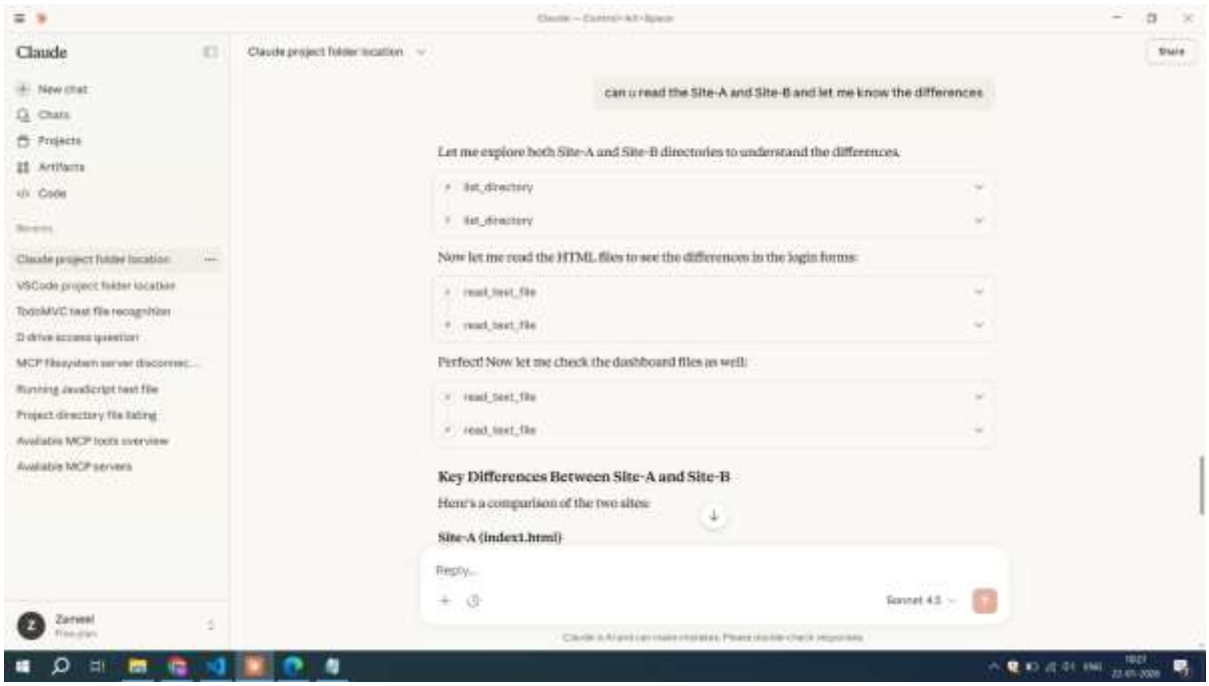

Claude
New chat
Chats
Projects
Artifacts
Code
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zarreel
Share
ok lets go to Site-Tests
list_directory
Perfect! This looks like a Playwright testing project. I can see:
.env - environment variables file
node_modules - npm dependencies
package.json & package-lock.json - Node.js project files
playwright.config.js - Playwright configuration
playwright-report - test reports directory
test-results - test results directory
tests - the actual test files
Would you like me to:
1. Look at the test files in the tests directory?
2. Check the package.json to see what dependencies and scripts you have?
3. Review the Playwright configuration?
4. Something else?
Reply...
Sonnet 4.5

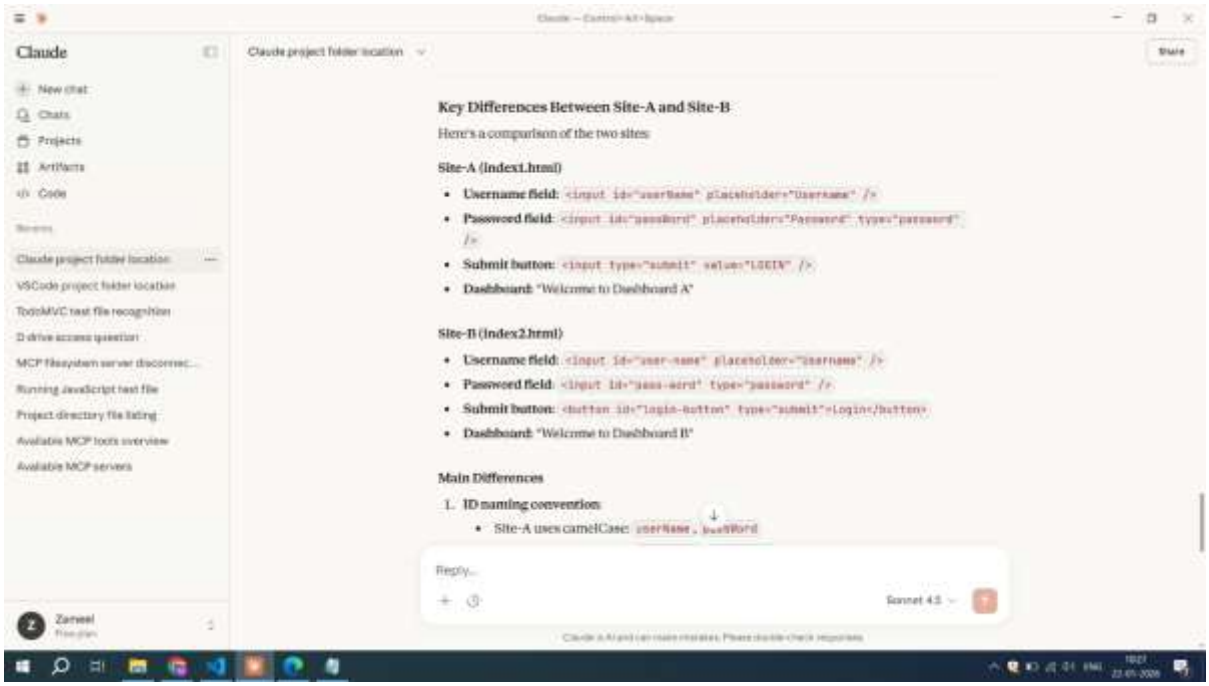

SITE_A_URL=http://localhost:4201
SITE_B_URL=http://localhost:4202
Running 1 test using 1 worker
1) tests\specs\login.spec.js:10:7 › Login Tests › Valid login
Test timeout of 30000ms exceeded.
Call log:
- waiting for locator('//input[@id='username']')
at tests\pages\LoginPage.js:19
async login(username, password) {
await this.usernameField.fill(username);
await this.passwordField.fill(password);
await this.loginButton.click();
}
at LoginPage.login (D:\app-files\vscode\claude-project\Site-Tests\tests\pages\LoginPage.js:19:34)

