**Use case Titie : Predicting Purchase Behaviour with Logistic Regression**

You are working as a data scientist at an e-commerce company that wants to predict whether a customer will purchase a product based on their **age** and **estimated salary**. Your task is to build a **logistic regression model** that will:

1. Preprocess and split the data from a CSV file named purchase\_data.csv.
2. Train a logistic regression model on the training dataset.
3. Evaluate the model using accuracy and a classification report.
4. Predict whether a new customer will purchase a product based on their age and salary.

You are to **implement and complete** the following functions in the purchase.py module so that the provided **unit test file passes all tests**:

**Samples dataset field are {Age,EstimatedSalary,Purchased}**

**The Values are stored in the csv file name purchase\_data.csv**

**1. preprocess\_data(filename='purchase\_data.csv')**

* Load the dataset from the CSV file.
* Extract the Age and EstimatedSalary columns as features (X) and Purchased as the target (y).
* Split the data using train\_test\_split() with test\_size=0.2 and random\_state=42.
* Return X\_train, X\_test, y\_train, y\_test.

**2. train\_model(X\_train, y\_train)**

* Train a LogisticRegression model using the training data.
* Return the trained model.

**3. evaluate\_model(model, X\_test, y\_test)**

* Print the accuracy and the classification report using accuracy\_score and classification\_report.

**4. predict\_new(model, age, salary)**

* Use the trained model to predict whether a new user with given age and salary will purchase the product.
* Print an appropriate message such as:
  + "Person with Age=35 and Salary=70000 will purchase the product."
  + "Person with Age=25 and Salary=30000 will not purchase the product."

**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) then you need to use **CTRL+Shift+B** -command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
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 setup environment:

You can run the application without importing any packages

1. To launch application:

**python3 purchase.py** run Test cases:

**python3 -m unittest**

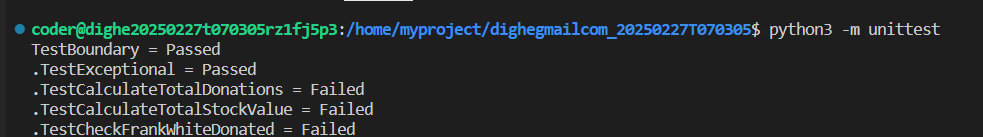
1. You need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

**Screen shot to run the program**



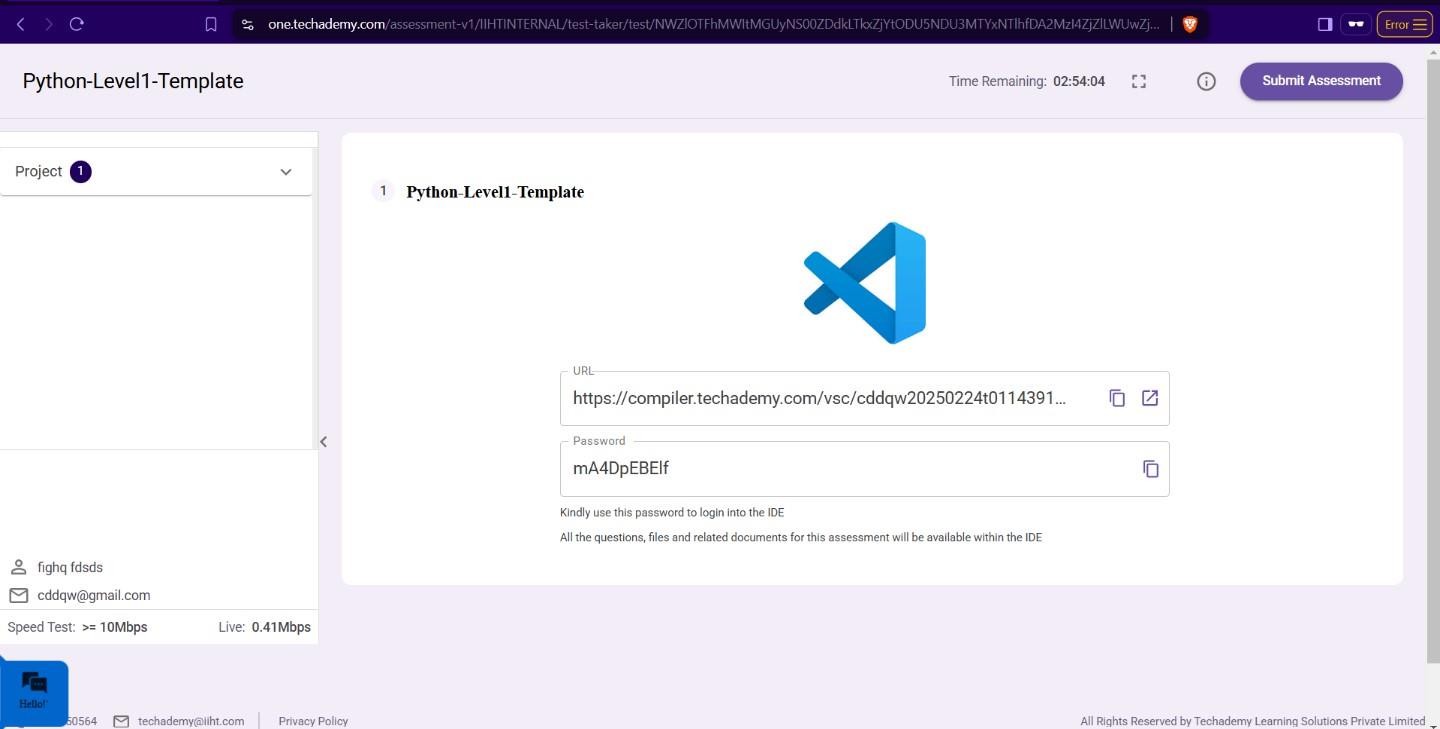
**To run the application**

**python3 purchase.py**



**To run the testcase**

* + **python3 -m unittest**

****

1. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.**