|  |  |
| --- | --- |
| Zwv6LprU | **CYPRUS INTERNATIONAL UNIVERSITY**  **FACULTY OF ENGINEERING**  **COMPUTER ENGINEERING** |

CMPE300

TRAINING REPORT

21/12/2019

STUDENT NAME: YUNUS

SURNAME: KAHRAMAN

NUMBER: 20151305

TRAINING PLACE NAME: TİPOSOFT BİLİŞİM HİZMETLERİ LTD.ŞTİ

ADDRESS: FULYA MAH. ORTAKLAR CAD. BAHÇELER SOK. EFEHAN NO: 20 KAT 3 D: 15 ŞİŞLİ/İSTANBUL

TRAINING DATE: Starting: 20/06/2019 Completion: 02/10/2019

PHONE NO: (0212) 212 53 34

WEB ADDRESS: www.tiposoft.com

Table of Contents

1. INTRODUCTION 6

2. INFORMATION ABOUT THE COMPANY 7

2.1. Aim and Establishment of the Company 7

2.1.1. Position of Company in the World 8

2.1.2. Types of services and Products 9

2.1.3. Certificates 11

2.2. Departments and Personnel of the Company 12

2.2.1. Personnel Information 12

2.2.2. Departments 14

3. WORK EXPERIENCE 15

3.1. Problem Definition 15

3.1.1. Integrated Development Environment 16

3.1.2. Programming Language 16

3.1.3. Framework 16

3.1.4. Design Pattern 16

3.1.5. Database 17

3.2. Work Done 18

3.2.1. Adaptation Process to Company 18

3.2.2. Create Static Website 20

3.2.3. Create Dynamic Website 20

3.3. Limitations and Experience Gained 27

3.3.1. Limited Time 27

3.3.2. What was Lacking 27

3.3.3. How Can Gained more Experiences? 27

4. CONCLUSION 28

5. REFERENCES 29

6. APPENDIX 30

List of Figures

Figure 2.1: Logo of the Company 7

Figure 2.2: Position of the Company 8

Figure 2.3: Tipo Admin 9

Figure 2.4: Tipo Checker 10

Figure 2.5: Tipo 3 10

Figure 2.6: Tipo 7 11

Figure 2.7: Tipo Mini 11

Figure 2.8: Certificates 11

Figure 2.9: Organizational Chart 12

Figure 6.1: Database Diagram 30

Figure 6.2: Entity Frameworks 30

Figure 6.3: RenderBody Error 31

Figure 6.4: Page of Category Section 31

Figure 6.5: View of New Category Section 32

Figure 6.6: View of New Customer Section 32

Figure 6.7: View of New Product Section 33

Figure 6.8: Syntax 34

Figure 6.9: Paging 34

Figure 6.10: Implementing Alert 34

Figure 6.11: Delete Alert 34

List of Listings

Listing 3.1: Giving Link to Menus 22

Listing 3.2: Fetching Values 23

Listing 3.3: Removing Category 23

Listing 3.4: Removing Product 24

Listing 3.5: Removing Customer 24

Listing 3.6: Category Information Updating 24

Listing 3.7: Product Information Updating 25

Listing 3.8: Product Validation Controlling 25

Listing 3.9: Implementing Paging Code Sample 25

Listing 3.10: Implementing Paging another Code Sample 25

Listing 3.11: Implementing Alert 26

Listing 3.12: Deleting with Alert 26

Listing 3.13: Implementing Search Panel 26

Listing 6.1: Defining Model Entity 31

Listing 6.2: Index of Category Controller Codes 31

Listing 6.3: Codes of New Category Section 32

Listing 6.4: Codes of New Customer Section 33

Listing 6.5: Usage of LINQ 34

List of Table

Table 2.1: Personnel Information 12

1. INTRODUCTION

This report was written to convey the experiences acquired during summer training. The theoretical knowledge acquired in the university has been provided with the opportunity to practice practically. Overall, it was a really good experience for me to encode the front end and the back end with the .Net framework. During this period, an analysis of the company's unique working conditions was observed.

The aim of the training is to learn professional engineering life. My personal knowledge increases with the application of the learned information. Thus, business life was experienced before graduation. I had the opportunity to observe teamwork and work sharing. This report consists of 6 main sections.

