Ministry of Education of the Republic of Moldova
Technical University of Moldova
Department of Applied Informatics

Report

Laboratory Work Nr.3
on Web Technologies

Do Some CRUD

Performed by Metei Vasile
Supervised by Plugaru Tudor

May 8, 2018

Goals:

- Understand what is an ORM and how to use it;
- Get more familiar with MVC pattern;
- Understand BusinessLogic pattern;

Main requirements:

- Define one model;
- Implement basic Create, Read, Update, Delete operations for the defined model;

Bonus points:

- 2 or more models are defined and there are some relations between them (FK, MtM...);(2pt)
- Define an ImageField or something related in a model. Display the uploaded image in detail view;(1pt)
- On the list view, implement some basic filtering (search field, etc);(1pt)

Laboratory Work Analysis

Object-relational mapping (ORM, O/RM, and O/R mapping tool) in computer science is a programming technique for converting data between incompatible type systems using object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language. There are both free and commercial packages available that perform object-relational mapping, although some programmers opt to construct their own ORM tools.[1]

In order to simplify my work while manipulating the data from database I have used Entity Framework. It is an Object Relational Mapper (ORM) which is a type of tool that simplifies mapping between objects in your software to the tables and columns of a relational database. With the Entity Framework, developers can work at a higher level of abstraction when they deal with data, and can create and maintain data-oriented applications with less code compared with traditional applications.[2]

There are three different approaches you can use while developing your application using Entity Framework:

- Database-First
- Code-First
- Model-First

In the database-first development approach, you generate the context and entities for the existing database using EDM wizard integrated in Visual Studio or executing EF commands.[2]

Use Code-First approach when you do not have an existing database for your application. In this approach, you start writing your entities (domain classes) and context class first and then create the database from these classes using migration commands.[2]

In the model-first approach, you create entities, relationships, and inheritance hierarchies directly on the visual designer integrated in Visual Studio and then generate entities, the context class, and the database script from your visual model.[2]

By splitting the solution into 3 logical layers:

- Presentation
- Business Logic
- Data Access

We construct better our development efforts and increase maintainability.

With regards to an application, the domain refers to the business entities that are involved. For example, in a school application, the domain objects include Teacher, Classroom and Discipline. Domain objects can contain domain objects; a simple example being School containing Teachers. Or Teacher having more than one Disciplines.

The model of an application refers simply as to the way in which domain objects are described. For example, for the domain object Student, the model object will consist of Name, Class, Disciplines and Grades.

Note that for every domain object, we should have at least one model object. The domain objects reside in the business logic layer. The model objects act as a sort of bridge between the business logic layer and the actual data access layer.

Laboratory Work Implementation

In this laboratory work I have continued the development of the previous assignments. Here I connected a database and manipulated its data or in other words performed some CRUD. In order to make it easier I used Entity Framework while working withe the database. From all the approaches available in Entity Framework I used Code-First. I find it more useful and easier. Using this approach means that I first created my models and context class and based on them my tables were generated using migration commands.

There are 2 models defined: Product and Brand. There is a relation between them represented by the Foreign Key Brand_Id from Products Table which is tied with Id field from Brands Table.

Beside the fields discussed above there are some more in Products table, such as: Name, Price and Picture. The picture field is used for storing the path of the product's image. It is not a good idea to store the picture is the database that's why I am storing its path. The picture is displayed in a view with the corresponding product.

After you click ”View” button, all the products from the database will be shown. After they show up you can filter them by the price or search by name. In the Figure 1 this features can be observed.

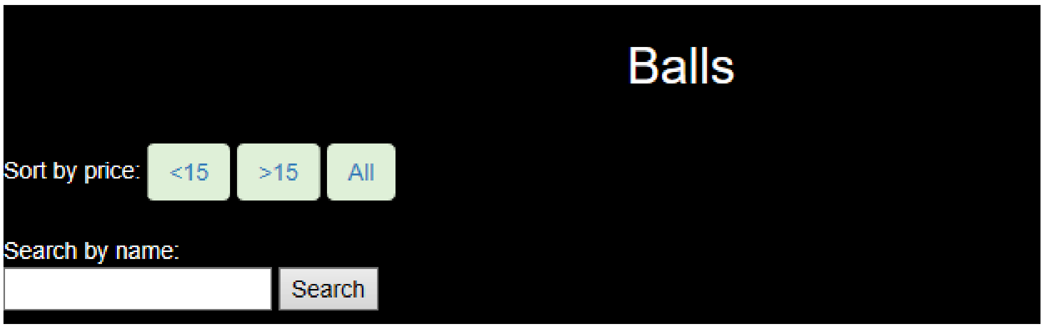
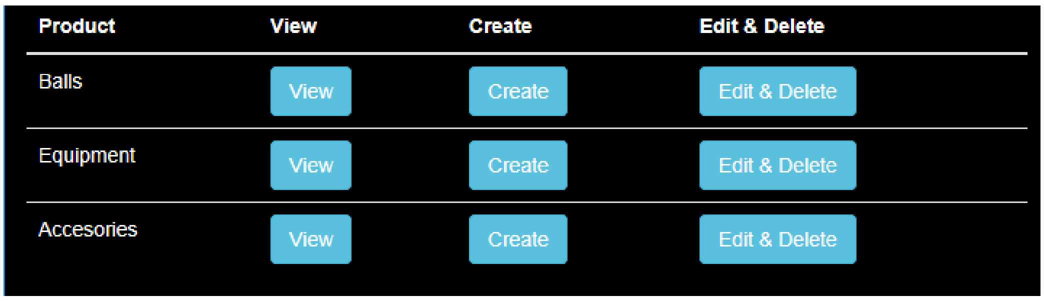


Figure 1: Search and filter options

There are also implemented CRUD operations. Because this laboratory work doesn't require authentication everyone can perform Create, Edit or Delete option. But in the future these operations will be available only for admins. In order to perform one of the above mentioned operations click the corresponding button.