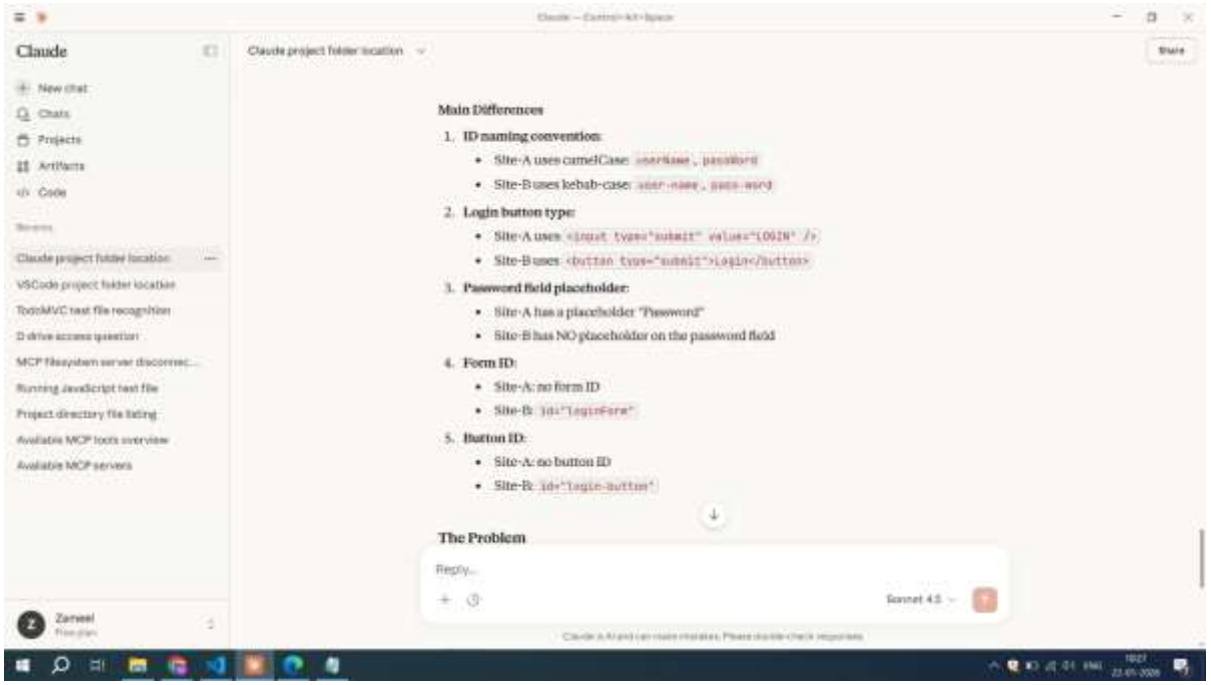

Claude
Claude project folder location
Share
New chat
Chats
Projects
Artifacts
Code
Recents
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zarreel
Free plan
I can see the test is timing out while trying to find the username field with XPath //input[@id='username']. Let me investigate this issue by looking at your test files and the screenshot/error context.
read_text_file
read_text_file
Now let me check the error context and screenshot to see what's happening:
read_text_file
Perfect! I found the issue. Looking at the page snapshot, the actual page structure doesn't match your selectors. The page has:
A textbox with accessible name "Username"
Another textbox (likely password)
A button "Login"
But your selectors are looking for input[@id='username'] and input[@id='password'] which don't exist on the page.
Let me check the screenshot to confirm:
Reply...
Sonnet 4.5

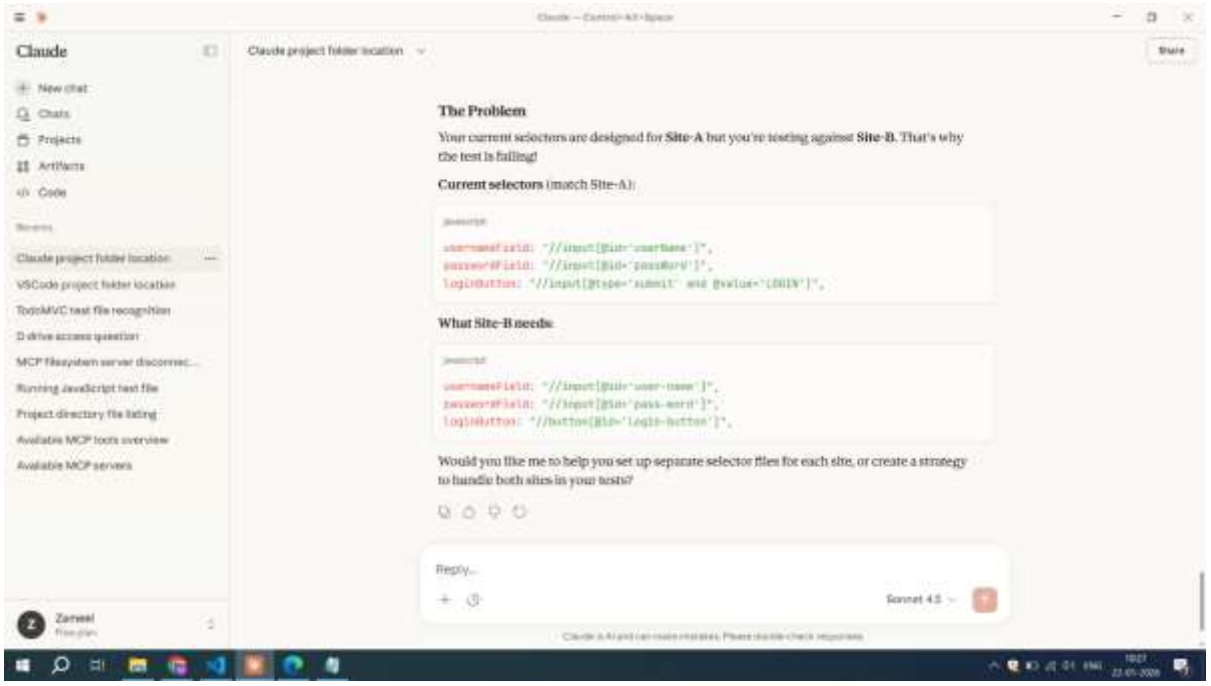

Claude
Claude project folder location
Share
New chat
Chats
Projects
Artifacts
Code
Recents
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zarreel
Free plan
Let me check the screenshot to confirm:
read_media_file
Perfect! I can see the login page. The issue is clear: your selectors are incorrect for Site B. The page has a simple form with placeholder text "Username" but your XPath selectors are looking for specific IDs that don't exist.
The Problem
Your selectors in login.selectors.js are:
//input[@id='username']
//input[@id='password']
//input[@type='submit' and @value='LOGIN']
But Site B's login page doesn't have those IDs or a submit input - it has a button instead.
The Solution
You need to update your selectors to work with Site B. Based on the page snapshot, here are better selectors:
javascript
export const loginSelectors = {
Reply...
Sonnet 4.5

