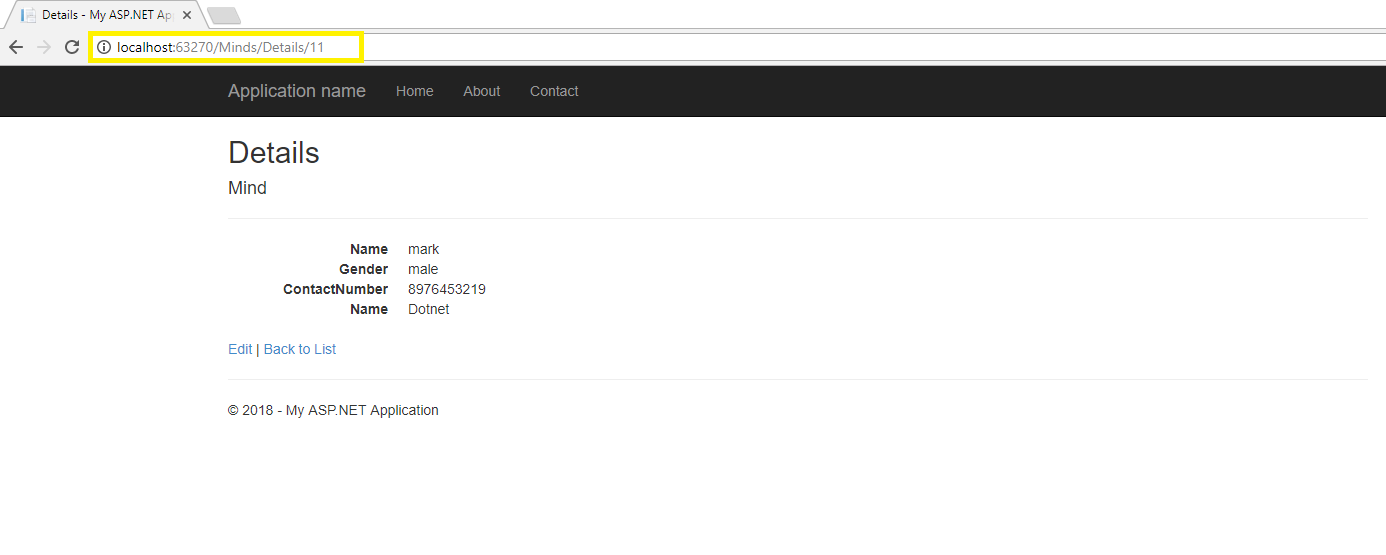
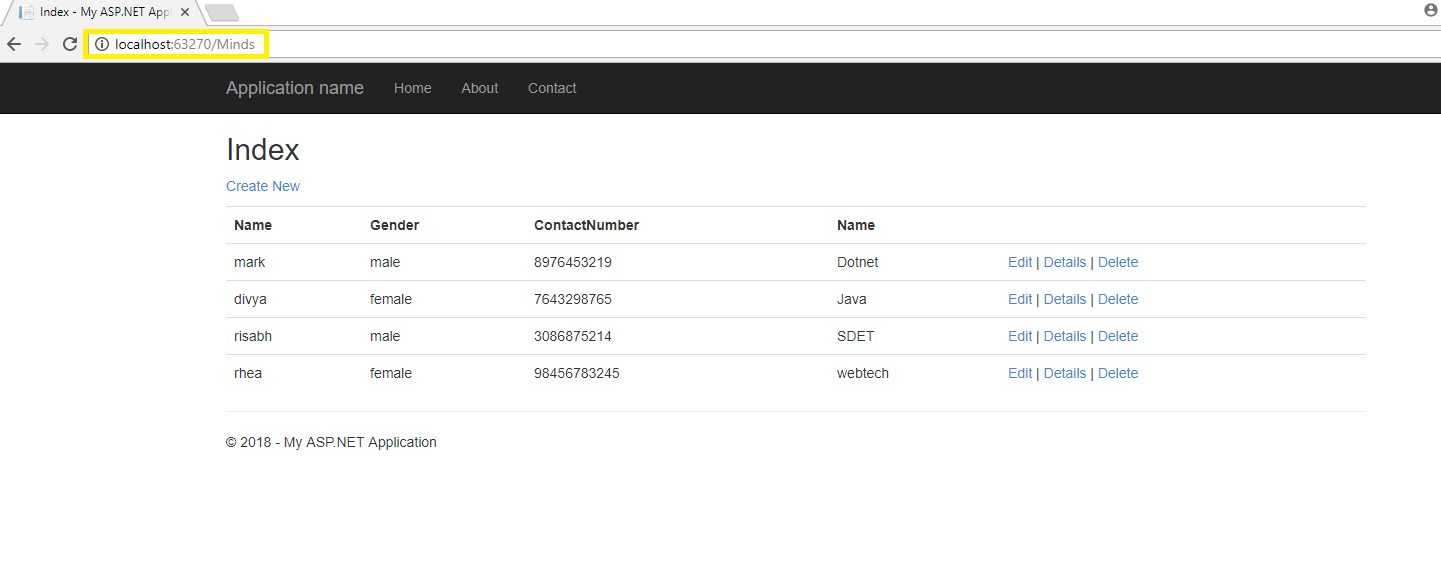
**Capabilities Covered:**

