
System Requirements Specification Index

For

Pyspark Usecase

Employee Salary and Tenure AnalysisVersion L1

1.0

Step to access the work environment

Step 1 use the URL to login provide the username and password

PySpark-Data Analytics-Employee-L1-VIR-Template

180 Mins

1 Sections

1 Skills

1 Questions

2 Total Attempts

0

60% - Cut Off

100%

System Requirements

- Recommended Browser (Chrome, Safari, Etc)
- Javascript should be enabled in the browser

Link Validity and Cut-off Details

- Link Validity Start Date and Time - 6/9/2024, 5:47 PM
- Cut Off Date and Time - 30/9/2025, 2:50 PM

YAKSHA

Registration Details


First Name *

Last Name *

Email *

Phone (Optional)

☐ I'm not a robot


reCAPTCHA
Privacy - Terms

Start

Description

PySpark-Data Analytics-Employee-L1-VIR-Template

Step 2 Click on the launch assessment Environment

Speed Test Avg: 4.40Mbps

Live: 4.40Mbps

Time Remaining: 00:00:00

Final Submission

Question

Data Analytics-Employee-VIR-

Instructions

Introduction
This is a project-based assessment, in which you will be provided with a template code to work. You will be provided with a pre-configured Virtual Machine (VM) to develop the case study.

Launch
You can launch VM by clicking "Launch Assessment Environment". It will take around **5-10 mins to launch the VM.**

Configuration
Once launched, if you see **any configuration screen, skip it.**

Cloning
Once VM is up, it will take another **30 sec-1 min to clone your project template on desktop** of your VM. Please wait till then.

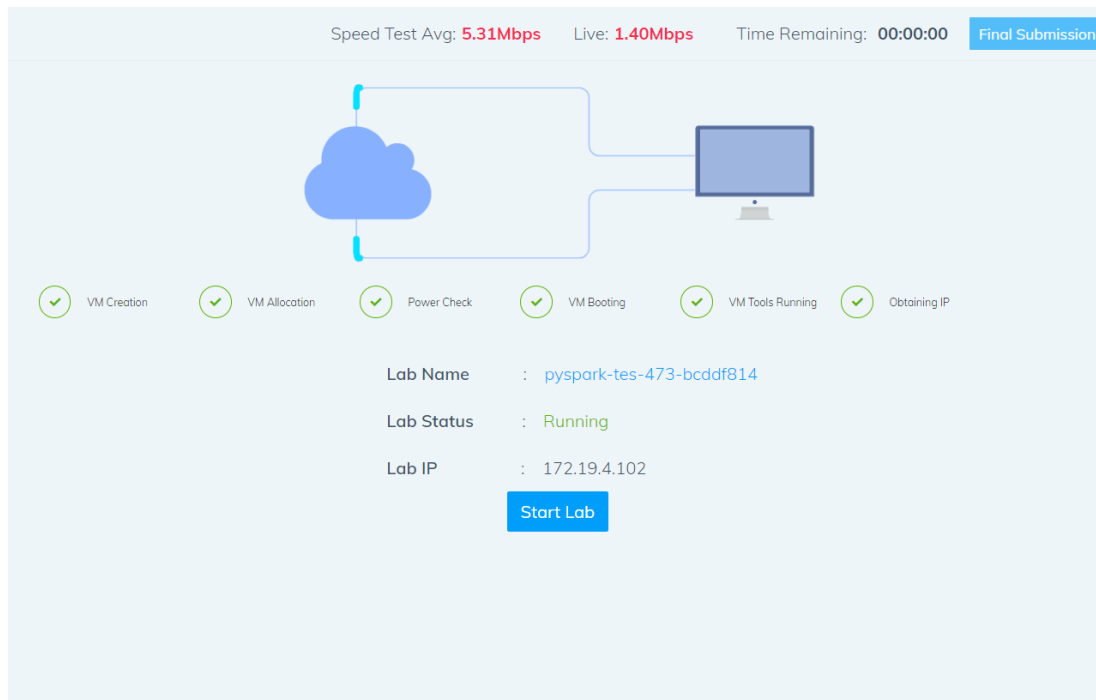
Document
The project will be cloned in a folder, named same as your email ID. The folder contains template code and case study document. You are required to open the case study document and **thoroughly go through it to understand project and mandatory process.**

Development
To develop use case all **IDEs, Database required are available in VM.** There is a README file on desktop containing any credentials you need.

