**Topic 1 DQ 1**

What are the primary responsibilities of the Model, View, and Controller components in the MVC (Model-View-Controller) framework? Provide examples of how each component interacts within a .NET MVC application.

Hello everyone,

The Model-View-Controller (MVC) framework is a widely used architectural pattern in software development, particularly in web applications. It separates an application into three interconnected components: Model, View, and Controller, each with distinct responsibilities.

### Responsibilities of MVC Components

1. **Model**:  
   The Model represents the data and business logic of the application. It is responsible for retrieving data from the database, processing it, and returning it to the Controller. The Model also notifies the View of any changes in data, allowing for dynamic updates.

**Example**: In a .NET MVC application, a Model could be a class representing a "Product" entity, which includes properties like Id, Name, and Price. It would also contain methods for interacting with the database, such as GetProducts() to retrieve a list of products.

1. **View**:  
   The View is responsible for rendering the user interface and displaying data to the user. It presents the information provided by the Model in a format that is easy for users to understand. The View is generally composed of HTML, CSS, and JavaScript.

**Example**: In a .NET MVC application, a View could be a Razor file (e.g., Index.cshtml) that displays a list of products. It binds to the Model to display each product's details, such as its name and price.

1. **Controller**:  
   The Controller acts as an intermediary between the Model and the View. It handles user input, processes it (often by interacting with the Model), and determines which View to display. The Controller responds to user actions, such as button clicks or form submissions.

**Example**: In a .NET MVC application, a Controller might have an action method like Index(), which retrieves a list of products from the Model and passes that data to the View for rendering. The method might look something like this:

csharp

public ActionResult Index(){ var products = \_productModel.GetProducts(); return View(products);}

### Interaction Between Components

The interaction between the Model, View, and Controller in a .NET MVC application can be summarized as follows:

* The **Controller** receives user input and calls methods on the **Model** to retrieve or update data.
* The **Model** processes this data and returns it to the **Controller**.
* The **Controller** then selects the appropriate **View** and passes the data to it.
* Finally, the **View** renders the data and presents it to the user.

### References

1. Freeman, A., & Robson, E. (2014). Head First C#. O'Reilly Media.
2. Steinberg, D. (2016). Spring MVC: A Beginner’s Guide. McGraw-Hill Education.
3. Microsoft. (n.d.). ASP.NET MVC Overview. Retrieved from <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>