ii)using the files of the Sites used

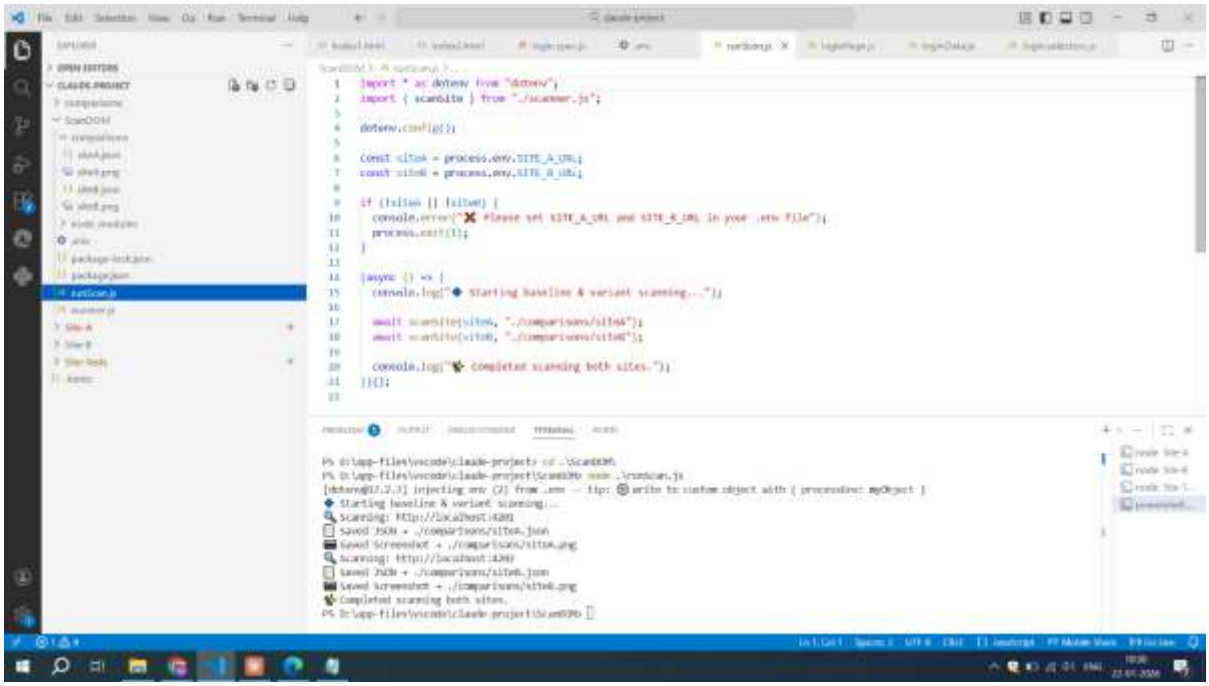


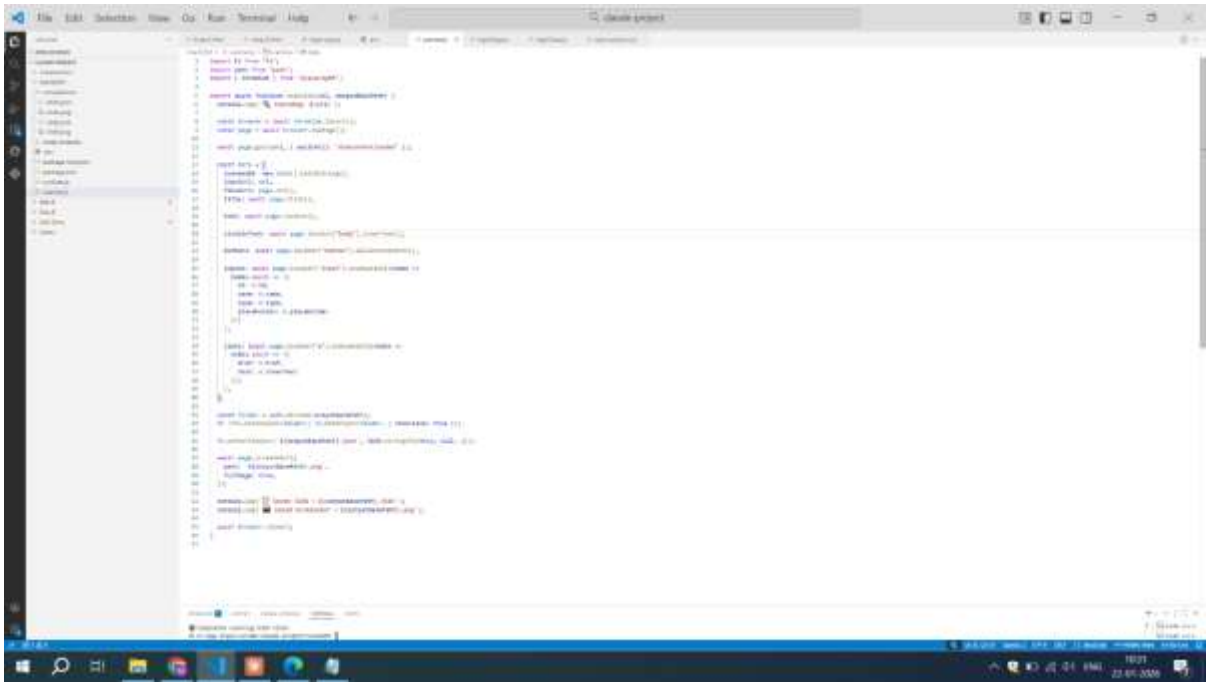


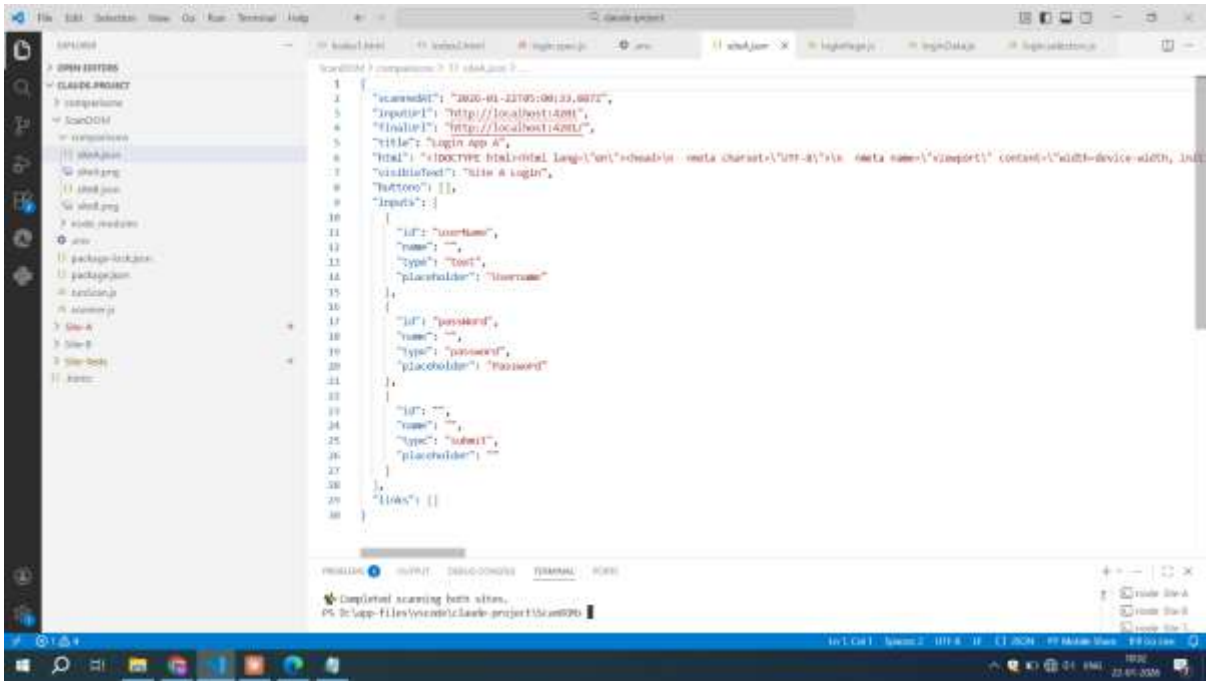

Claude
Claude project folder location
Share
New chat
Chats
Projects
Artifacts
Code
Recents
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zaheel
Free plan
Main Differences
1. ID naming convention:
• Site-A uses camelCase: userName , passWord
• Site-B uses kebab-case: user-name , pass-word
2. Login button type:
• Site-A uses <input type="submit" value="LOGIN" />
• Site-B uses <button type="submit">Login</button>
3. Password field placeholder:
• Site-A has a placeholder "Password"
• Site-B has NO placeholder on the password field
4. Form ID:
• Site-A: no form ID
• Site-B: id="loginForm"
5. Button ID:
• Site-A: no button ID
• Site-B: id="login-button"
The Problem
Reply...
Sonnet 4.5
Claude is AI and can make mistakes. Please double-check responses.

