

Class: **BSE-6 (A/B)**

BAHRIA UNIVERSITY (KARACHI CAMPUS)

ASSIGNMENT # 1 - SPRING 2023

**Cloud Computing** (**SEN-325**)

**[CLO 2]**

Max Marks: **5**

Course Instructor: **Engr. Muhammad Faisal**

# [The marks of this assignment may increase or decrease]

**Read Carefully:**

* The deadline for this assignment is *before* or *on* **Thursday, 5th May, 2023.**

**WARNINGS**:

* This is an individual assignment; you must implement it by yourself. Any form of plagiarism will result in receiving zero in the assignment.
* Late submission will not be accepted. Any assignment submitted after the cutoff time will receive zero.

This assignment is divided into two parts

1. Compare the following architecture patterns
   1. MVC
   2. MVVM
   3. MVP
2. Write the entire process of creating Web API using
   1. MVVM
   2. MVP

# To submit these answers the deadline is:

5th May, 2023

If you have any query, feel free to contact at: [**mfaisal.bukc@bahria.edu.pk**](mailto:mfaisal.bukc@bahria.edu.pk)

Page **1** of **1**

**Answer 1**

**MVC**

The Model-View-Controller (MVC) architecture comprises three interconnected components Model, View, and Controller that work together to create an application. The Model represents the data and business logic, the View represents the user interface, and the Controller facilitates communication between the Model and the View. By separating the application into distinct parts, MVC promotes a clear separation of concerns, making the application easier to maintain and evolve over time.

**MVVM**

The MVVM (Model-View-View Model) pattern builds on the MVC (Model-View-Controller) architecture and introduces a new component called the View Model. In MVVM, the View represents the user interface, the Model represents the data and business logic, and the View Model acts as a mediator between them. The View receives input from the user and displays data using the information provided by the View Model. MVVM is commonly used in modern online and mobile applications, where the user interface can be complex and interactive.

**MVP**

An architecture pattern called Model-View-Presenter (MVP) is like MVC in that it also has a Presenter component. By expressing the business logic and data in the Model and the user interface in the View, the Presenter serves as a bridge between the Model and the View. The Presenter takes input from the View, processes it using the logic of the Model, and then changes the View with the outcomes. The Presenter's ease of unit testing is one of the MVP's main advantages.

The main differences between these architecture patterns are the presence or absence of a View Model or Presenter component, and the way they handle communication between the Model and the View. The choice of architecture pattern depends on the specific requirements of the application and the preferences of the development team.

**Answer 2**

Creating a Web API involves designing and implementing the API endpoints, handling the business logic, and connecting to a database or other data source. Here are the general steps for creating a Web API using the MVVM and MVP patterns:

**Creating a Web API using MVVM:**

* **Design the API endpoints:** Define the API endpoints and their associated HTTP methods, such as GET, POST, PUT, and DELETE**.**
* **Implement the Model:** Define the data model and any necessary data access methods to interact with the data source.
* **Implement the View Model:** Create a View Model that handles the business logic of the API. The View Model should handle requests from the View and interact with the Model to retrieve or update data.
* **Implement the View:** The View in MVVM is typically a client application, such as a web or mobile app, that interacts with the API. The View should send HTTP requests to the API and display the data returned by the View Model.
* **Connect to the data source:** Implement a data access layer to connect to the data source, such as a database or external API. The View Model should interact with the data access layer to retrieve or update data.
* **Test and deploy the API:** Test the API endpoints using a tool such as Postman and deploy the API to a web server or cloud platform.

**Creating a Web API using MVP:**

* **Design the API endpoints:** Define the API endpoints and their associated HTTP methods, such as GET, POST, PUT, and DELETE.
* **Implement the Model:** Define the data model and any necessary data access methods to interact with the data source.
* **Implement the Presenter:** Create a Presenter that handles the business logic of the API. The Presenter should handle requests from the View and interact with the Model to retrieve or update data.
* **Implement the View:** The View in MVP is typically a client application, such as a web or mobile app, that interacts with the API. The View should send HTTP requests to the API and display the data returned by the Presenter.
* **Connect to the data source:** Implement a data access layer to connect to the data source, such as a database or external API. The Presenter should interact with the data access layer to retrieve or update data.
* **Test and deploy the API:** Test the API endpoints using a tool such as Postman and deploy the API to a web server or cloud platform.