Information, aim, and establishment of the company, staff, and departments were given under the title "Information about the Company".

The "Work Experience" section describes all the information regarding responsibilities, exactly what needs to be done to solve tasks, the problem, the missing one, how to get a better job, how to gain more experience. Worked part, job description of machines, computer programs used, computer programs used will be read. Each task will be explained in separate subsections.

The "Conclusion" section of this Summer Training report includes ideas, thoughts, and observations, comments about the company and work experience.

The "References" section contains all references listed in the IEEE format and all references cited in the text.

The "Appendix" section includes all codes, figures, and tables shown figure numbers and captions.

1. INFORMATION ABOUT THE COMPANY

Tiposoft is a software company that provides betting service 24 hours a day, 7 days a week. Tiposoft, the betting industry in 2011, was targeted to become a leader. It attracted the attention of its competitors by making great steps in a short time. Tıiposoft reached 10 countries on 3 different continents. It continues to provide its customers with a reliable, high-quality, enthusiastic betting experience on the road that continues eagerly on the first day without slowing down. And by renewing itself day by day, it gets the appreciation of its customers. In the following sections, Aim and Establishment of Tiposoft, Department and Personal of the Tiposoft, Work Experiences will be discussed.



**Figure 2.1: Logo of the Company**

* 1. Aim and Establishment of the Company

Tiposoft is a virtual dealer that has been broadcasting matches from many betting leagues on both the website and the dealers since 2011. The betting program Tiposoft Bilişim Hizmetleri is produced only to serve abroad. There is no activity within the borders of Turkey and North Cyprus. Tiposoft’s timeline of information is shown below.

**Tiposof Founded - 2011:**

In 2011, Tiposoft released its first program to the market with the vision and goal of becoming a permanent institution.

**Introduction to the Market – 2012:**

In 2012, Tiposoft started to gain a place in the business sector quickly after making entrance with a young and dynamic staff.

**Customer Satisfaction – 2013:**

In 2013, after receiving a satisfying amount of demand from the market, Tiposoft began taking steps towards bringing its existing program to a wider audience.

**System Being Renovated -2014:**

In 2014, Tiposoft considered a number of its customers reviews and suggestions and went into a deep-rooted renovation to offer their customers the best service. The latest product that was presented to the market included renewed software, 9 sports branches and 32+ bet types.

**Trusted Brand -2015:**

Tiposoft gained more power with its experience, quality understanding and the trust of the market it has gained over its past years.

**Online Platforms -2016:**

Thanks to its Web & Mobile Shop feature, Tiposoft has brought a different dimension to its activity in the market, aiming to offer betting services that is accessible everywhere, exclusive to its terminal customers.

**Journey to the Top -2017:**

Offering its brand-new software to the market with 18 sports branches and 100+ bet types, Tiposoft continues to work, produce and develop its services with the goal of being the world leader in the field.

2.1.1. Position of the Company in the World

Tiposoft provides software services and dealership to many countries of the world, as shown in the Figure 2.2. It is one of the various legal betting companies representing our country in the international arena. As per company policy, their position in other countries is not specified.



**Figure 2.2: Position of the Company**

2.1.2. Types of Services and Products

This subheading will provide information about the company's services and products. The software and products are described. The reader will have sufficient information about each product and service.

**Tipo System**

In this subheading, information is given about the working principles, service and facilities provided by Tiposoft Software Company.