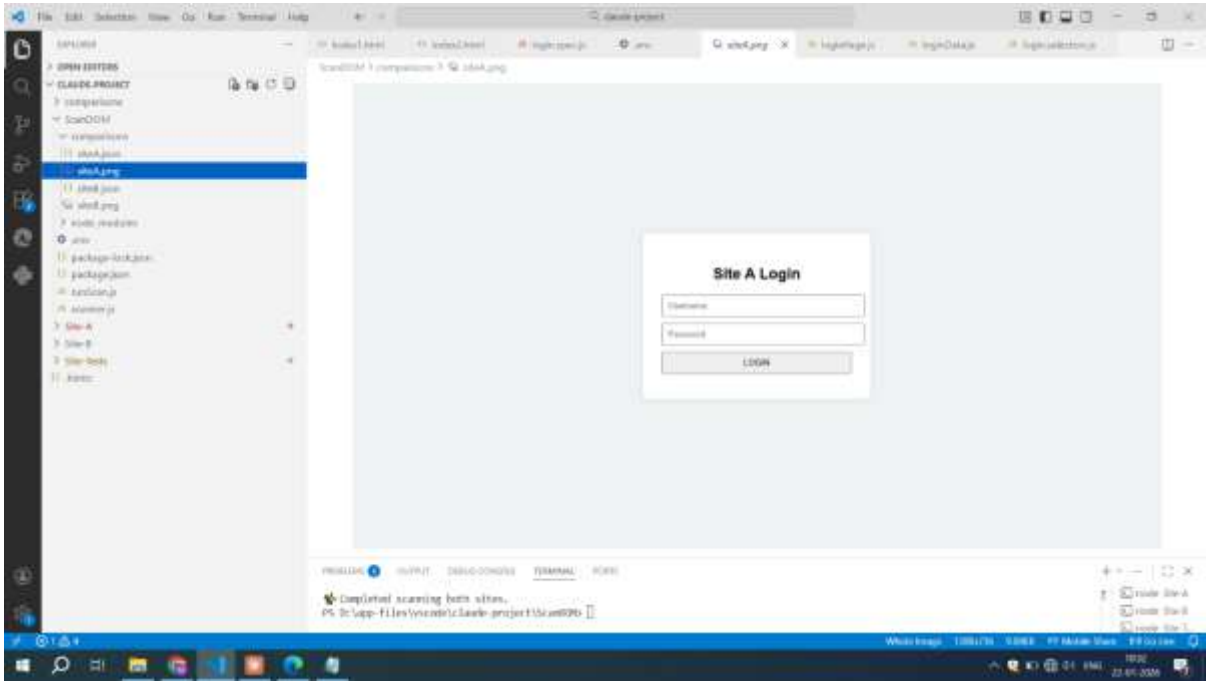

Claude
Claude project folder location
Share
New chat
Chats
Projects
Artifacts
Code
Recents
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zaheel
Free plan
The Problem
Your current selectors are designed for Site-A but you're testing against Site-B. That's why the test is failing!
Current selectors (match Site-A):
javascript
usernameField: "//input[@id='userName']",
passwordField: "//input[@id='passWord']",
loginButton: "//input[@type='submit' and @value='LOGIN']",
What Site-B needs:
javascript
usernameField: "//input[@id='user-name']",
passwordField: "//input[@id='pass-word']",
loginButton: "//button[@id='login-button']",
Would you like me to help you set up separate selector files for each site, or create a strategy to handle both sites in your tests?
Reply...
Sonnet 4.5
Claude is AI and can make mistakes. Please double-check responses.

