* Create a python project. You can choose an IDE of your own. However, I have preferred to use PyCharm as it is most industry wide used.
* Create a new Virtual Environment.
* Create a package named **tests**. Entire framework is be divided into divided into different packages. Other would be subsequently created in later steps.
* Create a class for each testcase. Every testcase should be implemented as pytest method **ex: def test\_e2e(self)** and every pytest method should be wrapped under a class.
* Use fixtures for *setup* and *teardown* methods. Place the browser invocation and closure codes inside fixtures.
* Generalise the fixture methods for all testcases by placing them in **contest.py** file.
* Create a **utilities** package and place all the reusable classes, methods there. Doing so makes our code less reductant.
* Create a Base class inside utilities package and call the fixture there **@pytest.mark.usefixtures("setup")**. Inherit **Base** class to all the testcases.
* Create a package **resources**. Place browser drivers **(ex: chromedriver.exe, geckodriver)**, input\_data files, etc there.
* Pass command line arguments to select browser at runtime. Means we should be able to select which browser we need to run our cases on. **(ex: py.test --browser\_name=chrome)**. Use **pytest\_addoption(parser)** method and **request.config.getoption("browser\_name")** function to achieve this.
* Implement pageObject mechanism:
  + Create a package named pageObjects.
  + Create separate classes for each page **(ex: HomePage, LoginPage, BookingPage, etc)**.
  + Define constructor which will initialize the driver which was passed form the testcases.
  + Define objects and their locators as class variables
  + Define getter methods for each class variable.
* Create package actions. Define all the actions(like click, sendkeys, select, etc). Create and object for actions in each testcase and use actions object to call all the actions.
* Also if there are any other reusable functions, you can place them in Base class inside utilities package.
* Implement the logging feature. Use **logging** package to achieve this.
* Implement HTML reports: Use **pytest-html** to achieve this. Run the cases with **--html=report.html** parameter which helps us to create html reports.
* Also add screenshots for failed cases. To achieve this we should tweak few functions inside pytest-html package, you can find the tweaked code in the my framework code.
* Parameterize data from external excel sheet using **openpyxcl** library.