This lab task is one of FIVE lab components that contribute to Assessment 1. The complete rubric can be found on the Assessment 1 page. Marks for this Lab task will be allocated towards Assessment 1 based on the completeness and correctness of the submission.
This assessment consists of practising simple Python data science and machine learning libraries. The assessment requires you to demonstrate your understanding and use of Python data science and machine learning libraries and tools for small-sized tasks.
Context for this Lab task:
The classic Olivetti faces dataset contains 400 grayscale 64 × 64–pixel images of faces. In this lab task, we will cluster these face images using K-Means and visualize the result to observe similarities in each cluster.

Assessment details
Your Jupyter notebook should include at least one markdown cell and at least one code cell for each task in the weekly tasks. The markdown cell briefly introduces the task and the code cell includes the Python code that you implement the task. Also, note that your Python program must contain meaningful comments and sensible variable and function names. Otherwise, up to half of your original mark can be deducted at the marker's discretion. Please submit your lab work before the submission deadline.

Please complete the following tasks:

Task 1 (3 points)
Data Preparation.

Step 1: Load the Olivetti face dataset.
lab 5 clustering.jpg

Hint: Load the dataset using the sklearn.datasets.fetch_olivetti_faces() function. Then answer the following questions:

Explain why each row of 'data' is a 1D vector of size 4096, and how many different labels this dataset has.
What else can you tell from this dataset by observing and visualizing the dataset?
Step 2: Reduce Dimensionality with PCA.

Reduce the training data's dimensionality using PCA. Please set the ratio of variance you wish to preserve as 99%. Then answer the following questions:

What's the dimensionality of the original training data?
What's the dimensionality of the compressed training data?
Task 2 (3 points)
Training a ML model.
Step 3: Train Test Split. Hint: Split the PCA-compressed dataset into training and test sets with train_test_split()
Step 4: Cluster the compressed training images using K-Means.
Step 5: Visualize the clusters, then answer the following questions:
What do you think of the clustering result?
What variations can you observe from this result?
What other clustering algorithms can you apply to solve this problem?
Hint: Expected clustering outputs may look like this:
lab 5 example result.jpg