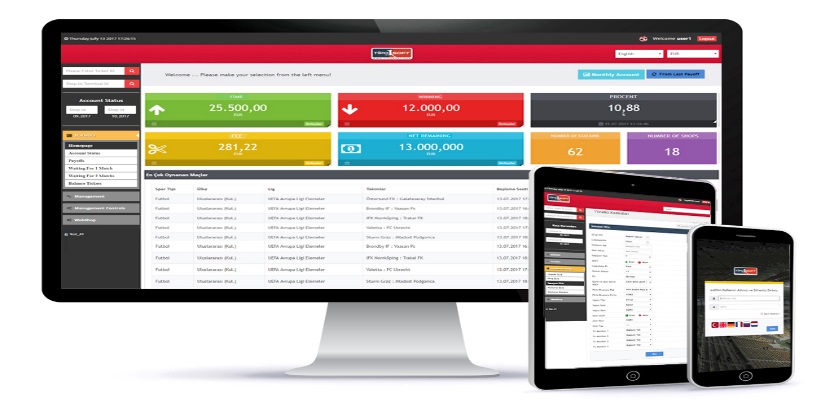
* Betting without wasting time on details such as user login and password.
* Easy deposit instead of bank and other methods for deposit and withdrawal.
* Global usage with 10 different language options.
* Easy and fast betting ability with friendly user interface.
* 500 different categories, 10,000 different tournaments.

**Tipo Admin**

In this subheading, Information about Tipo Admin is provided. The benefits and functionalities of Tipo Admin are indicated.

* The maximum and minimum amount of money that can be invested in your terminals can be determined.
* Minimum and maximum bets for sports betting and live betting can be determined.
* Users control by placing a limit on maximum winnings per bet.
* The user can personalize their coupons. The user can put a letter, logo or slogan on the coupon about the company.
* Adjustable bet confirmation time.
* By choosing the one that best suits you from the special betting packages, you can offer your customers different betting options ranging from classic betting types to the most comprehensive betting types. Users can switch between packages at any time from your admin page.
* Users can switch off unwanted bets on their terminals.
* Sports bets can be closed at the terminals and can be offered to the customer at the same time as the bets can be played.
* Users can easily and securely access all account information.

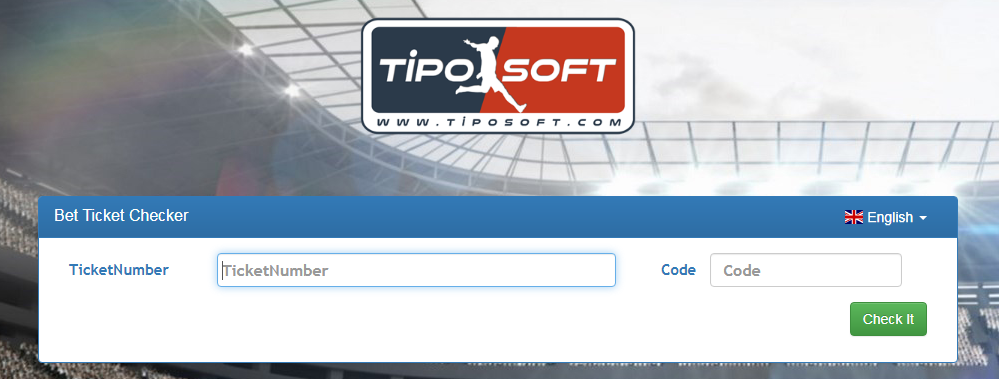
Tipo Admin simulation shown below in the Figure 2.3.



**Figure 2.3: Tipo Admin**

**Tipo Checker**

Tipo Checker is available globally, and can be operated in 6 different languages. Users can check coupons with this panel. The coupon checking system is shown in Figure 2.4.



**Figure 2.4: Tipo Checker**

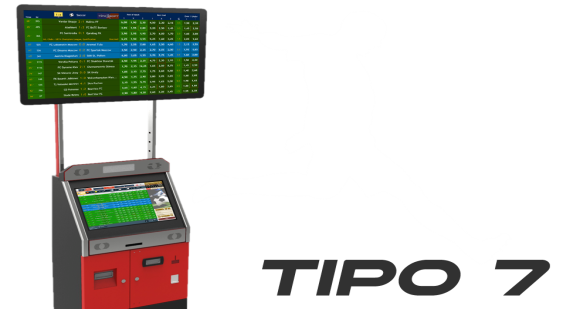
**Tipo 3**

The centre is equipped with an ultra-fast dual-core processor. For ease of use, the EloTouch Display offers users a high level of technology. This product has been created for dealers. Customers can use this machine to engage in betting. Customer's Product is shown in Figure 2.5.



