System Requirements Specification Index

For

Pyspark Use Case

Refactor the job ETL --- L1

1.0

IIHT Pvt. Ltd.

[fullstack@iiht.com](mailto:fullstack@iiht.com)

ETL\_JOB.PY

* User is provider with ~5 GB CSV input and crashes with **OutOfMemoryError: Java heap space**.

**Objective**

* Refactor the job to avoid driver OOM **without changing cluster size**. Hint: remove collect() and keep transformations on the cluster.

**Running the testcase**

* use the code python -m unittest to run the testcase
* There are total three testcase you need to follow the hint to clear all the testcase

**Execution Steps to Follow:**

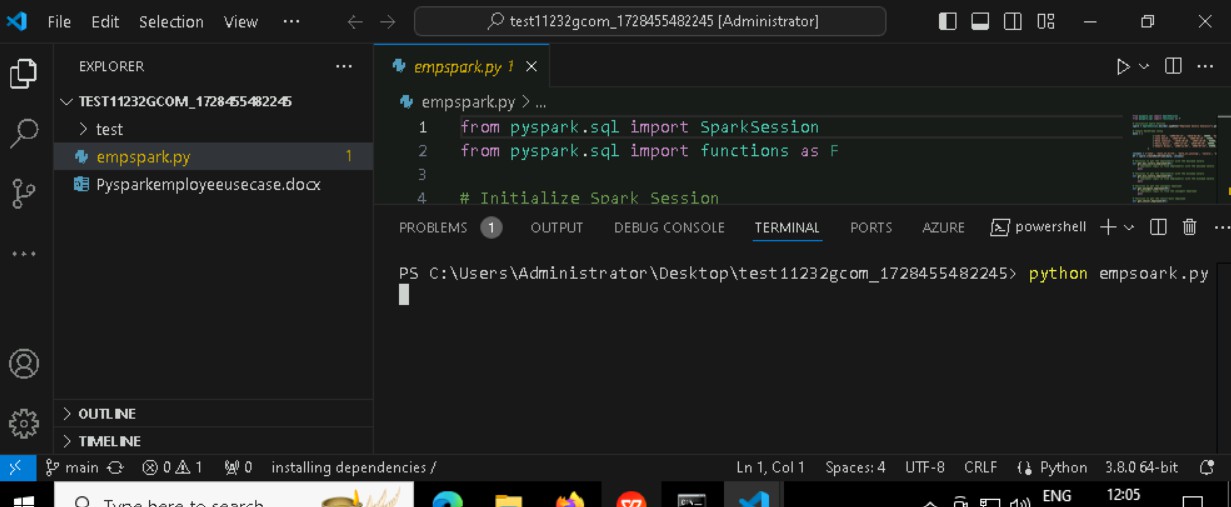
1. All actions like build, compile, running application,running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
3. This editor Auto Saves the code
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page)
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To launch application:

**Python empspark.py**

1. To run Test cases:

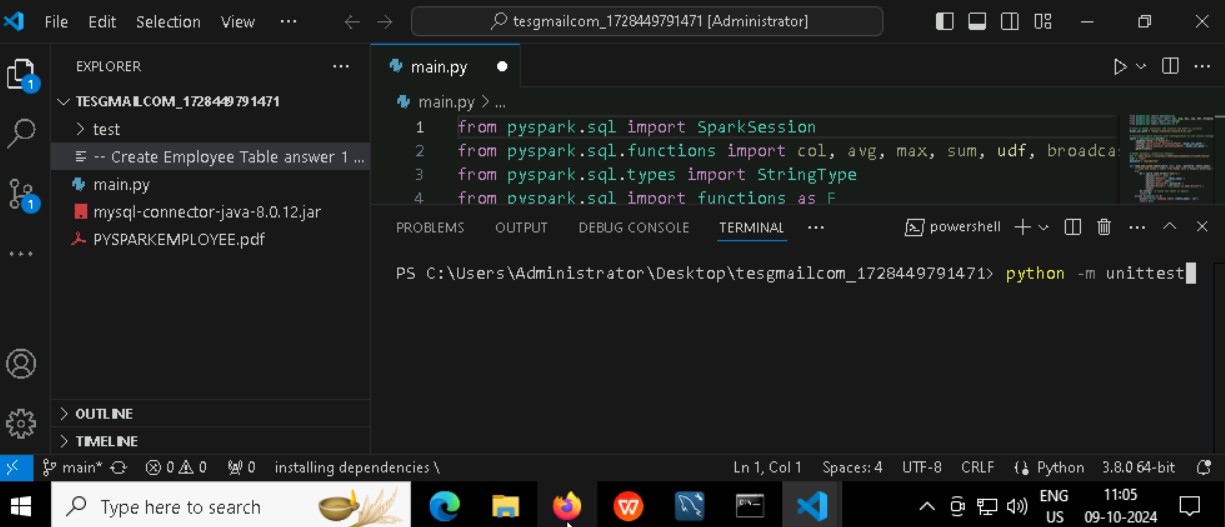
**python -m unittest**

**Screen shot to run the program**



**To run the application**

* + **Python .py**



**To run the testcase**

* + **Python -m unittest**

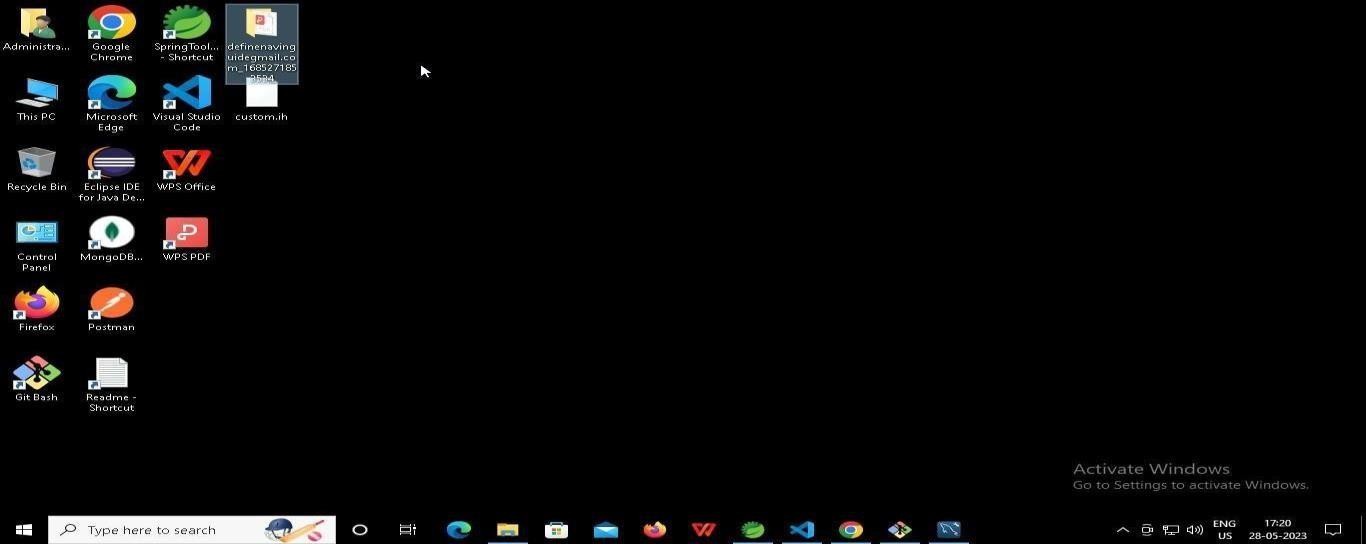
**Screenshot to push the application to github**