Product	View	Create	Edit & Delete
Balls	View	Create	Edit & Delete
Equipment	View	Create	Edit & Delete
Accesories	View	Create	Edit & Delete

Figure 2: CRUD

If you choose to Create a product a form view with 4 fields will show up. There you will have to enter the corresponding information about the product. In order to add its image, you have to download the image and put it in the Content folder which is in eUseControl.Web project. And then paste its path in corresponding field. The Brand field will display a dropdown list of available brands, you have to choose one of them.

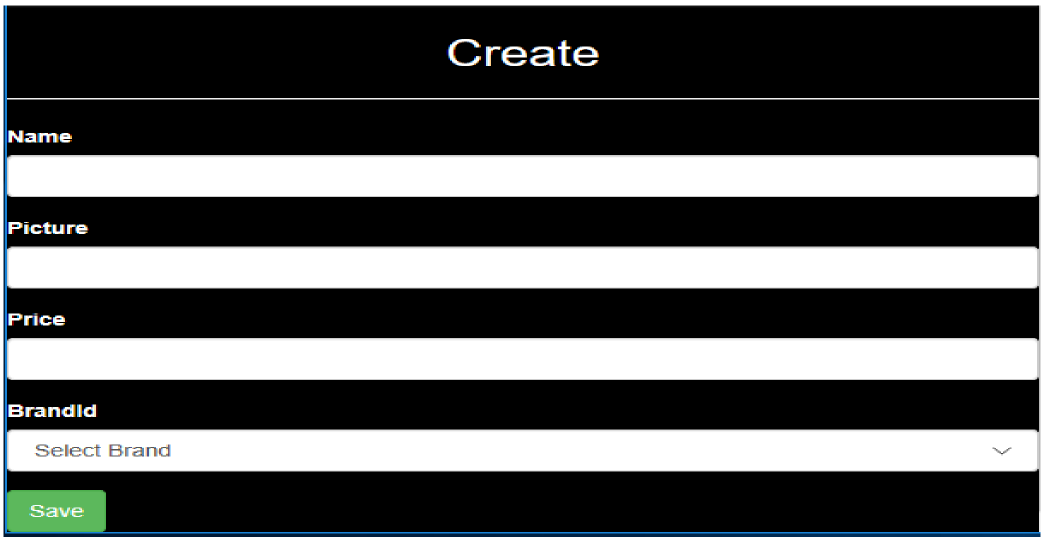
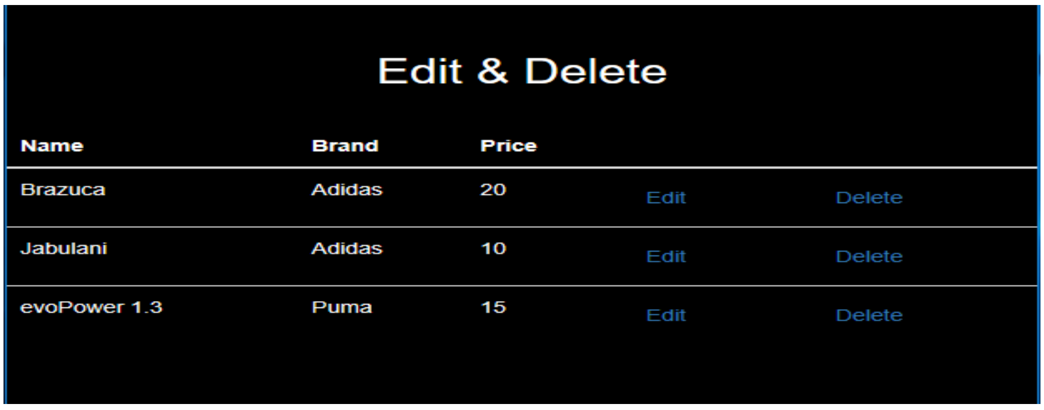


Figure 3: Create View

Beside Create option there are 2 more. If you choose to perform one of those then a table with all the products will show up. On the right side of every Product will be two options Edit and Delete. Figure 4 represents how this table looks. Edit option will render a form view similar to the create form view. After you finish editing the product the information from database will update. Delete option will delete the product from the database.



Edit & Delete

Name	Brand	Price		
Brazuca	Adidas	20	Edit	Delete
Jabulani	Adidas	10	Edit	Delete
evoPower 1.3	Puma	15	Edit	Delete

Figure 4: Edit and Delete Table

Conclusion:

In this laboratory work I discovered what an ORM is and how it should be used. I used Entity Framework as an ORM tool. From all the approaches given by this framework I used Code-First. It made my work with database much more useful and easier.

I learned what a Foreign Key is and why we need it. Because it was not defined at the first I met some problems while creating it.

The idea of storing an image in database is not good. That's why I discovered how a picture should be stored. So the easier way is to store its path, this is what I done.

This laboratory work gave me the possibility to have a better understanding about how a Web Site is created and how it interacts with a database.

References

[1] URL: `https://en.wikipedia.org/wiki/Object-relational_mapping`. (accessed: 08.05.2018).

[2] URL: `http://www.entityframeworktutorial.net/what-is-entityframework.aspx`. (accessed: 08.05.2018).