**Figure 2.5: Tipo 3**

 **Figure 2.7: Tipo Mini**

 **Figure 2.6: Tipo 7**

**Tipo 7**

Tipo 7 can be expanded up to 50 inches with DVI/HDMI/VGA connections using a TV display. This machine is faster and more reliable in comparison to Tipo 3. Customers can use this machine for playing bet. Product is shown in the Figure 2.6.

**Tipo Mini**

This machine has a mini size and big technology. The Tipo Mini has an elegant and stylish design; Allowing you to visually adapt your venue and give different betting experiences with its ergonomic dimensions. Product is shown in the Figure 2.7.

**2.1.3. Certificates**

Certificates are a very important requirement for a legal bet system. Tiposoft has acquired the necessary certificates to play legal bets. ISO is Stands for "International Organization for Standardization." The ISO works with standards institutes from over 150 countries to develop technology and product standards. These standards lead to a more efficient, safer, and cleaner development of products. It also leads to more standardized products for consumers. The ISO is important to the computer industry since the organization standardizes many of the technologies used by your computer hardware and software. The required ISO certificates are shown in Figure 2.8.



**Figure 2.8: Certificates**

* 1. Departments and Personnel of the Company

Tiposoft has three main departments which are Software, IT, Control. In this subheading, the organization chart and personnel information are shown in the table in detail.

**Figure 2.9: Organizational Chart**

2.2.1. Personnel Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Staff** | **Age Scala** | **Gender** | **Education** | **Position** |
| Ali Özak | >30 | Male | B.Sc. | General Director |
| HüseyinTaşkın | >30 | Male | B.Sc. | Assistant Director/Software Developer |
| Murat Demir | <30 | Male | B.Sc. | Software Developer |
| Adnan Sezer | >30 | Male | B.Sc. | Software Developer |
| Caner Kulak | <30 | Male | B.Sc. | IT Specialist |
| Esra Danışman | <30 | Female | Associate Degree | Secretary |
| Serhan Bilek | >30 | Male | Associate Degree | Data Controller |
| Volkan Karakaş | >30 | Male | Associate Degree | Data Controller |
| Oğuz Tepeli | <30 | Male | Associate Degree | Data Controller |
| TufanYıldırım | >30 | Male | B.Sc. | Data Controller |
| Doğan Ergün | <30 | Male | Associate Degree | Data Controller |
| Hasan Kayık | >30 | Male | High School | Data Controller |
| Samet Elalmış | >30 | Male | Associate Degree | Data Controller |
| Ahmet Akay | >30 | Male | Associate Degree | Data Controller |
| Zeynep Gonca | <30 | Female | B.Sc. | Data Controller |

**Table 2.1: Personnel Information**

The table shows the staff information in detail with Table 2.1.

2.2.2. Departments

As stated in the following section, Tiposoft has three main departments, and these departments are:

**IT Department**

In the IT department, the entire computer related work of the company is handled, this includes coordinating the necessary hardware and software, planning, purchasing, and installation, writing the necessary programs, maintenance, networking and so on. The department is referred to as the responsible department. For most companies, this is a very vital part of it.

The system support expert deals with the operation and maintenance of computer systems. It is dominated by all units of computer systems such as software, hardware, and network. It can also intervene when necessary to shape the operation of the system. System support personnel are people who work to ensure the proper functioning of computer systems and are specialized in this field. The duties and responsibilities of the system support specialist are solving problems that occur in infrastructure systems used by companies.

**Software Department**

Software development is the process of thinking, defining, designing, programming, documenting, testing and error correction involved in the creation and maintenance of applications, frameworks, or other software components.

Some of the tasks and responsibilities of the software support specialist are to carry out the development and operation of the institutes application software, as well as examining existing processes to improve or automate software. Documentation of user and system requirements related to software and developing required designs is also a vital part of the departments duties. The software department is also responsible in ensuring the efficient and safe use of resources, performing system and network analysis, testing and maintenance of the company.

**Control Department**

The Control department team is responsible for checking live score results and system vulnerabilities. As an example, if there are no live betting results from the company called Betradar, the control team automatically reports to the software support department.

The software support department then responds to this problem. If the intervention is not successful, the 'IT' department can contact the company and solve the problem.