Submission
Before you do final submission of your code there are 2 mandatory things you have to follow, else your evaluation result would be affected

1. RUN TEST CASE (AS MENTIONED IN DOCUMENT)
2. PUSH CODE TO GIT (AS MENTIONED IN DOCUMENT)

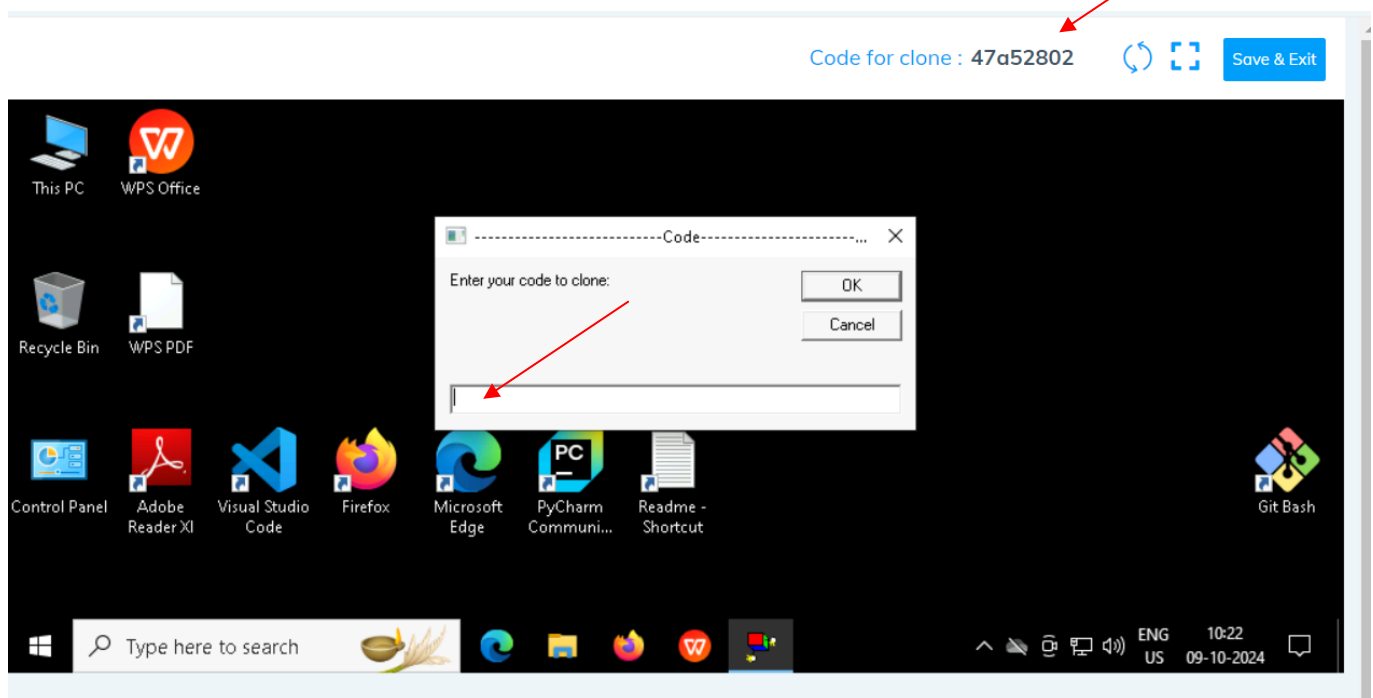