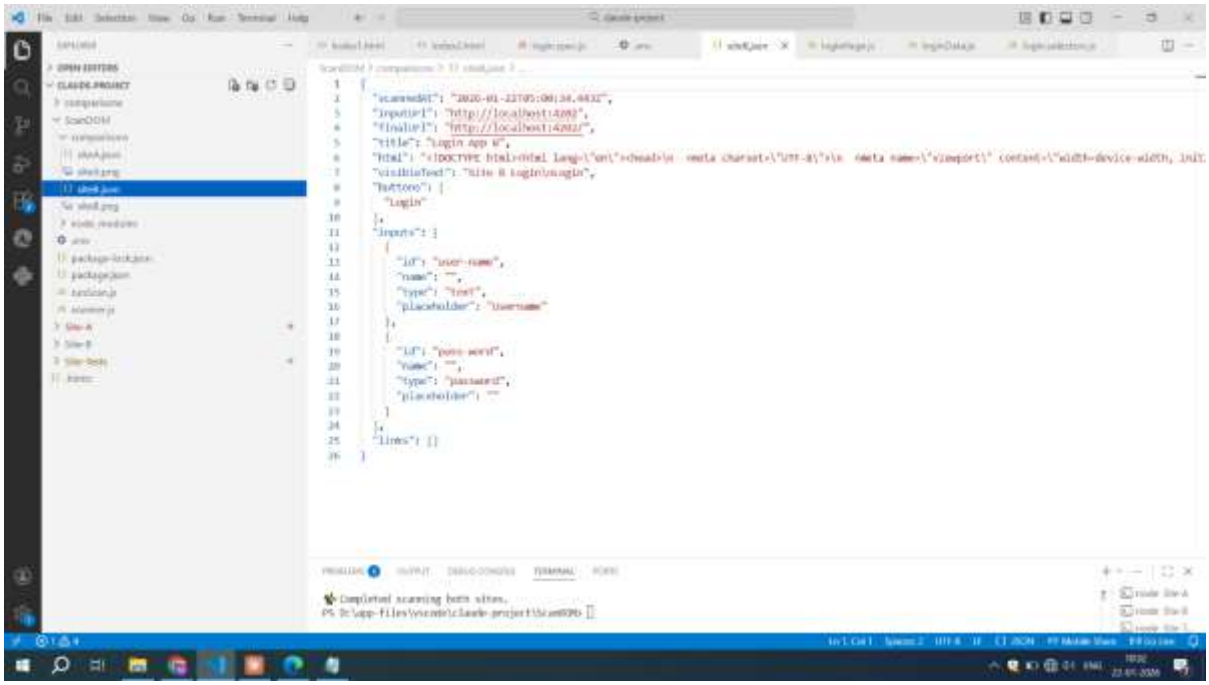
iii) using ScanDOM and its results

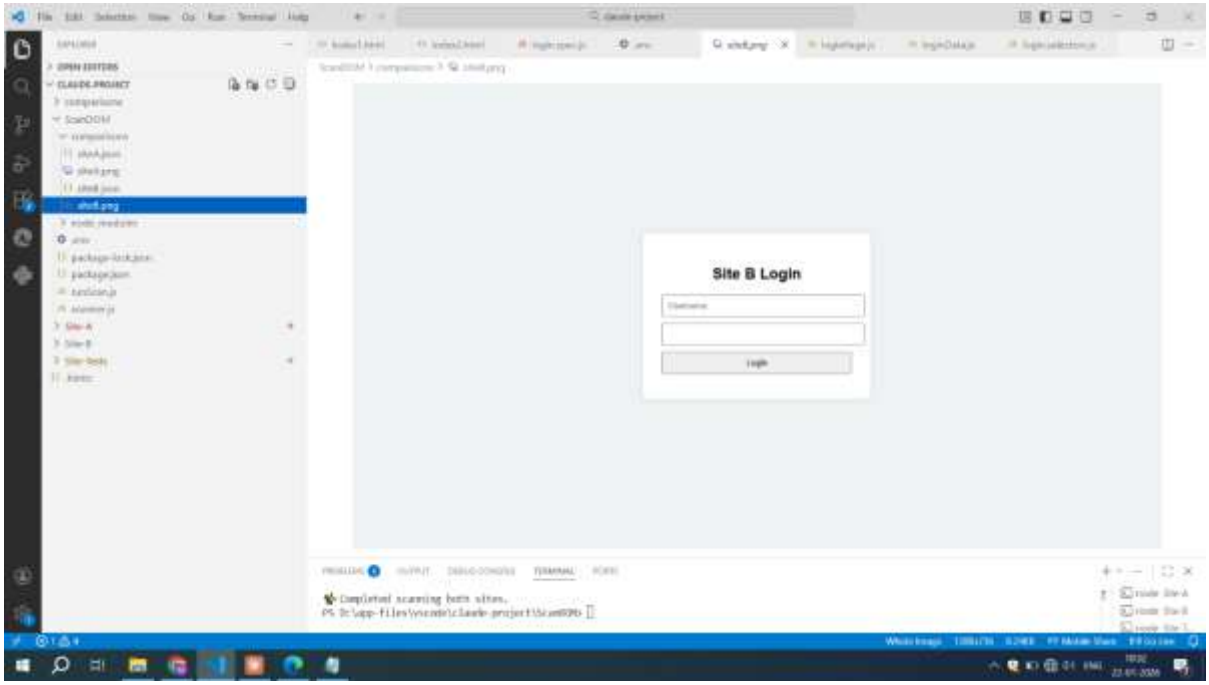


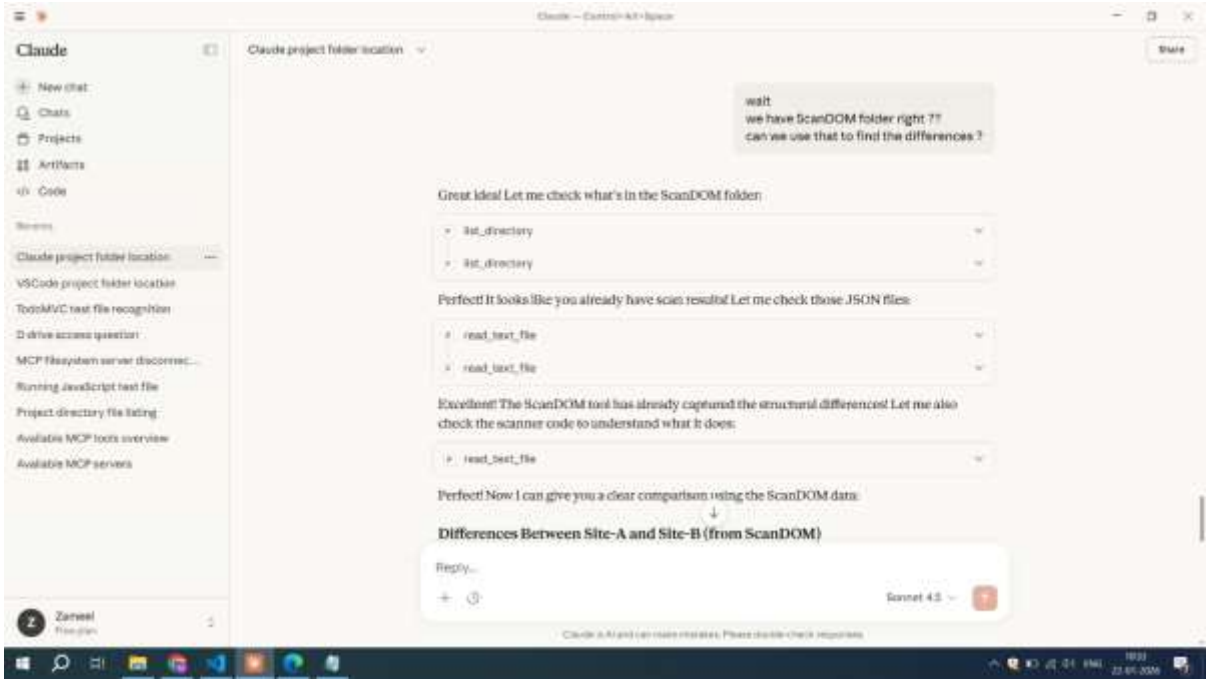


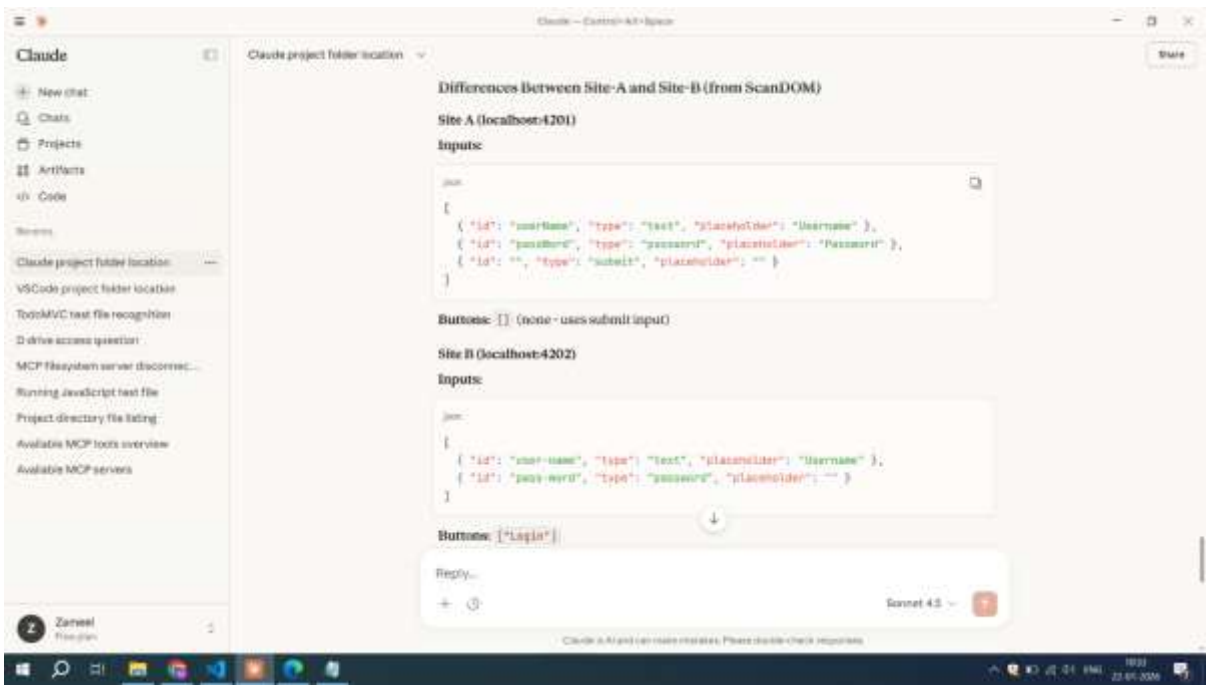

{
"scannedAt": "2026-01-22T05:00:33.607Z",
"inputUrl": "http://localhost:4001",
"finalUrl": "http://localhost:4001/",
"title": "Login App A",
"html": "<!DOCTYPE html><html lang=\"en\"><head>\n <meta charset=\"UTF-8\">\n <meta name=\"viewport\" content=\"width=device-width, ini
"visibleText": "Site A Login",
"buttons": [],
"inputs": [
{
"id": "userName",
"name": "",
"type": "text",
"placeholder": "Username"
},
{
"id": "password",
"name": "",
"type": "password",
"placeholder": "Password"
},
{
"id": "",
"name": "",
"type": "submit",
"placeholder": ""
}
],
"links": []
}
Completed scanning both sites.