1. WORK EXPERIENCE

This section will provide all the information regarding the responsibilities I was given in the following subsections. Work-done, experiences, project process, technologies and more details are explained in the sections below.

* 1. Problem Definition

This subsection describes the working part, the work, and the framework used during my time in the company. I worked as a web developer in the Software Department, as a result of this, many projects were developed by myself using dynamic and static .Net framework. I gained a lot of theoretical knowledge from the IT specialist because of my desire to learn. Clustered systems, ISS, Unit test, storage, stress test and so on were also part of my responsibilities.

A Web developer is a kind of programmer who specializes in the development of applications relating to the World Wide Web or distributed network applications, which typically run protocols like HTTP from a Web server to a client browser using associated programming languages like HTML/CSS, JavaScript to name a few.

A Web developer is usually concerned with the back end or the programming aspect of creating a website or Web application and is not to be confused with a Web designer, who only deals with the aesthetics of a website or application, although many professionals have both skill sets.

In particular, many dynamic and static web projects have been developed using MVC technology and .Net Framework. Since these technologies are not familiar with the research, there has always been researched during the internship.

During developing, applications Used technologies, programming languages, IDE, installation, and process of tools will mention in the following sections.

My responsibilities:

* The adaptation process to the company.
* Create Static Website.
* Create Dynamic Website.
  + 1. Integrated Development Environment

An Integrated Development Environment is a type of software that aims to enable computer programmers to develop programs quickly and comfortably and includes all the tools that can organize the development process together with all the tools that contribute to the efficient use of the development process.

* + 1. Programming Language

Programming as a word means the whole process that needs to be done according to the conditions and order. The term programming refers to the process of writing a series of commands to the computer or electronic circuits. The programming language used during the internship period is briefly explained in the next subtitle.

**VISUAL C#**

C # is derived from the two most commonly used software languages, C and C ++, in the software industry. In addition, C # has many similarities with Java, a portable language programming language on common platforms. Its most important feature is a completely object-oriented software language prepared for .Net Framework platform. That is, objects are already written in classes. The programmer simply has to drag that object and then write the lines of code that will execute the object appropriately.

* + 1. Framework

The framework is the name of a structure that contains pre-prepared libraries for software developers and allows the modification and development of these libraries. The framework used during the internship period is explained in the next subtitle.

**.NET FRAMEWORK**

The .NET Framework is a software framework developed by Microsoft.  
Applications written with .Net (from C # and Vb.Net .Net languages) require .Net Framework to able to function. There are thousands of libraries in the .Net Framework.

**BOOTSTRAP**

Bootstrap is an open source, free framework. With this library, designs can be made to suit all devices. So this structure where everything is ready to be called only. Bootstrap works seamlessly by all browsers. First of all, our page should be HTML5. To use the Bootstrap library, you need to add it to the project.

* + 1. Design Pattern

Design patterns are patterns that provide solutions to problems encountered when developing applications in object-oriented languages. Design patterns are not running code or algorithm. Design Pattern used is described in the next subtitle.

**Model-View-Controller**

The MVC design pattern was first designed by Trygve Reenskaug at Xerox PARC in the 1970s. According to him, “the main purpose of MVC is to bridge the gap between the mental model of the human user and the digital model found in the computer". [1]

Throughout the 1970s, SmallTalk and some other languages such as Si-mule I gradually built the paradigm of object-oriented programming and establishing concepts such as objects, classes, encapsulation, inheritance and polymorphism [2].

Although these languages are not currently used to implement commercial applications, the concepts left in the world of software development are currently in force and are the basis of modern languages such as C ++, Java or C #.

* + 1. Database

A database is a regular collection of structured information or data, usually stored electronically on a computer system. The database is usually controlled by a database management system (DBMS). The database management system software used during the internship period is explained in the next subtitle.

**MICROSOFT SQL SERVER**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

* 1. Work Done

This subsection describes exactly what was done to accomplished the tasks described in the previous subsection.

* + 1. Adaptation Process to Company

Introduction to the internship, acquaintance with the first company, and responsibilities in the internship were discussed. In addition, each section was explained and (this section is explained in detail under “Department”). Simple information about Cloud Systems, Clustered Systems, IIS, Unit Testing, and Stress Testing was learned.