Launch Assessment Environment



Step 3 Click on the start lab button

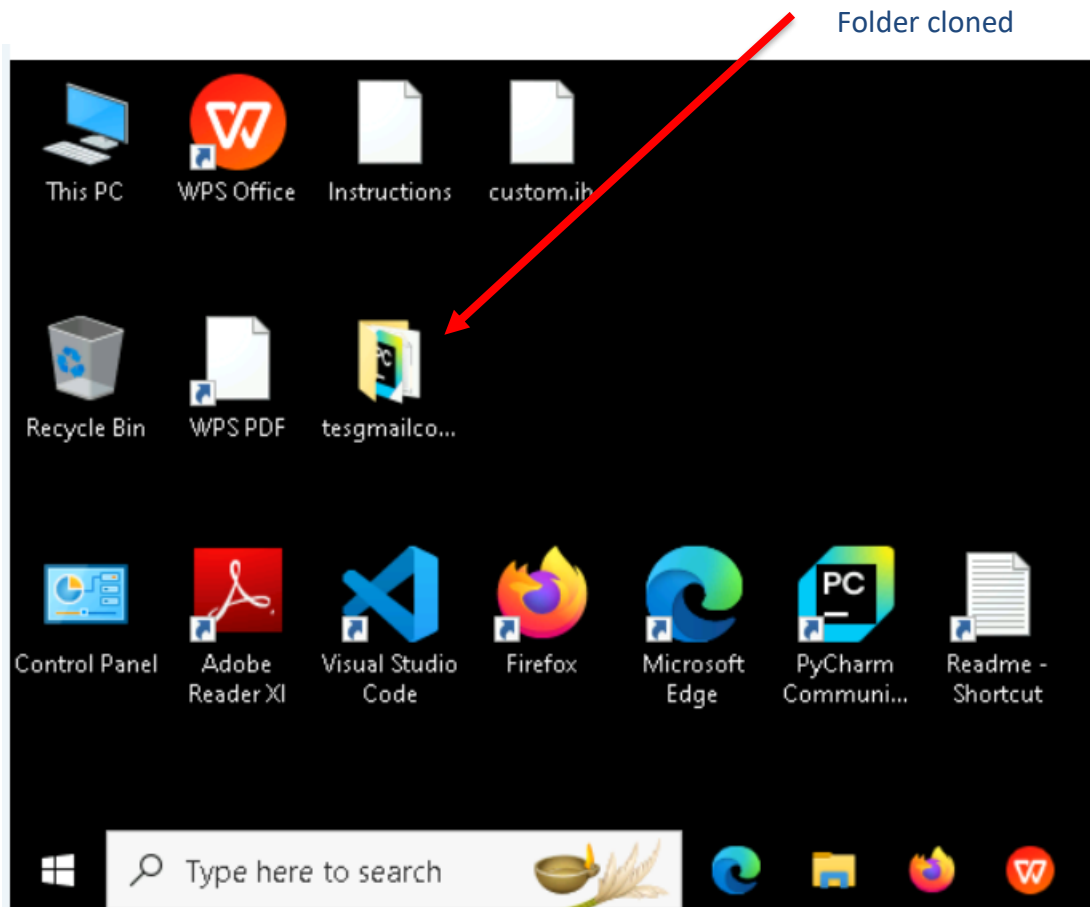
Step 4 you will get a window you need to type the code from that top corner

- You need to type the code in the window . It will take few minutes to start the window

