class **LoginPage**(BasePage)

def should\_be\_login\_page(self):

self.should\_be\_login\_url()

self.should\_be\_login\_form()

self.should\_be\_register\_form()

class **BasePage ()**

def \_\_init\_\_(self, **browser, url,** timeout = 10) : …

def **open**(self): …

def **is\_element\_present**(self, how, what):…

test\_main\_page.py

def test\_guest\_can\_go\_to\_login\_page(**browser**)

page = **MainPage**(browser, link)

page .open()

lp = page.go\_to\_login\_page()

def test\_guest\_should\_see\_login\_link(browser):

page = MainPage(browser, link)

page.open()

page.should\_be\_login\_link()

class **MainPage( BasePage )**

def **go\_to\_login\_page**(self):

self.browser.find\_element(

\*MainPageLocators.LOGIN\_LINK)

.click()

return LoginPage(browser, url)

def **should\_be\_login\_link**(self):

assert self.is\_element\_present(

\*MainPageLocators.LOGIN\_LINK)

conftest.py

@pytest.fixture(autouse=True)

def browser(request):

… …