x

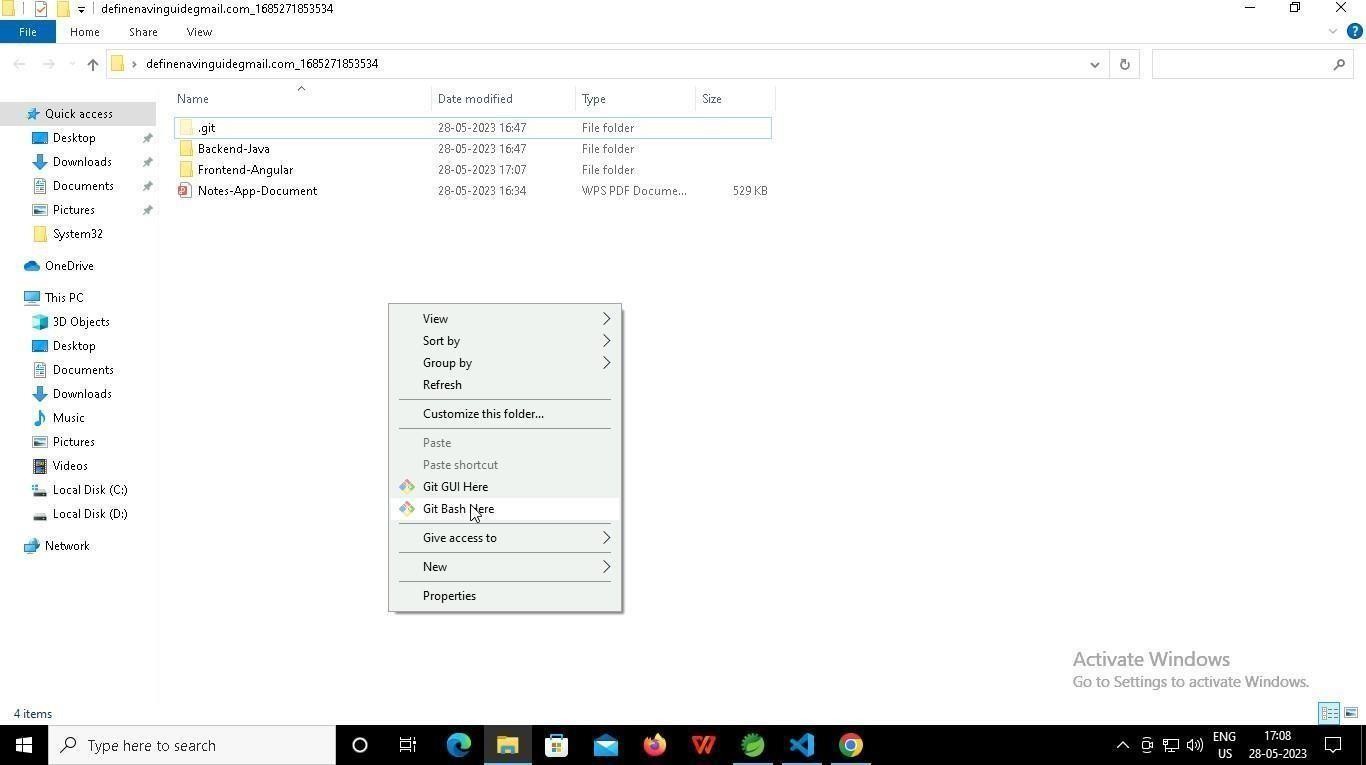
*You can run test cases as many numbers of times and at any stage of Development, to check howmany test cases arepassed/failed and accordingly refactor your code.*

1. **Make sure before final submission you commit all changes to git**. For that

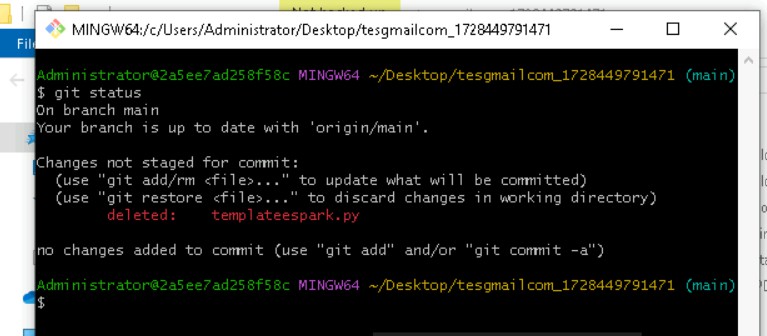
open theproject folder available on desktop



* 1. **Right click in folder and open Git Bash**

****

* 1. **In Git bash terminal, run following commands**
  2. **git status**

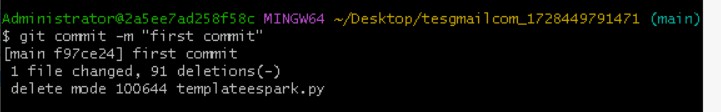


* 1. **git add .**

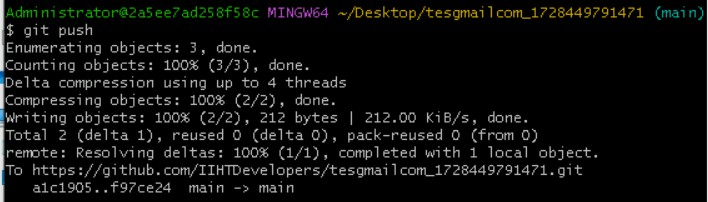
****

* 1. git commit -m “First commit”