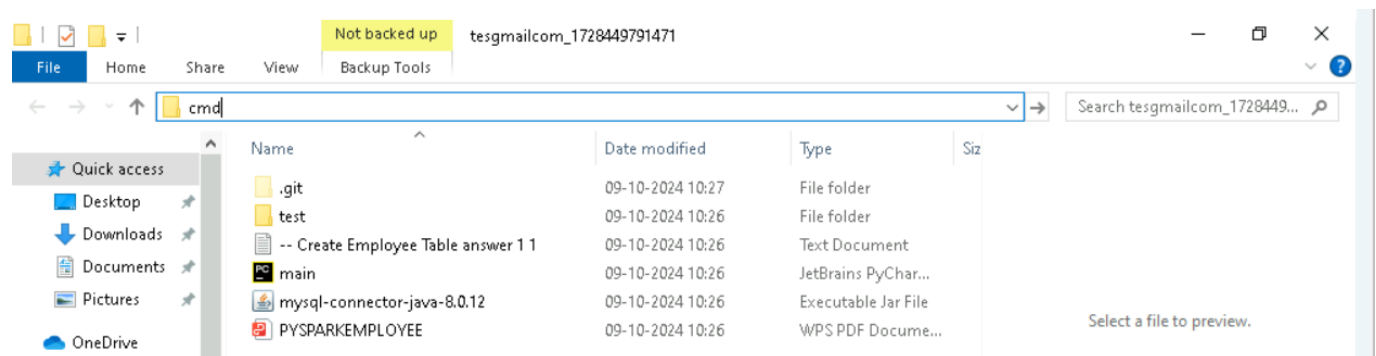


Click on ok

Step 5 after few seconds we can see that the your folder is cloned in the desktop .

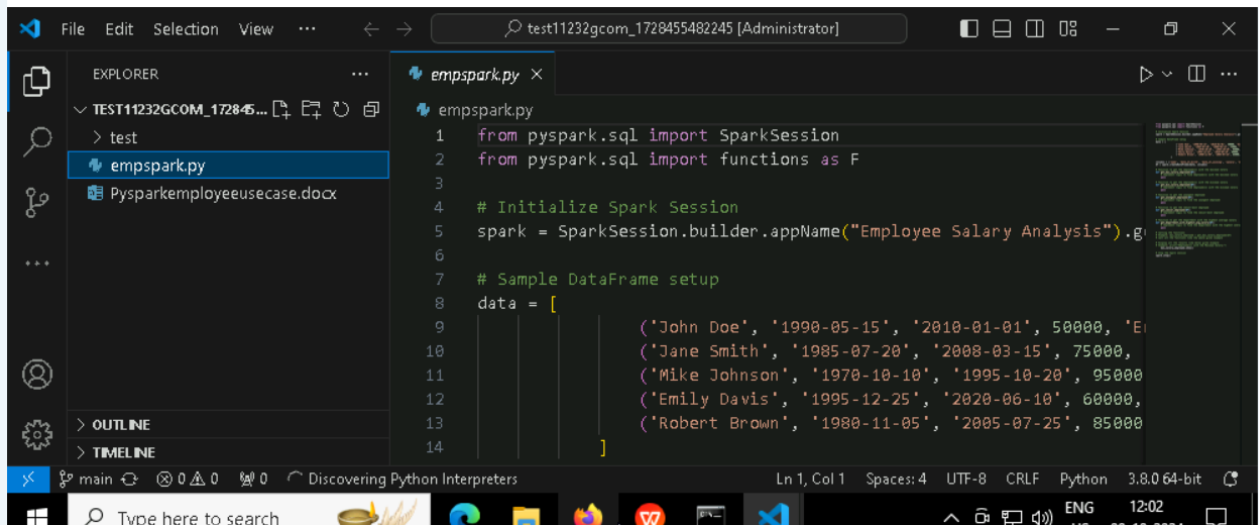


Step 6 go inside the folder type cmd in the top of the file explorer



- Type **code**. And hit enter you can see that workspace is opened in the visual code





- You can see that workspace is ready to code

Note Please only work with visual code not with any other IDE

- In the folder cloned you will have all the project files needed .

Problem Statement : **Employee Salary and Tenure Analysis**

Description : Use relevant methods operations to perform specified activities which are given in the instructions.

XYZ Corporation is conducting an internal analysis to understand the distribution of salaries across departments, identify the highest and lowest earners, and gain insights into the tenure and demographics of its employees. The company wants to make data-driven decisions regarding salary adjustments, promotions, and department-level resource allocation. They have collected data on employees' names, dates of birth, dates of joining, salaries, and departments. The company's data science team has been tasked with analyzing this data using PySpark to efficiently handle and process large datasets.

