* Code first approach - In this approach, the domain classes is defined first and then the entity framework generates the database schema based on those classes.
* Database first approach – in this approach the existing database is started with first and the entity framework generates the domain classes based on that database schema.
* Model first approach – in this approach the model is created first using the entity framework designer and then generate the database schema from that model.
  1. This is used to create dynamic web pages by using C# code within the HTML markup.
  2. The model represents the business logic and the data of the system.
  3. The controller acts as an intermediary between the view and model, it also facilitates communication between the model and view.
  4. Bootstrap is designed to develop responsive websites and streamline the process of developing websites, and includes the CSS and JavaScript components.

1. Naming patterns such as ID and [entityname]ID are used to indicate primary keys. The preferrable naming pattern is to include the entity name along with the ID, such as “ProductID” to make it clearer.

* Constructor approach: You can utilize the derived DbContext class and specify the database connection string.
* Configuring connection string externally: config the connection string externally, either in the web.config file or through dependency injection.
* DbContext constructor with no parameters: Entity Framework will look for a connection string with the same name as your context class in the configuration file.
* This can lead to loss of production data.
* It disrupts the service for users during the database migration process.

1. Fluent API: used to develop fluent and code-based approach for configuring model entities, relationships.

Data Annotations: Attributes are directly applied to model classes to specify configuration details.

1. The exact version of MVC framework used in the web application is version 5.0.17 and it can be found in the in the ContosoUniversity.csproj