

pavanpw-34

Page Object Model Pattern

pages folder can be created at tests or at root level.

page classes-

```
1  //
2  exports.LoginPage =
3  class LoginPage {
4
5      //constructors will have page object
6      //constructor will also have basic locators.
7      constructor(page) {
8          this.page = page;
9          this.loginLink="#login2"
10         this.usernameInput = '#loginusername';
11         this.passwordInput = '#loginpassword';
12         this.loginButton = '//button[normalize-space()='Log in]';
13     }
14
15     async gotoLoginPage(){
16         await this.page.goto('https://www.demoblaze.com/index.html');
17     }
18
19     async login(username, password) {
20         await this.page.locator(this.loginLink).click();
21         await this.page.locator(this.usernameInput).fill(username);
22         await this.page.locator(this.passwordInput).fill(password);
23         await this.page.locator(this.loginButton).click();
24     }
25 }
26 }
```

```

1  //
2  exports.HomePage = class HomePage {
3
4      constructor(page) {
5          this.page = page;
6          this.productList= '/*[@id="tbodyid"]/div/div/div/h4/a';
7          this.addToCartbtn='a[normalize-space()="Add to cart"]';
8          this.cart='#cartur'
9      }
10
11
12      async addProductToCart(productName) {
13          const productList = await this.page.$$$(this.productList);
14          for (const product of productList) {
15              if (productName === await product.textContent()) {
16                  await product.click()
17                  break;
18              }
19          }
20          await this.page.on('dialog', async dialog=>{
21              if(dialog.message().includes('added')){
22                  await dialog.accept();
23              }
24          })
25          await this.page.locator(this.addToCartbtn).click();
26      }
27
28      async gotoCart() {
29          await this.page.locator(this.cart).click();
30      }
31
32  }

```

```
1  //
2  exports.CartPage = class CartPage {
3
4      constructor(page) {
5          this.page = page;
6          this.noOfProducts= '//tbody[@id="tbodyid"]/tr/td[2]'
7      }
8
9
10     async checkProductInCart(productName) {
11         const productsInCart=await this.page.$$$(this.noOfProducts)
12         for (const product of productsInCart) {
13             console.log(await product.textContent())
14             if (productName === await product.textContent()) {
15                 return true;
16                 break;
17             }
18         }
19     }
20 }
21
22 }
```

codesnap.dev

test class-

```
1 //
2 import { test, expect } from '@playwright/test';
3 import { LoginPage } from '../pages/LoginPage'; //login page is name of the class.
4 import { HomePage } from '../pages/HomePage';
5 import { CartPage } from '../pages/CartPage';
6
7
8 test('test', async ({ page }) => {
9
10     //Login
11     const login=new LoginPage(page); //create object of page first.
12     await login.gotoLoginPage();
13     await login.login('pavanol', 'test@123')
14     await page.waitForTimeout(3000)
15
16     //Home
17     const home=new HomePage(page)
18     await home.addProductToCart("Nexus 6")
19     await page.waitForTimeout(3000)
20
21
22     //Cart
23     const cart=new CartPage(page)
24     await page.waitForTimeout(3000)
25     const status=await cart.checkProductInCart('Nexus 6')
26     expect(await status).toBe(true);
27 });
```