Objective:

The goal is to answer several key business questions based on the provided employee data:

1. **Identifying the Employee(s) with the Maximum Salary:**
XYZ Corp wants to recognize high-performing employees, particularly those who are the top earners in the company. The management needs to know who earns the highest salary to consider them for leadership roles or other recognitions.
2. **Identifying the Employee(s) with the Minimum Salary:**
The company is concerned about pay equity and wants to identify those who are at the lower end of the salary spectrum. Understanding who earns the least can help HR review compensation structures and ensure fair pay across roles.
3. **Identifying the Youngest Employee:**
The management is keen on understanding the demographics of its workforce. Identifying the youngest employee helps in creating mentoring programs where experienced employees can guide the younger ones.
4. **Identifying the Senior-most Employee:**
Seniority often correlates with experience and loyalty to the company. Identifying the

longest-serving employee is crucial for recognizing their contributions and possibly involving them in key decisions or as part of the company's legacy initiatives.

5. **Identifying the Department with the Highest Average Salary:**

To assess budget allocation, the management wants to identify which department has the highest average salary. This information could be used for future budgeting, talent acquisition, or re-evaluating departmental pay scales.

Questions Based on the Code:

1. **Who are the employees with the maximum salary, and how might their roles impact their earnings?**
2. **Who are the employees with the minimum salary, and are their roles potentially undervalued in the company?**
3. **Who is the youngest employee in the company, and what department do they belong to?**
4. **Who is the senior-most employee, and how has their experience possibly contributed to their department or the company as a whole?**
5. **Which department has the highest average salary, and what factors could contribute to this?**

Use the same dataset

(John Doe', '1990-05-15', '2010-01-01', 50000, 'Engineering'),
(Jane Smith', '1985-07-20', '2008-03-15', 75000, 'HR'),
(Mike Johnson', '1970-10-10', '1995-10-20', 95000, 'Finance'),
(Emily Davis', '1995-12-25', '2020-06-10', 60000, 'Engineering'),
(Robert Brown', '1980-11-05', '2005-07-25', 85000, 'Finance')

Execution Steps to Follow:

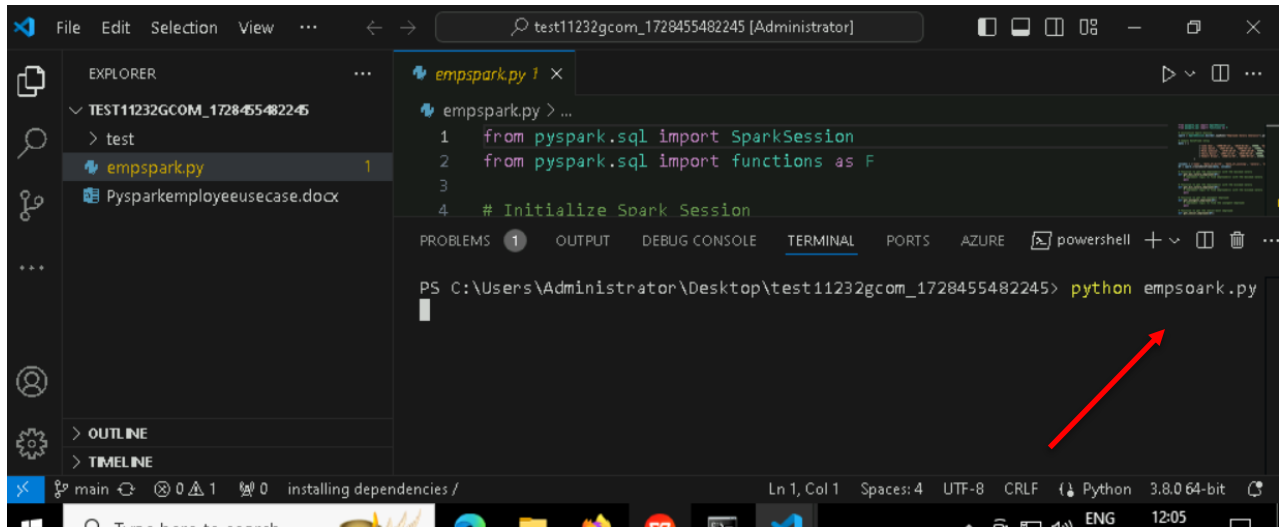
1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
3. This editor Auto Saves the code
4. If you want 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