**Cloud Systems:**

One of the information obtained during the summer training internship was Cloud Systems. In this subheading, Cloud Systems is explained by myself in short words.

Cloud Systems allows users to access computer’s data and programs over the Internet rather storing them on a hard drive. When using Cloud Systems, it is possible to communicate data and information only when the Internet is available. As an example, it can provide office 365 services through web browsers to edit and store Ms Office documents without installing any program on user’s devices.

Tiposoft stored their data in the Data centre instead of the Cloud System. Because Cloud Storage has some disadvantages.

Below are some of Cloud Storage's disadvantages and why the company prefers storing its data in the Data Centre:

* **Fixed Internet Speed required and Performance issue:** Cloud computing only works when there is internet connection. Without Internet connection, it is not possible to use cloud services, so access to cloud services stops. It is not possible to access applications stored or running in the cloud. . If there is a fluctuation in the internet connection speed used or if the internet connection speed is not sufficient, it is not possible to get a healthy service from the service provider.
* **Security:** By identifying vulnerabilities in the system, it can give malicious users the opportunity. For example, on a system where more than one user resides on a server. A malicious hacker can use and steal data from users on the same server.
* **Limited Control:** Cloud Systems are controlled by Internet Service Providers and therefore have limited access control over data and programs. Although cost-effective, they may not be preferred for such reasons.

That is why Tiposoft kept their data in the data center. Generally Big Companies kept their information in the data centers.

**Clustered Systems:**

Tiposoft Company use this system on their servers. Cluster architecture or cluster structures can simply be defined as specific configurations are clusters of servers running together or redundantly for the same purpose. To create each cluster in a cluster structure, two or more servers called "nodes" are needed.

The Tiposoft Company uses this system for its high availability and for performance improvement. During the internship, the System Specialist showed me the “Load Balancing” sets and explained to me how it is operated.

* **Load Balancing:** Load balancing is the process of sharing incoming network traffic between servers called a server farm or server pool. Service that provides load balance and performance by distributing incoming requests to multiple servers. For example, a website we serve is installed on 2 servers. A cluster is created from these servers with the cluster structure. When one server is off in this cluster, the other server continues to serve and access is not interrupted. This minimizes the margin of error. In the event of a server shutdown; the load balancer directs traffic to the online servers. If a new server is added to the server farm, the load balancer automatically sends the request to the new server.

**Internet Information Services (IIS):**

During the internship, information was obtained about what the IIS is and what it is used for. Internet Information Services is the web server of Windows operating systems. Internet Information Services host and delivers Web applications for publishing. Observed during the internship, the system specialist used it for test applications.

**Unit Test:**

Unit testing is the testing of software units, as the name implies. The so-called software unit is the smallest software component that can be tested. Given the object-oriented programming approach, software units are classes. What is done is simply to check whether the methods work correctly when certain inputs are provided and produce the desired result.

**Stress Test:**

The purpose of stress testing is to put stress on the system by putting pressure on the prompt. For example, if this tested application has a cluster structure, we can learn how the system behaves by sliding one of the nodes in the cluster.

* + 1. Create Static Website

Entry into the internship, meeting with the first company and what to do during the internship were evaluated. It was also experienced what each unit does. After the meeting, I personally investigated the .NET Framework and what I could do with it. This was the first mission.

With the recommendation of my supervisor, I was required to create a static website project to improve me technically and practically. I learned that making a static website would be easier than dynamic.

Firstly, to develop a web project in MVC, the concepts of model view and controller need to be well understood. Therefore, the logic of the model view and controller concept was investigated. The concept of ActionResult was also explained to me. ActionResult is the method that processes requests according to the controller structure and sends the information back to the user with View.

At my attempt to design a website, the HTML and CSS section took a lot of time, and as a result of research, a framework called bootstrap was discovered. It would be a practical method for the boot framework, and only the background of the work was discussed. A static template is integrated into the project. The template was downloaded from this site https://w3layouts.com/. But there was a problem, the CSS section did not exist, the file path was found to be incorrect, after reviewing this error, the bug was corrected. Finally, a Static Website was made.

