**Trong fordel tests -> fordel acceptance -> file navigation.feature:**

Feature: Test navigation between pages

We can have a longer description

That can span a few lines

Scenario: Homepage can go to Blog

Given I am on the homepage

When I click on the link with id “blog-link”

Then I am on the blog page

Scenario: Blog can go to Homepage

Given I am on the homepage

When I click on the link with id “home-link”

Then I am on the homepage

**Trong fordel tests -> fordel acceptance -> fordel steps -> file navigation.py:**

from behave import \*

from selenium import webdriver

use\_step\_matcher(‘re’)

@given(‘ I am on the homepage’)

def step\_impl(context):

context.browser = webdriver.Chrome()

context.brower.get(‘http://127.0.0.1:5000’)

@given(‘ I am on the blog page’)

def step\_impl(context):

context.browser = webdriver.Chrome()

context.brower.get(‘http://127.0.0.1:5000/blog’)

@then(‘I am on the blog page’)

Def step\_impl(context):

Expected\_url = ‘http://127.0.0.1:5000/blog’

Assert context.brower.current\_url == expected\_url

@then(‘I am on the homepage’)

Def step\_impl(context):

Expected\_url = ‘http://127.0.0.1:5000’

Assert context.brower.current\_url == expected\_url

**Trong fordel tests -> fordel acceptance -> fordel steps -> file interactions.py:**

from behave import \*

use\_step\_matcher(‘re’)

@when(‘I click on the link with id”(.\*)”’)

def step\_impl(context, link\_id) :

link = context.browser.find\_element\_by\_id(link\_id)

link.click()

**Trong fordel tests -> fordel acceptance -> file content.feature:**

Feature: Test that pages have correct content

Scenario: Blog page has a correct title

Given I am on the blog page

Then There is a title shown on the page

And The title tag has content “This ia the blog page”

**Trong fordel tests -> fordel acceptance -> fordel steps -> file content.py:**

from behave import \*

from selenium.webdriver.common.by import By

use\_step\_matcher(‘re’)

@then(‘There is a title shown on the page’)

def step\_impl(context, link\_id) :

title\_tag = context.browser.find\_element(By.TAG\_NAME, ‘h1’)

assert title\_tag.is\_displayed()

@step(‘The title tag has content “(.\*)”’)

def step\_impl(context, context):

title\_tag = context.browser.find\_element(By.TAG\_NAME, ‘h1’)

assert title\_tag.text = content()