* Implement MVC framework using .Net MVC 5
* Use Entity framework to implement persistence layer of an application

**Entity Framework Assignment**

**Problem description:** Create a Web Application Called CampusMindTrackAllocation using MVC-5 and Entity Framework-6.0 where a user can perform CRUD operations.

Output should be like below:



**Assignment 1: Database First Approach**

**pe**

* PreRequisite:

Create a Table Minds which have following columns:

-MID(primary key)

-Name

-Gender

-ContactNumber

-TrackId(foreign key)

Create a Table Tracks which have following columns:

-TrackId(primary key)

-TrackName

-RoomAllocated

-LeadName

* Add ADO.net Entity Data Model named “TrackAllocation”under Model Folder. Select above two tables while adding it.

|  |
| --- |
|  |

* Entity Data Model Wizard in Visual Studio (2012/2015/2017) opens with four options to select from: EF Designer from database for the database-first approach, Empty EF Designer model for the model-first approach, Empty Code First model and Code First from database for Code-First approach. We are using database-first approach here, so select the EF Designer from database option and click Next.
* Then select your database and then tables as below:

|  |
| --- |
|  |

**Note:**

* **Pluralize or singularize generated object names** checkbox singularizes an entityset name, if the table name in the database is plural. Similarly, relationships between the models will be pluralized if the table has a one-to-many or a many-to-many relationship with other tables.
* The second checkbox,**Include foreign key columns in the model**, includes a foreign key property explicitly to represent the foreign key.
* The third checkbox, **Import selected stored procedures and functions into entity model**, automatically creates Function Imports for the stored procedures and functions. You don't need to import functions manually, like it was necessary prior to Entity Framework 6.0.
* After clicking on Finish, a TrackAllocation.edmx file will be generated.
* Open EDM(Entity Data Model) designer by double clicking on TrackAllocation.edmx. This displays all the entities for the selected tables and the relationships between them, as shown below:

|  |
| --- |
|  |

* EDM adds the provider and connection string settings in the App.config file.

|  |
| --- |
| **Note:**  Context & Entity Classes:  Every Entity Data Model generates a context and an entity class for each database table. Expand the .edmx file in the solution explorer and open two important files, <EDM Name>.Context.tt and <EDM Name>.tt.  Example:     * **TrackAllocation.Context.tt:** This template file generates a context class whenever you change the Entity Data Model (.edmx file). You can see the context class file by expanding TrackAllocation.Context.tt. The context class resides in the <EDM Name>.context.cs file. The default context class name is <DB Name>Entities. * **TrackAllocation.tt:** School.tt is a T4 template file that generates entity classes for each DB table. Entity classes are POCO (Plain Old CLR Object) classes. |

* After that build your solution and then proceed as below adding a controller.
* Create a MindsController where all crud operation being followed.(can use Scaffolding)
* While creating controller choose “MVC 5 controller with views, using Entity Framework”
* Then select Modelclass Minds from dropdown and Specified datacontext class from dropdown and then click on add button, refer the below sample screenshot:

|  |
| --- |
|  |

* If you get any error after above step then u would'nt have builded your solution after edmx diagram generation.
* If everything is fine then as soon as you click add button all codes for crud operation will get generated
* Check the required output is coming or not.

