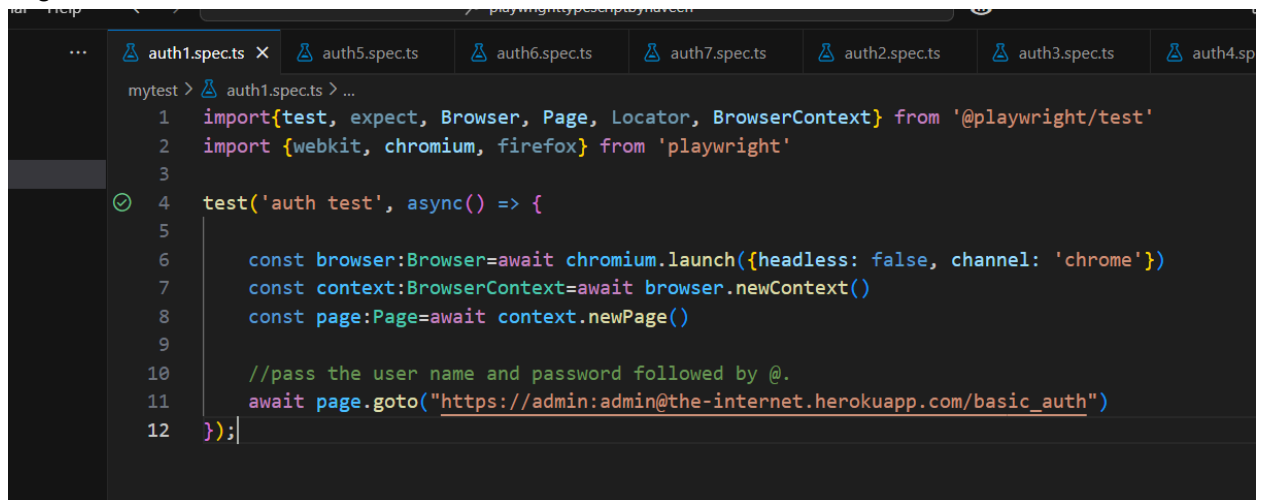
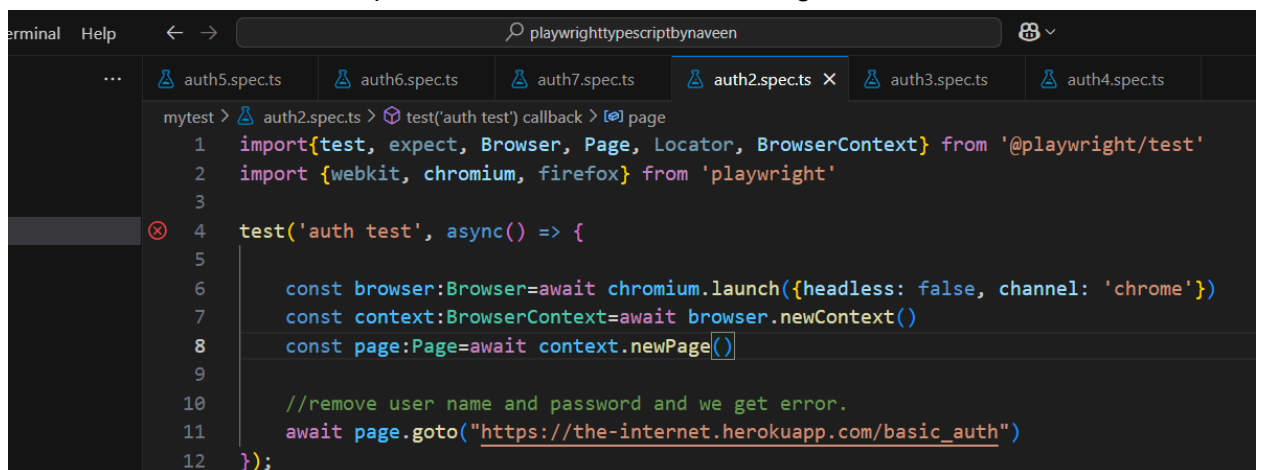


