**Use Case:** **Predicting Diabetes Using Logistic Regression**

You are working as a data scientist at a healthcare analytics company. Your task is to build a logistic regression model to predict whether a patient is diabetic based on their Age, BMI, and Blood Pressure. This model will help healthcare professionals identify patients who may need further testing or lifestyle interventions.

Your responsibilities include:

* Loading and preprocessing data from a CSV file named diabetes\_data.csv.
* Training a logistic regression model using scikit-learn.
* Evaluating the model’s accuracy and generating a classification report.
* Making predictions for new patients based on their medical profile (Age, BMI, Blood Pressure).

You must ensure that your implementation passes the provided unit tests, which validate the core functionality of each part of the pipeline.

You are provided a CSV file named diabetes\_data.csv with the following structure:

Samples dataset in the diabetes\_data.csv

| Age | BMI | BloodPressure | Diabetic |
| --- | --- | --- | --- |
| 45 | 31.2 | 85 | 1 |
| 23 | 22.0 | 75 | 0 |
|  |  |  |  |

* Age: Age of the patient
* BMI: Body Mass Index
* BloodPressure: Measured blood pressure
* Diabetic: Target column (1 = Diabetic, 0 = Non-Diabetic)

You are to complete the following functions in the main.py module. The goal is to ensure all tests in test/test\_purchase.py pass successfully.

1. load\_and\_preprocess(filename='diabetes\_data.csv')

* Load the dataset using pandas.
* Select Age, BMI, and BloodPressure as feature columns (X).
* Use Diabetic as the target column (y).
* Split the data using train\_test\_split() with test\_size=0.2 and random\_state=1.
* Return: X\_train, X\_test, y\_train, y\_test.

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

* Train a LogisticRegression model on the training dataset.
* Return: The trained logistic regression model.

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

* Predict on the test data using the trained model.
* Print the following:
  + "Evaluation Results:"
  + "Accuracy: " (with 2 decimal places)
  + "Classification Report:" using classification\_report.

4. predict\_new(model, age, bmi, bp)

* Use the trained model to predict diabetes status for a new patient with given age, BMI, and blood pressure.
* Print one of the following messages:
  + "Patient (Age=45, BMI=32.5, BP=88) is diabetic."
  + "Patient (Age=25, BMI=22.1, BP**=72) is not diabetic."**

**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 main.py** run Test cases:

**python3 -m unittest**

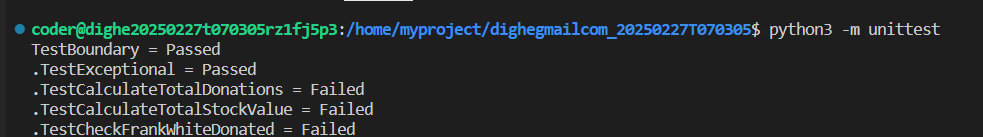
1. You need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

**Screen shot to run the program**



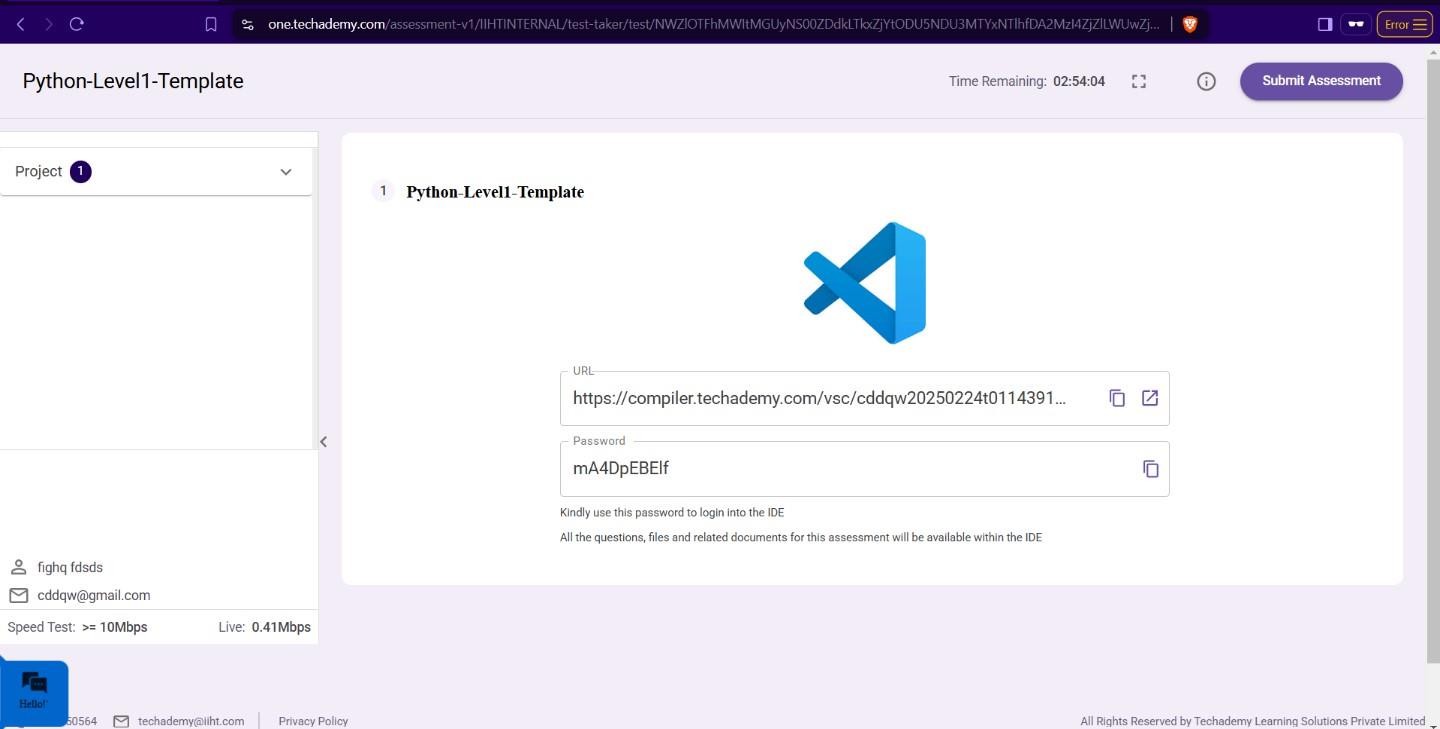
**To run the application**

**python3 main.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.**