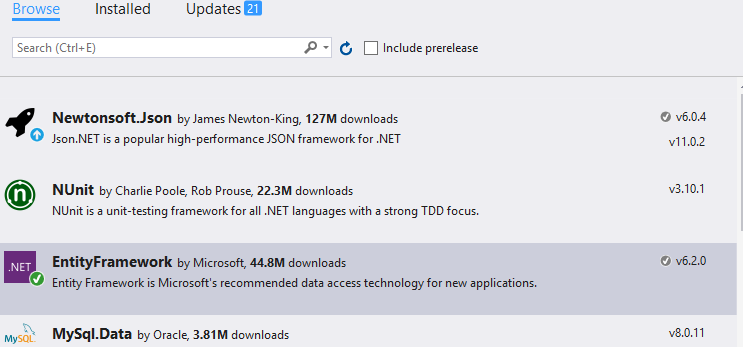
**KeyFocus Area of Database First Approach:**

* EDM
* How to use database first approach
* You can open an EDM designer in XML view where you can see all the three parts of the EDM: Conceptual schema (CSDL), Storage schema (SSDL) and mapping schema (MSL), together in XML view.

**Assignment 2: Code First Approach**

**pe**

* Add ‘EntityFramework’ reference. For that right click on project->Manage Nuget Package-Browse->install EntityFramework



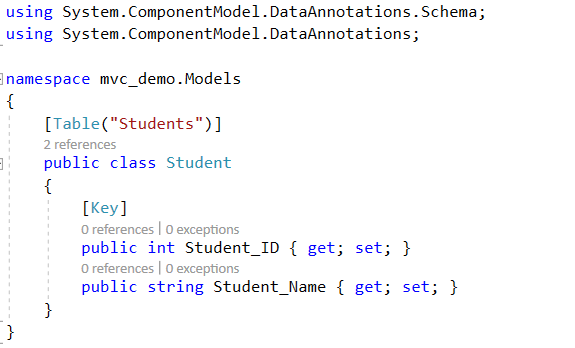
After installing you can see in References folder the below references will be added :



* Configure the web.config with proper connectionstring:

Connectionname, database server name, database name (MindDB – with this name it will be created in the corresponding database server), Integrated security, proviodername.

* After that, create two models named: Minds, Tracks with the corresponding fields mentioned in the prerequisite.
* Now configure the fields of the model class with primary key, foreign key property and the table name as “tbMinds” and “tbTrackDetails”. As example below:



In the above example, Student is a model class with primary key Student\_ID and database table name “Students”.

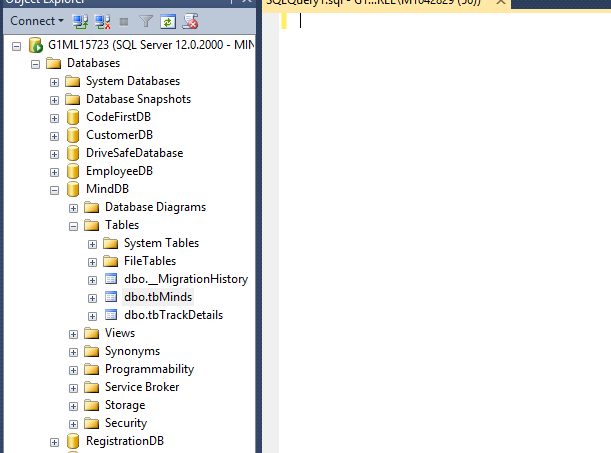
* Create a DbContext class named as “MindDBContext”.

|  |
| --- |
| public class MIndDBContext : DbContext  {  public MIndDBContext() : base("name=<ConnectionStringName>")  {  }  public DbSet<Model Class 1> Minds { get; set; }  public DbSet<Model Class 2> Tracks { get; set;}  } |

* Now create database with Code based migration :

1. Enable-Migrations
2. Add-Migration ‘<any meaningfulname>’
3. Update-database –verbose

* Now check the database whether the database has been created or not:



* Now add a controller for CRUD operation and check required output is coming or not.

**KeyFocus Area of Code First Approach:**

* DbContext, Dbset
* Need to know two types of migrations: Automated Migration and code based migration and differences between them.
* Configuration of connectionstring