* + 1. Create Dynamic Website

My supervisor decided my next responsibility would be to make a dynamic website and necessary research was made on the internet. A project in which stock follow-up was started.

**Create Database**

When starting the project, the first task was to create a database from SQL Server. A database was created under the name MvcDbStock. Under MvcDbStock, 4 basic tables were created: products, customers, sales and categories.

The first table was created, a name was written for each product, and 3 basic headings called product ID, product category, price, and stock were created.

The second table was called customers. For each customer, customer name, customer name and customer surname were created.

It was thought that there should be a category for each product, and again the category ID, category name for each category created in the category.

Sales names were created in the fourth table. Because customers had to keep the data in the database when buying something. Sales ID, product, customer, quantity, and price were created for each sale.

**Create Database Diagram**

A database diagram was used to determine the relationship between the tables. Three relationships were established between the tables. The first relationship was to establish a relationship between the category and the product table. This means that a category can contain multiple products. The second relationship was to establish a relationship between the customer and the sales table. The third relationship was to establish a relationship between the product and the sales table. The database schema is shown in Figure 6.1.

**Data Entries**

Product information was entered in the generated tables and database part is completed.

**Model Creation**

In the MVC world, this model is where application data or status is stored, so the project used ADO.NET's asset data model. A model based on an existing database in EF Designer was created. The database connection, settings for the model, and database objects to be included in the model can be selected. The classes of the application consist of a model to interact with and the tables associated with the project were implemented. Each table is automatically assigned to the project as a class. Database First Method Selected. Users have 3 options as can be seen in Figure 6.2.

**Creation Category Controller and Listing Category**

A controller was created and made the outlook for the category table and then used in order to view categories. An error was received as shown at this point. Figure 6.3. After research on the internet, one can learn how to correct this problem. The error came from the RenderBody () section because there was a RenderBody () and Layout relationship. The layout cannot be used without using RenderBody ().

Before starting this section, “Model” created in “CategoryContorler ”was defined as a library. As you can see, Figure 6.4. Because of the need to use and manage the data within the model, an object named “db” is derived to access the tables, after which values are created to hold the objects as you can see in the categories. Figure 6.5. A table was created in the "view" section of the category, and data was added to the table structure created using the Foreach loop.

**Creation of Products Controller and Listing Products**

First, an empty controller was added to my product Controllers project. For a list of products, the model was redefined as a library (previously created) in the project. This was done because the model required access to the data. An object named db is derived for access tables and the ToList () method is used to list products. After that, a view was added for the project. At the top of the product view page, the model is also identified. After that, a table was created and each product information was written below the <tr> table row.

The foreach loop was used to extract all data from the database.

**Creation of Customer Controller and Listing Customers**

An empty controller was added to the project named Customer Controllers. For a list of customer information, the model was redefined as a library. As mentioned earlier, it was possible to access the data in the model because it was done. An object named "db" was derived to access the tables, and the ToList () method was used to list customers. A view was then added to the project. At the top of the product display page, the model is also identified. A table was then created and each product information was written below the <tr> table row. The foreach loop was used to extract all data from the database.

**Giving Links to Menus**

The goal of the link in this section was written to achieve the goal. A code example is shown below. A link has been written to href. And an example of a piece of code is shown in Listing 3.1.

<td>< a href="/PRODUCT/CallProduct/@prod.productID"class="btnbtn-warning">Update</a></td>

**Listing 3.1: Giving Link to Menus**

**Creating Update and Delete Functions**

It was necessary to add an update and delete button for the project. In fact, there was no idea that the research had the necessary information and how to do it. Finally, the delete and update buttons were explicitly added and the bottom of the page is added below the new category button. Example of appearance shown in Figure 6.6.

**Implementing HTTP Get and HTTP Post Methods**

I noticed that each time the page refreshes, it adds new data. Research was done to resolve this error. I learned that HTTP get, HTTP post methods should be used for this situation. If the user does not do anything on the page, the [HTTP Post] method performs the operation when the user presses the save button after any action, no changes are made to the page.

**Adding New Category**

In the first step, a new ActionResult is created on the category controller, and then a new view is created for the New Category Page. After that, a form was created on the View Page. On the View Page of the New Category shown in Figure 6.7 and Figure 6.8.

