

Hi!

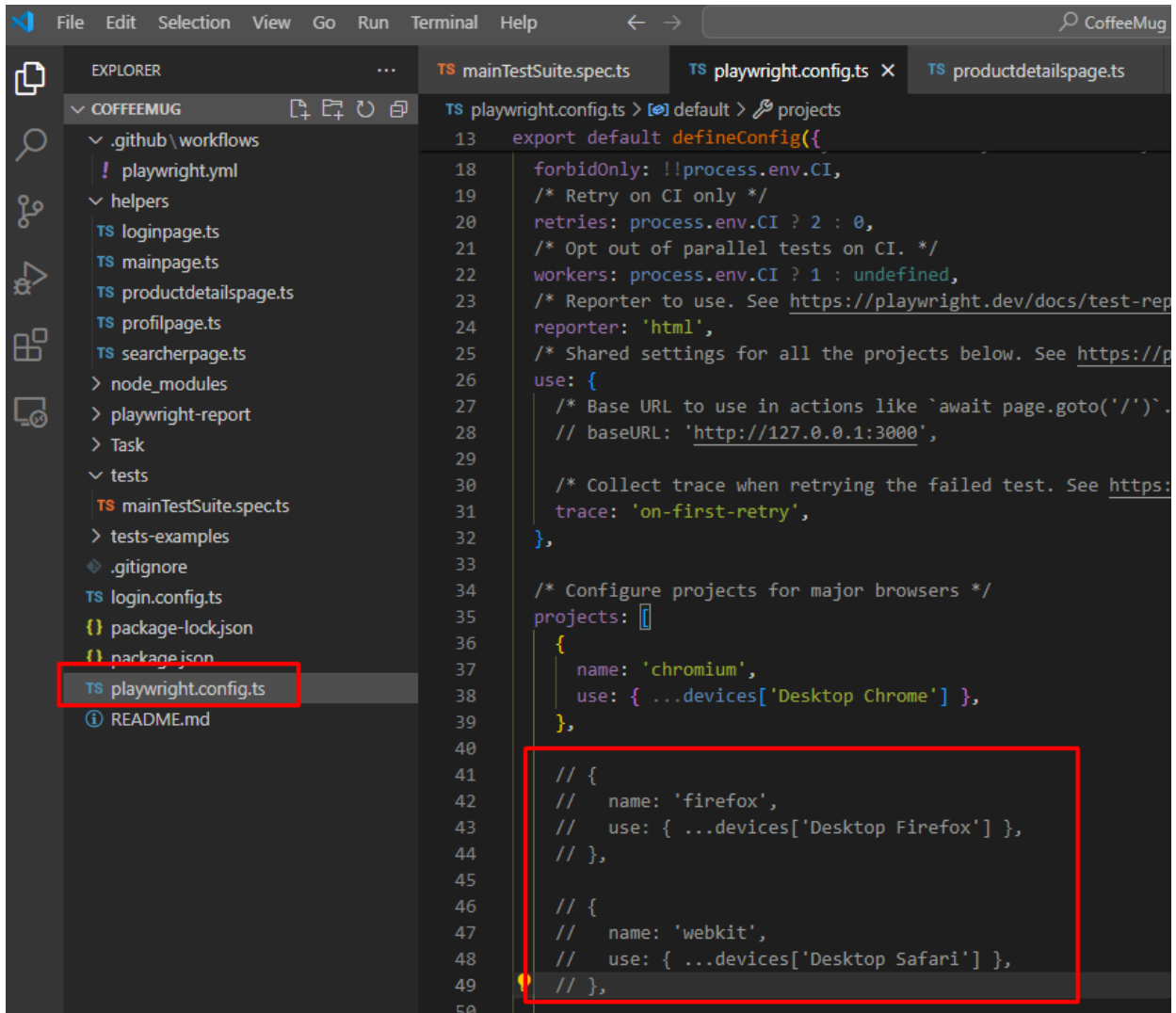
I want to explain a little my solutions in recruitment task which you've sent to me.

1. Website selected for tests.

My first idea was to choose one of biggest e-commerce platform – Amazon. Because it is well know and the website is English. But I had to find other example (I wanted to create tests on real page) because Amazon, Allegro and some other websites have captcha mechanism during registration. I didn't want to make any workarounds and I don't have access to any staging environment where captcha is off.

So I decided to choose Polish morele.net website.

2. Because we have one email to registrate tests are possible to run only for chromium, I commented part of code in playwright.config.ts file.



```
TS playwright.config.ts > [C] default > [P] projects
13 export default defineConfig({
18   forbidOnly: !!process.env.CI,
19   /* Retry on CI only */
20   retries: process.env.CI ? 2 : 0,
21   /* Opt out of parallel tests on CI. */
22   workers: process.env.CI ? 1 : undefined,
23   /* Reporter to use. See https://playwright.dev/docs/test-rep
24   reporter: 'html',
25   /* Shared settings for all the projects below. See https://p
26   use: {
27     /* Base URL to use in actions like `await page.goto('/')`.
28     // baseURL: 'http://127.0.0.1:3000',
29
30     /* Collect trace when retrying the failed test. See https:
31     trace: 'on-first-retry',
32   },
33
34   /* Configure projects for major browsers */
35   projects: [
36     {
37       name: 'chromium',
38       use: { ...devices['Desktop Chrome'] },
39     },
40
41     // {
42     //   name: 'firefox',
43     //   use: { ...devices['Desktop Firefox'] },
44     // },
45
46     // {
47     //   name: 'webkit',
48     //   use: { ...devices['Desktop Safari'] },
49     // },
50 }
```

3. To run tests you need to create gitignore file login.config.ts (described in README.md).

4. Task 1:

I needed to modify a little point with verification successful registration because on Morele page there is no information like 'registration success' or similar so I did workaround that after login to new account I'm checking if logged email is the same as email used to register new account:

```
async verifyNewAccount(email: string) {  
  await this.page.click('[href="/profil/"]');  
  await this.page.click(`//*[contains(text(), "Zweryfikuj e-mail")]`, {force: true});  
  expect(this.page.locator(`[value="${email}"]`)).toBeVisible;  
}
```

5. What also can be done as extension – for bigger project we could keep all locators in one file and pass them as imported parameters (similar as credentials from login.config.ts file). Some data-test-id's or other unique locators would be helpful but this is something we have no influence on when choosing a website which is not necessarily adapted to automatic testing.