Login auth-

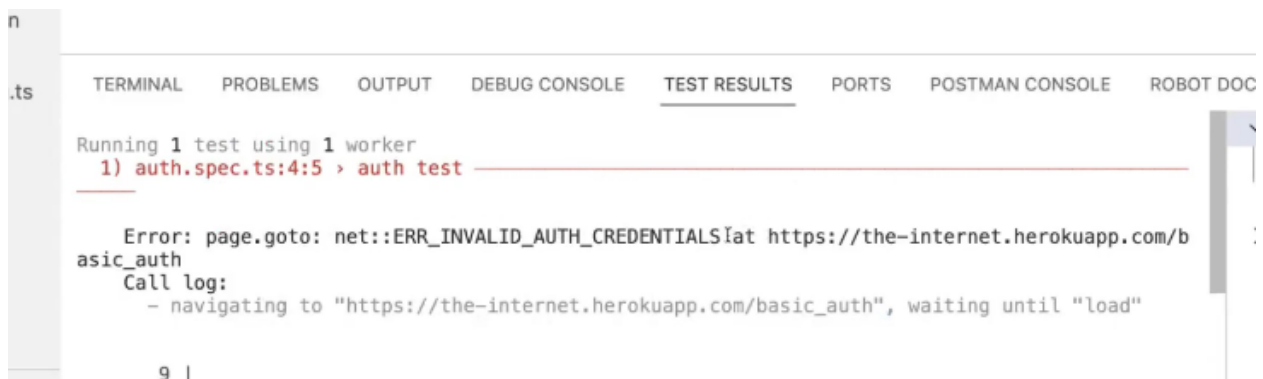


```
mytest > auth1.spec.ts > ...
1 import {test, expect, Browser, Page, Locator, BrowserContext} from '@playwright/test'
2 import {webkit, chromium, firefox} from 'playwright'
3
4 test('auth test', async() => {
5
6     const browser:Browser=await chromium.launch({headless: false, channel: 'chrome'})
7     const context:BrowserContext=await browser.newContext()
8     const page:Page=await context.newPage()
9
10    //pass the user name and password followed by @.
11    await page.goto("https://admin:admin@the-internet.herokuapp.com/basic_auth")
12 });
```

Remove the user name and password and run and see if we get error



```
mytest > auth2.spec.ts > test('auth test') callback > page
1 import {test, expect, Browser, Page, Locator, BrowserContext} from '@playwright/test'
2 import {webkit, chromium, firefox} from 'playwright'
3
4 test('auth test', async() => {
5
6     const browser:Browser=await chromium.launch({headless: false, channel: 'chrome'})
7     const context:BrowserContext=await browser.newContext()
8     const page:Page=await context.newPage()
9
10    //remove user name and password and we get error.
11    await page.goto("https://the-internet.herokuapp.com/basic_auth")
12 });
```

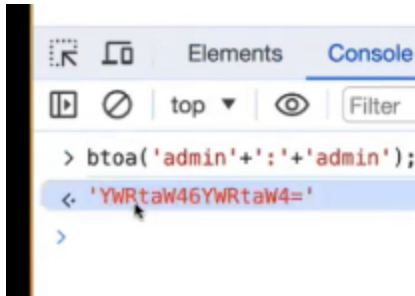


```
Running 1 test using 1 worker
1) auth.spec.ts:4:5 > auth test

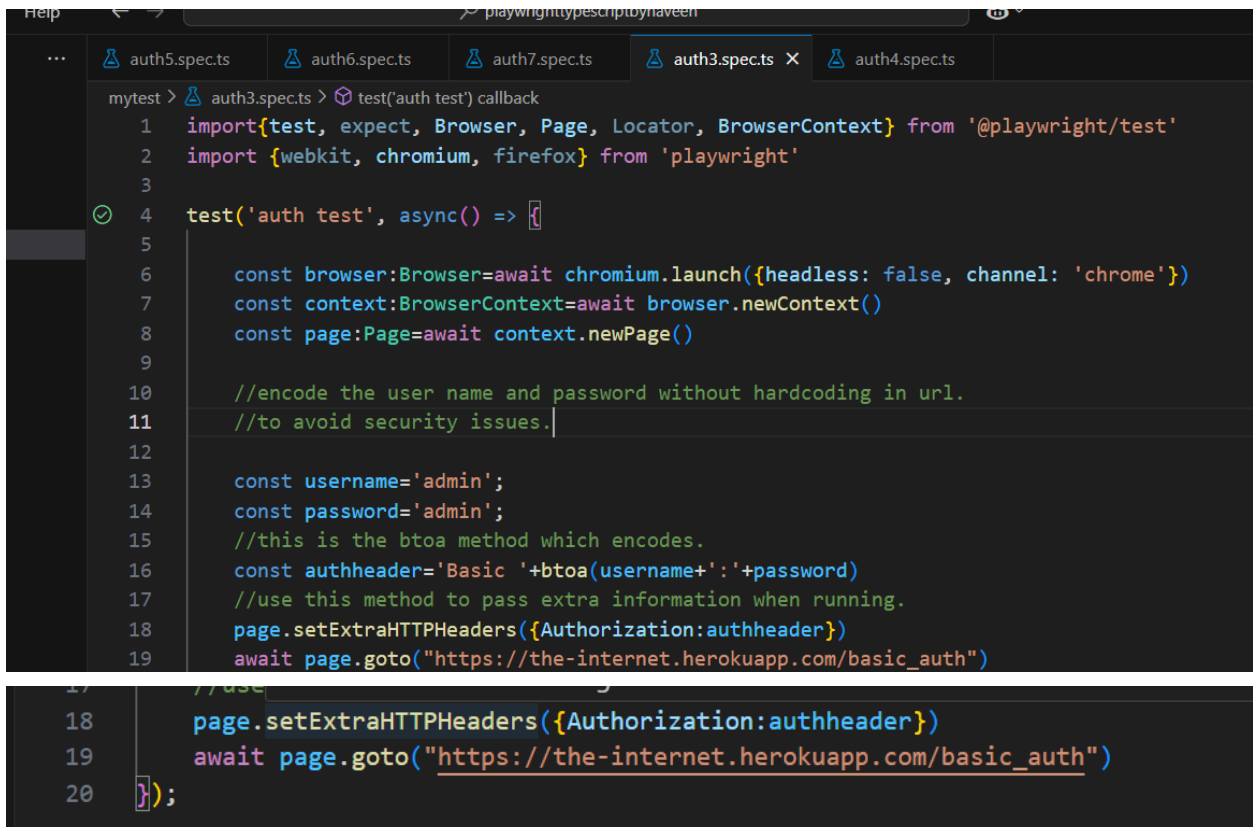
Error: page.goto: net::ERR_INVALID_AUTH_CREDENTIALS at https://the-internet.herokuapp.com/basic_auth
Call log:
- navigating to "https://the-internet.herokuapp.com/basic_auth", waiting until "load"

9 |
```