7. To run Test cases:

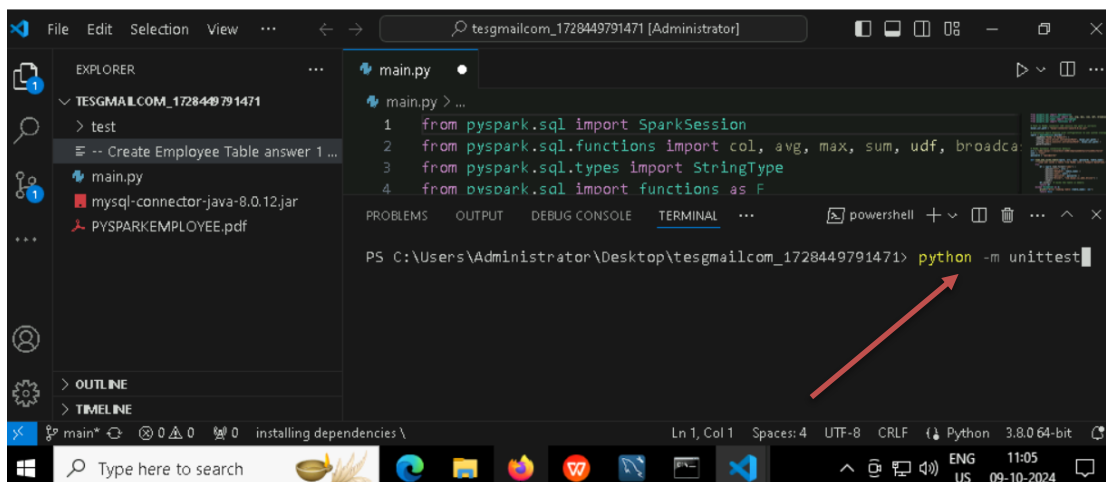
`python -m unittest`

Screen shot to run the program



To run the application

- Python empspark.py



To run the testcase

- Python -m unittest

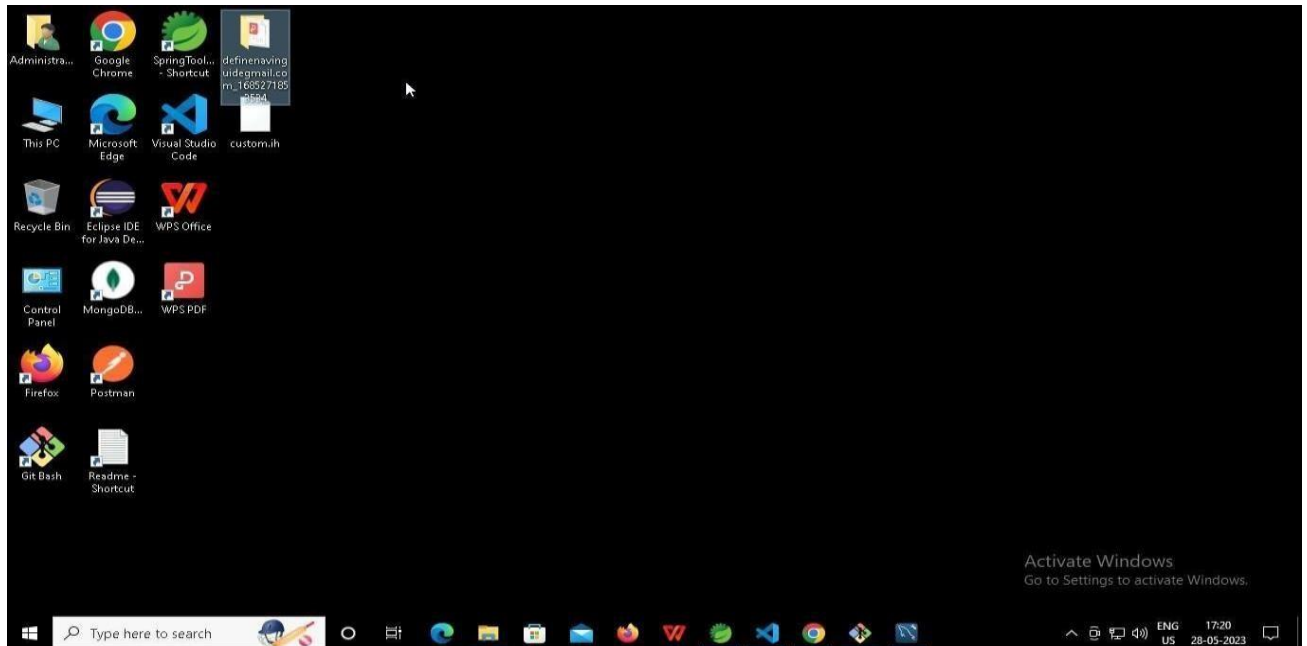
Screenshot to push the application to github