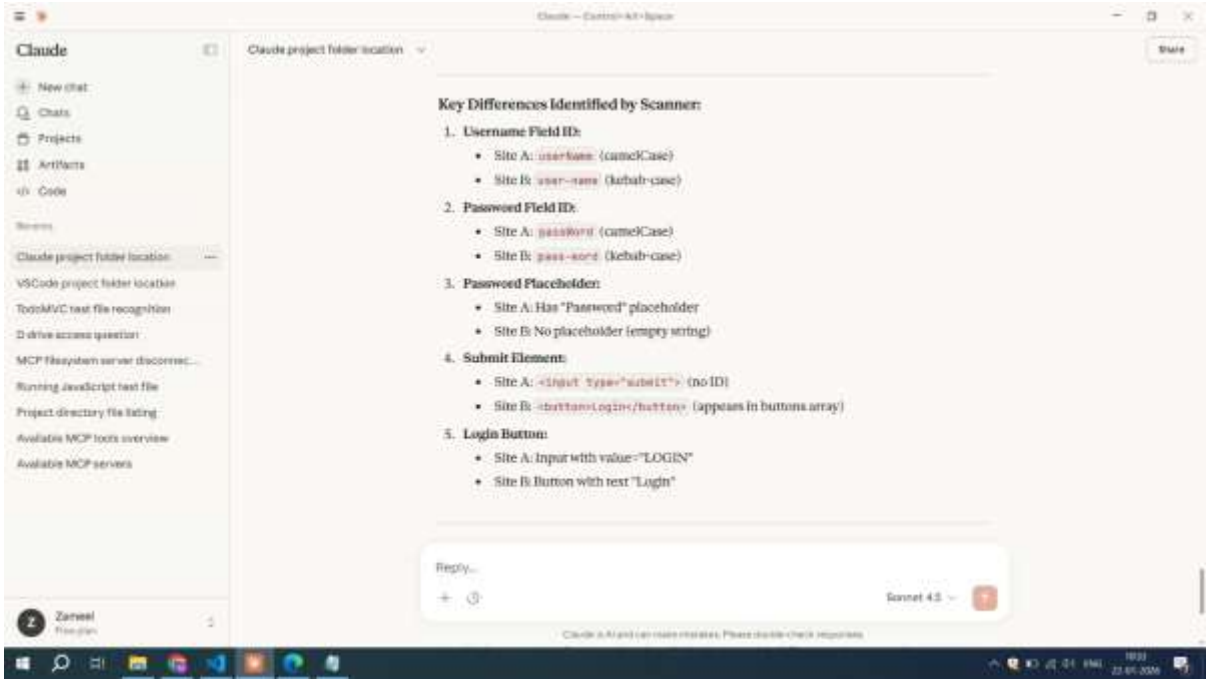

Site A Login
LOGIN
Completed scanning both sites.