How basic authentication is encoded to base 64-
You write in console using btoa method.



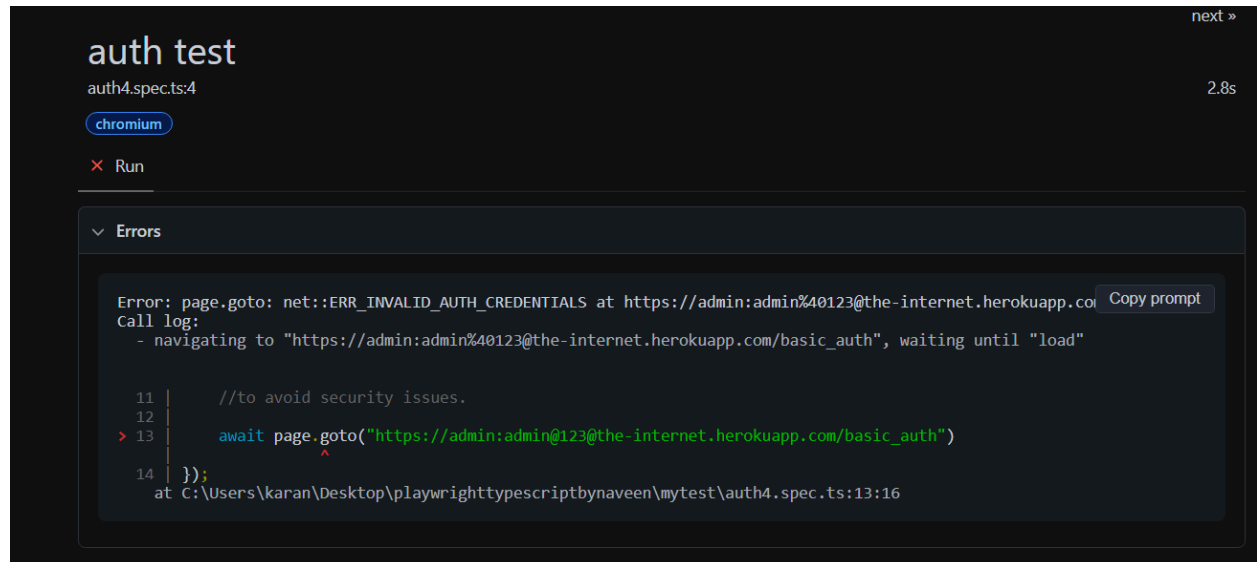
Add the same concept in vs code-



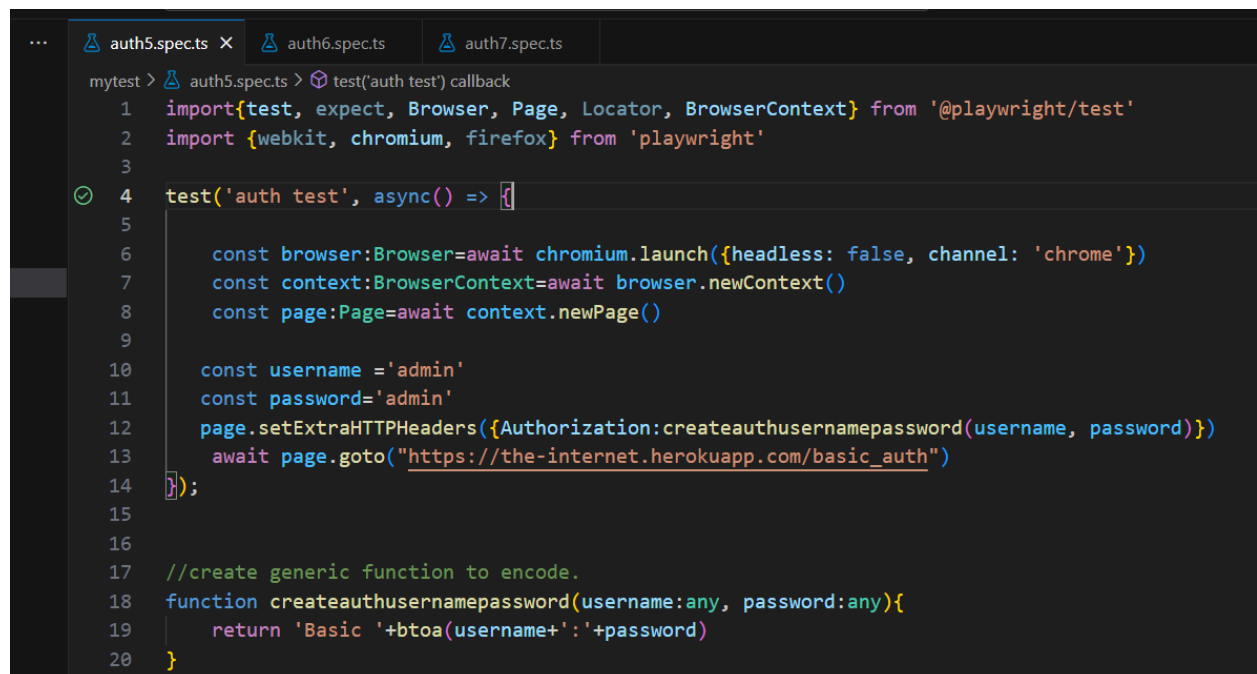
Problem with hardcoding-

We cannot have @ in the user name or password, else it will consider wrong url, wrong user name and password.

```
14
15    await page.goto('https://admin:admin@123@the-internet.herokuapp.com/basic_auth');
16
17
```



Added function-



Run on firefox-

```
...  auth6.spec.ts x  auth7.spec.ts
mytest >  auth6.spec.ts > test('auth test') callback > context
1  import {test, expect, Browser, Page, Locator, BrowserContext} from '@playwright/test'
2  import {webkit, chromium, firefox} from 'playwright'
3
4  test('auth test', async() => {
5
6      //run on firefox and see.
7      const browser:Browser=await firefox.launch({headless: false})
8      const context:BrowserContext=await browser.newContext()
9      const page:Page=await context.newPage()
10
11      const username ='admin'
12      const password='admin'
13      page.setExtraHTTPHeaders({Authorization:createauthusernamepassword(username, password)})
14      await page.goto("https://the-internet.herokuapp.com/basic_auth")
15  });
16
17
18  //create generic function to encode.
19  function createauthusernamepassword(username:any, password:any){
20      return 'Basic '+btoa(username+':'+password)
21  }
```

Headless run –

```
Run  Terminal  Help  playwrighttypescripby naveen
...  auth7.spec.ts x
mytest >  auth7.spec.ts > test('auth test') callback
1  import {test, expect, Browser, Page, Locator, BrowserContext} from '@playwright/test'
2  import {webkit, chromium, firefox} from 'playwright'
3
4  test('auth test', async() => {
5
6      //run on headless mode.
7      const browser:Browser=await chromium.launch({headless: true})
8      const context:BrowserContext=await browser.newContext()
9      const page:Page=await context.newPage()
10
11      const username ='admin'
12      const password='admin'
13      page.setExtraHTTPHeaders({Authorization:createauthusernamepassword(username, password)})
14      await page.goto("https://the-internet.herokuapp.com/basic_auth")
15  });
16
17
18  //create generic function to encode.
19  function createauthusernamepassword(username:any, password:any){
20      return 'Basic '+btoa(username+':'+password)
21  }
```