-----X-----

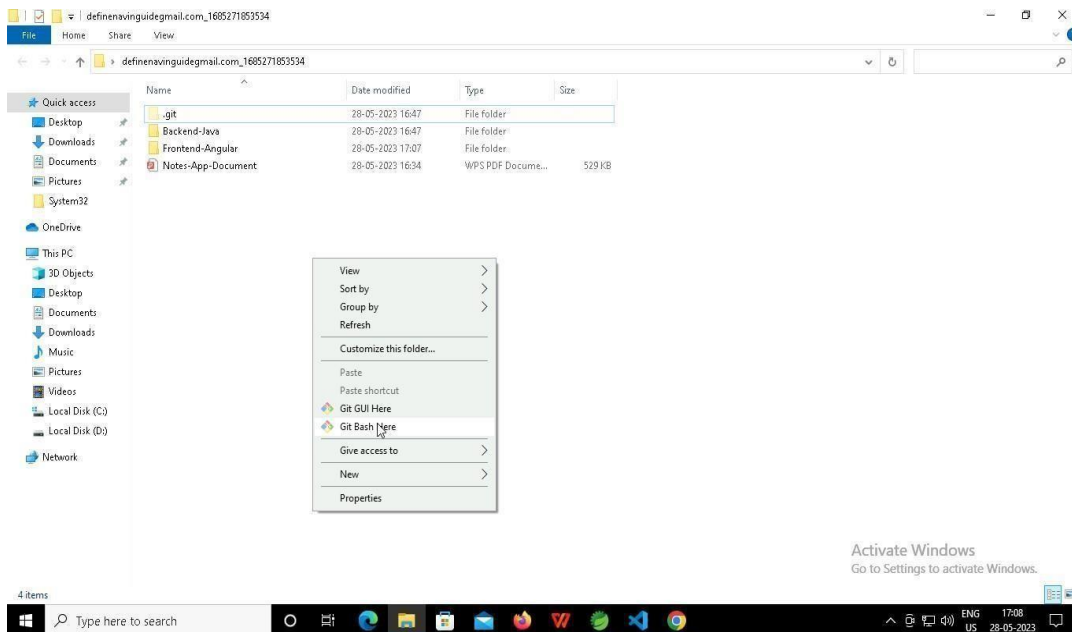
You can run test cases as many numbers of times and at any stage of Development, to check howmany test cases are passed/failed and accordingly refactor your code.

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

open the project folder available on desktop



a. Right click in folder and open Git Bash



b. In Git bash terminal, run following commands

c. `git status`


```
Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailto1728449791471
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    templateespark.py

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailto1728449791471 (main)
$
```

d. git add .

```
Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailto1728449791471 (main)
$ git add .
```

