**Lab Exercise 1 – MVC Architecture in Python**

Below is an example of the Model-View-Controller (MVC) pattern in Python, where each component is implemented in a separate file.

**model.py**

class Model:

def \_\_init\_\_(self):

self.\_data = None

def set\_data(self, data):

self.\_data = data

def get\_data(self):

return self.\_data

**view.py**

class View:

def show\_data(self, data):

print(f"Data from Model: {data}")

**controller.py**

class Controller:

def \_\_init\_\_(self, model, view):

self.\_model = model

self.\_view = view

def set\_data(self, data):

self.\_model.set\_data(data)

def update\_view(self):

data = self.\_model.get\_data()

self.\_view.show\_data(data)

**main.py**

from model import Model

from view import View

from controller import Controller

if \_\_name\_\_ == '\_\_main\_\_':

# Create model, view, and controller

model = Model()

view = View()

controller = Controller(model, view)

# Update the model and view through the controller

controller.set\_data("Hello, MVC Pattern!")

controller.update\_view()

With this structure, each component is in a separate file, promoting code organization and maintainability. The main script imports the necessary modules from the respective files to use the Model, View, and Controller classes.