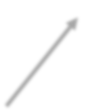
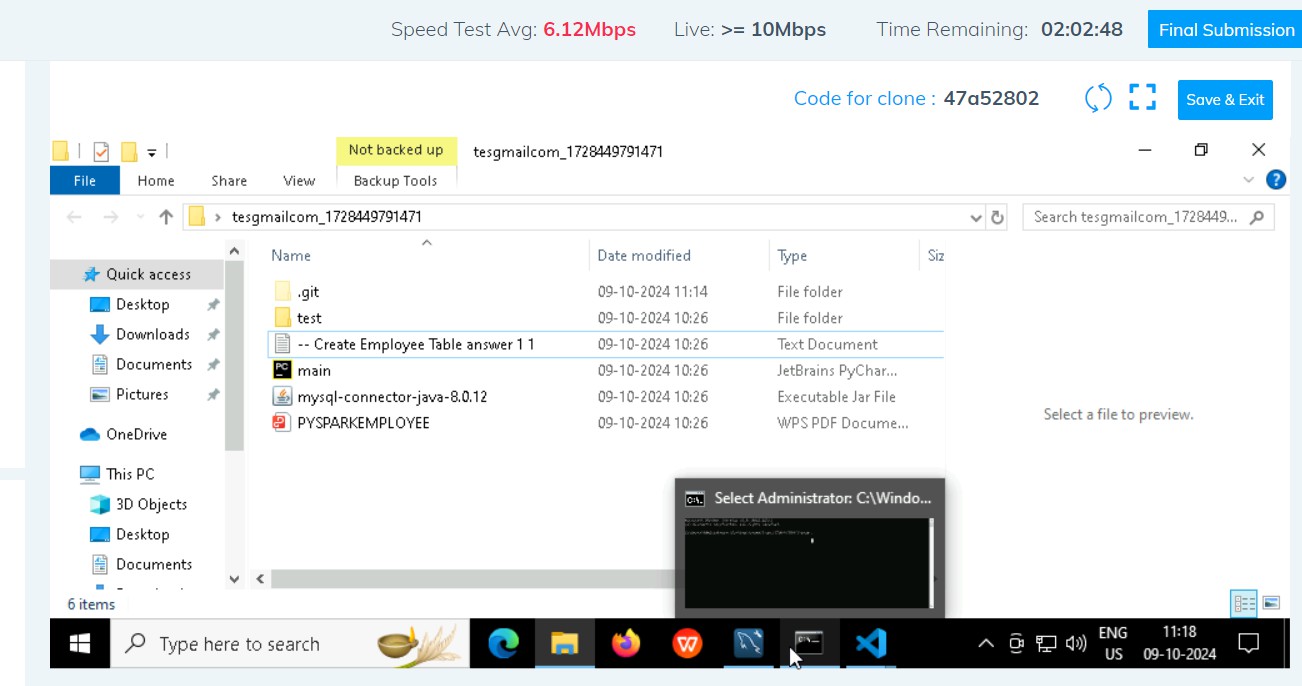
(You can provide any message every time you commit)



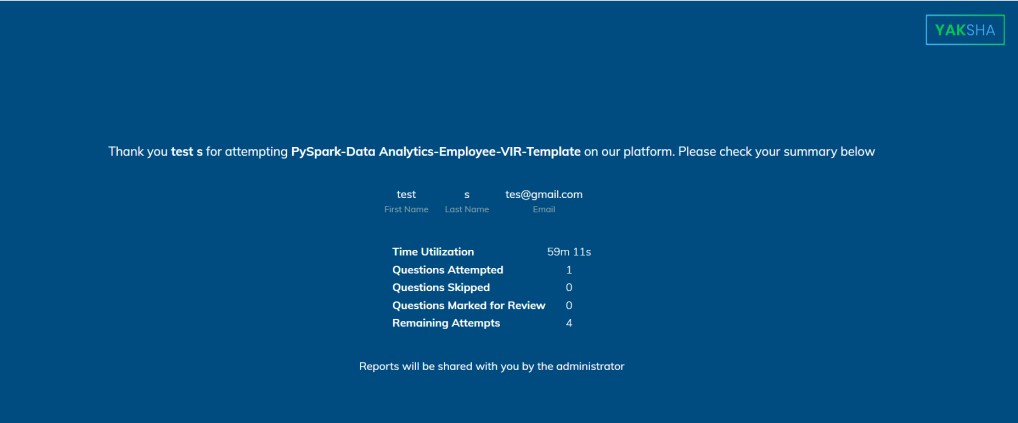
* 1. **git push**

****

After you have pushed your code Finally click on the final submission button



You should see a screen like this you will have to wait for the results . after getting this page you can leave the system



x