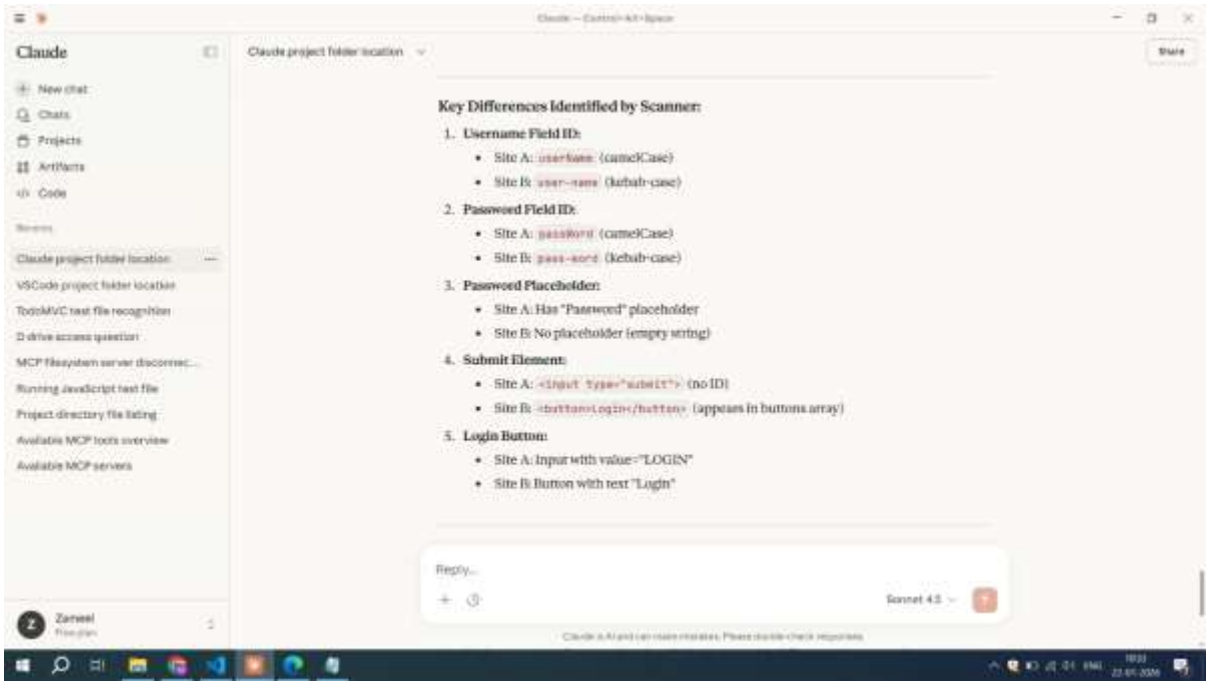

Site B Login

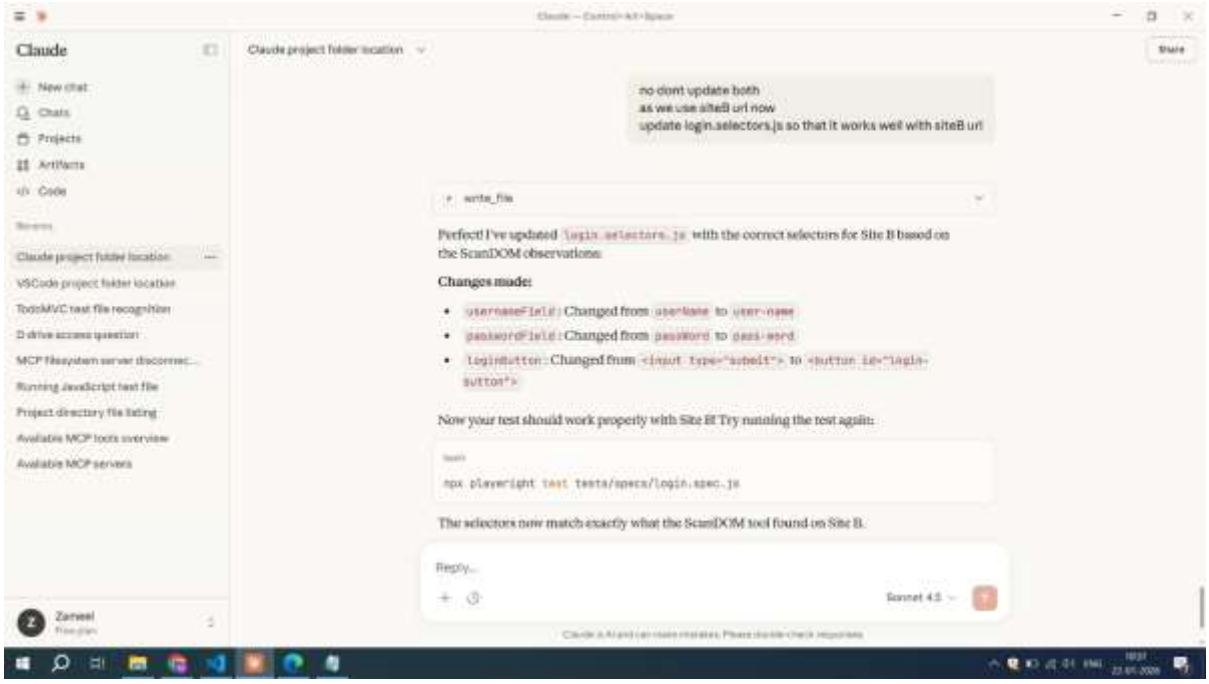

Claude
New chat
Chats
Projects
Artifacts
Code
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zaneel
Share
wait
we have ScanDOM folder right ??
can we use that to find the differences ?
Great idea! Let me check what's in the ScanDOM folder:
list_directory
list_directory
Perfect! It looks like you already have scan results! Let me check those JSON files:
read_text_file
read_text_file
Excellent! The ScanDOM tool has already captured the structural differences! Let me also check the scanner code to understand what it does:
read_text_file
Perfect! Now I can give you a clear comparison using the ScanDOM data:
Differences Between Site-A and Site-B (from ScanDOM)
Reply...
Sonnet 4.5

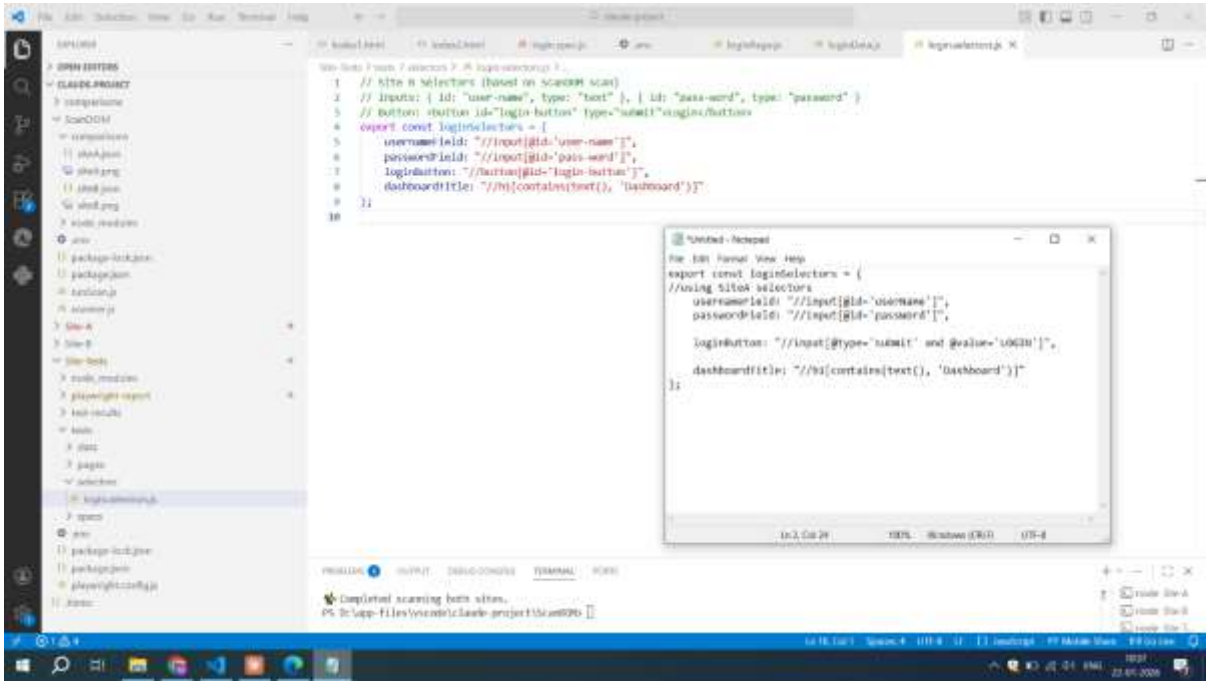

Claude
New chat
Chats
Projects
Artifacts
Code
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zaneel
Share
Differences Between Site-A and Site-B (from ScanDOM)
Site A (localhost:4201)
Inputs:
[
{ "id": "userName", "type": "text", "placeholder": "Username" },
{ "id": "passWord", "type": "password", "placeholder": "Password" },
{ "id": "", "type": "submit", "placeholder": "" }
]
Buttons: [] (none - uses submit input)
Site B (localhost:4202)
Inputs:
[
{ "id": "user-name", "type": "text", "placeholder": "Username" },
{ "id": "pass-word", "type": "password", "placeholder": "" }
]
Buttons: ["Login"]
Reply...
Sonnet 4.5



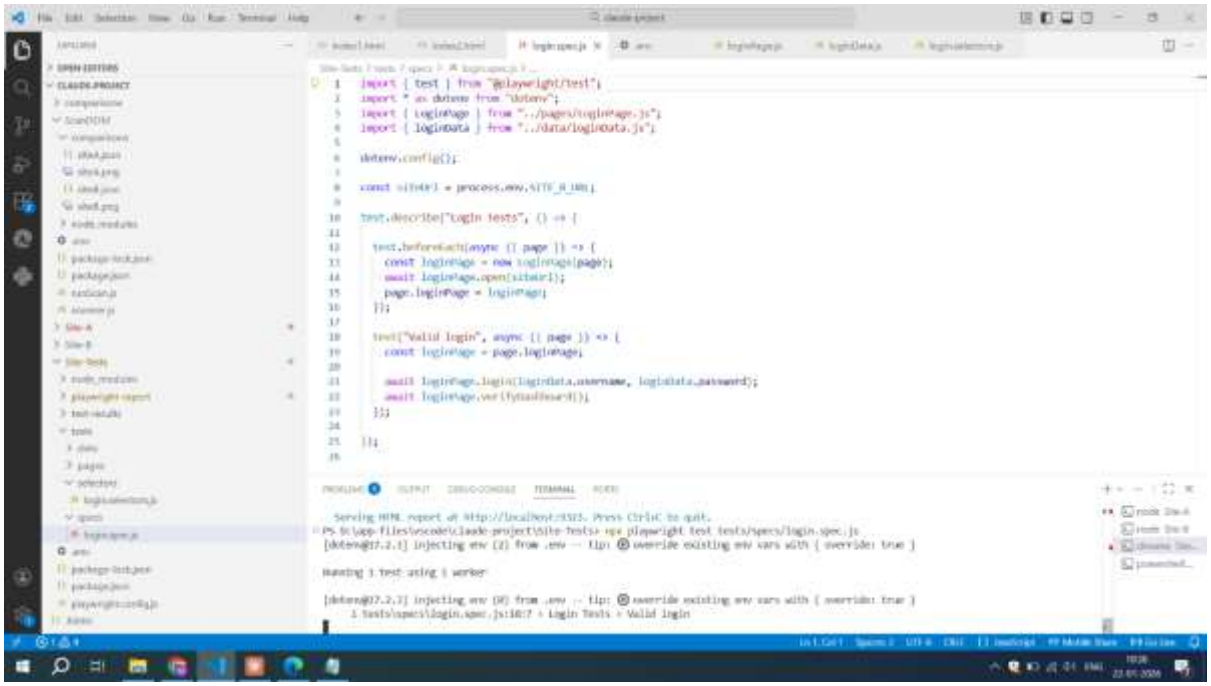
iv) using Filesystem protocol and Playwright protocol to make changes