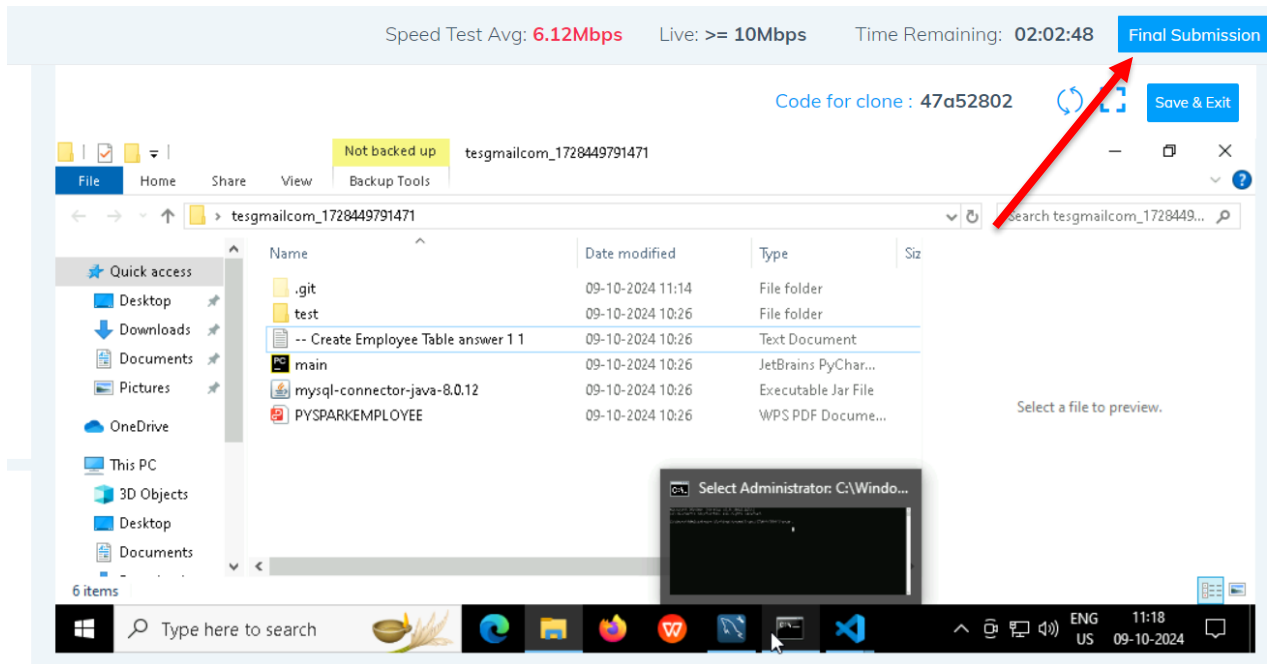
e. git commit -m "First commit"
(You can provide any message every time you commit)

```
Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailto1728449791471 (main)
$ git commit -m "first commit"
[main f97ce24] first commit
1 file changed, 91 deletions(-)
delete mode 100644 templateespark.py
```

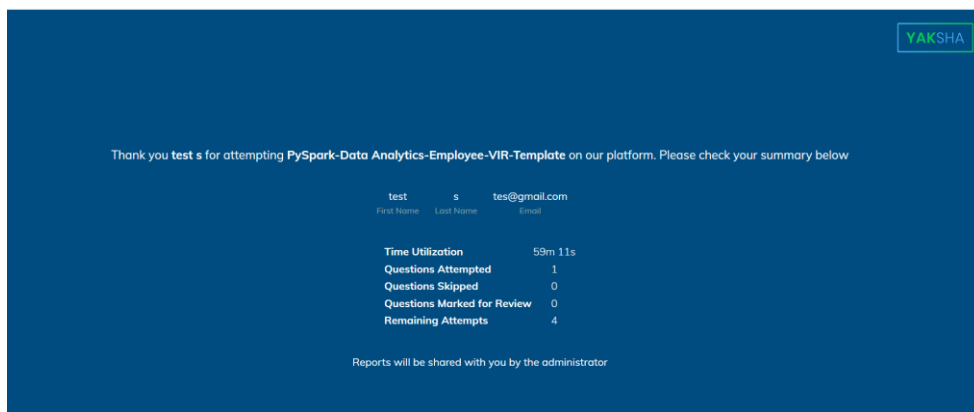
f. git push

```
Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailto1728449791471 (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 212 bytes | 212.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/IIHTDevelopers/tesgmailto1728449791471.git
a1c1905..f97ce24  main -> main
```

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-----