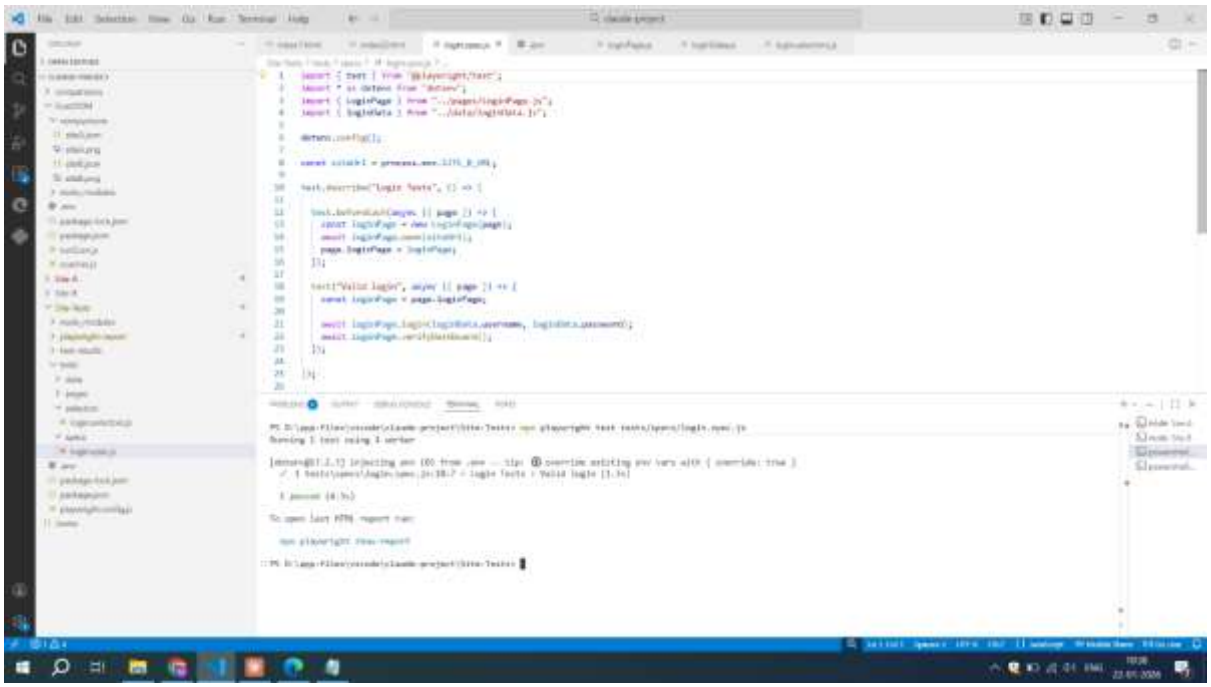


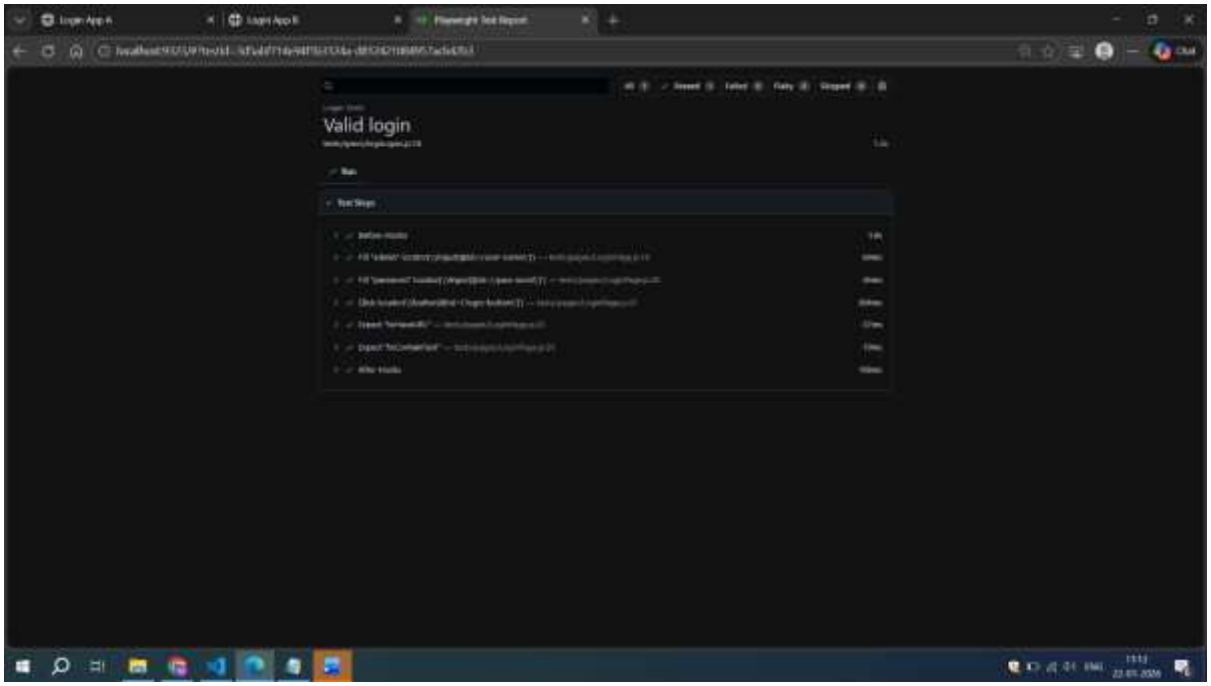

Claude
Claude project folder location
Share
New chat
Chats
Projects
Artifacts
Code
Claude project folder location
VSCode project folder location
TodoMVC test file recognition
D drive access question
MCP filesystem server disconnec...
Running JavaScript test file
Project directory file listing
Available MCP tools overview
Available MCP servers
Zaheel
no dont update both
as we use siteB url now
update login.selectors.js so that it works well with siteB url
write_file
Perfect! I've updated login.selectors.js with the correct selectors for Site B based on the ScanDOM observations:
Changes made:
usernameField: Changed from userName to user-name
passwordField: Changed from passWord to pass-word
loginButton: Changed from <input type="submit"> to <button id="login-button">
Now your test should work properly with Site B! Try running the test again:
bash
npx playwright test tests/specs/login.spec.js
The selectors now match exactly what the ScanDOM tool found on Site B.
Reply...
Sonnet 4.5


// Site A selectors (based on ScanDOM scan)
// Inputs: { id: "user-name", type: "text" }, { id: "pass-word", type: "password" }
// Button: <button id="login-button" type="submit">Login</button>
export const loginSelectors = {
usernameField: "//input[@id='user-name']",
passwordField: "//input[@id='pass-word']",
loginButton: "//button[@id='login-button']",
dashboardTitle: "//h1[contains(text(), 'Dashboard')]"
};
*Untitled - Notepad
File Edit Format View Help
export const loginSelectors = {
//using SiteA selectors
usernameField: "//input[@id='username']",
passwordField: "//input[@id='password']",
loginButton: "//input[@type='submit' and @value='LOGIN']",
dashboardTitle: "//h1[contains(text(), 'Dashboard')]"
};
Completed scanning both sites.

6) Running the tests






Valid login