**Adding New Customer**

In the first step, a new ActionResult is created on the customer controller, and then a new view is created for the New Customer Page. After that, a form was created on the View Page. On the View Page of the New Customer shown in Figure 6.9 and Figure 6.10.

**Adding New Product**

A new ActionResult was created on the product controller, and a new view was created for the New Product Page side. After that, a form was created on the View Page. On the View Page of the New Product shown in Figure 6.11.

**Usage of Dropdownlist and Extracting Data from the Database**

I continued my research to create a useful project and thought it would be useful to use DropDownList. And after detailed research I came across an article about the DropdownList, and took advantage of this article [7]. How this control element is applied in the project is explained. First, the element <SelectListItem> was defined, it is an element that helps to select the item from the list. Then the variable 'i' was created to get values from the table categories. Sends the category name from the variable 'i' to the text variable and sends the category 'ID' from the 'i' variable to the Value variable and can be listed using the toList () method. Finally, the viewbag command was used to transfer data between pages.

**Usage of LINQ**

As can be seen in Figure 6.12, method syntax comprises of extension methods and Lambda expression. The extension method Where() is defined in the Enumerable class.

**Adding New Product with LINQ and DropdownList**

Some options on the new product side have been changed and thought to be more useful. The drop-down list for the product category section was used as you see in Figure 6.13. A drop-down list allows users to choose an option that is more convenient and reliable.

**Fetch Category Name by ID**

An example of a piece of code is shown in Listing 3.2.

</td>@prod.Table\_Category.categoryName</td>

**Listing 3.2: Fetching Values**

**Removing Categoy**

First, the RemoveCategory function was created for this. This function is explained step by step. And an example of a piece of code is shown in Listing 3.3.

public ActionResult RemoveCategory(int id)

{

var category = db.Table\_Category.Find(id);

db.Table\_Category.Remove(category);

db.SaveChanges();

return RedirectToAction("INDEX);

}

**Listing 3.3: Removing Category**

First, the id value in the category is found, the values in the category are deleted, and the changes are saved and redirected to the Index page.

**Removing Product**

Firstly, find the value from id in categories; delete the value from category and then Saving changes and Redirecting to the Index page. And an example of a piece of code is shown in Listing 3.4.

public ActionResult RemoveProduct(int id)

{

var product = db.Table\_Category.Find(id);

db.Table\_Product.Remove(product);

db.SaveChanges();

return RedirectToAction("INDEX);

}

**Listing 3.4: Removing Product**

**Removing Customer**

The value of the identity on the customer is found, the value is deleted from the customer, and then the changes are saved and redirected to the Index page. And an example of a piece of code is shown in Listing 3.5.

public ActionResult RemoveCustomer(int id)

{

var customer = db.Table\_Customer.Find(id);

db.Table\_Customer.Remove(customer);

db.SaveChanges();

return RedirectToAction("INDEX);

}

**Listing 3.5: Removing Customer**

**Move Data between Pages**

CallCategory created an Actionresult that brings categories by category. The value from "id" is assigned to the category variable. A view to the controller was also added. Necessary definitions were made for the viewing page. A form method was created on the page. Category ID and category names were created with LINQ queries. After making these upgrades to the page, it could then be used to update information.

**Category Information Updating**

For this, the Update Function was created. This function will be explained step by step. And an example of a piece of code is shown in Listing 3.6.

public ActionResult Update(Table\_Category p1)

{

Var category = db.Table\_Category.Find(p1.categoryID);

category.categoryName = p1.categoryName;

db.SaveChanges();

return RedirectToAction("INDEX);

}

**Listing 3.6: Category Information Updating**

This update function updates according to the parameter. It will assign the name from the parameter to the category name.

**Adding Validation Controller**

One of the sine qua non for a project is verification. This is why validation forms were used in the Project.

**Product Information Updating**

The product information section was made. In this section, LINQ query is used from the project. The parameter has been created. And an example of a piece of code is shown in Listing 3.7.

public ActionResult UpdateProduct(Table\_Product p1)

{

var product = db.Table\_Product.Find(p1.productID);

product.productName = p1.productName;

product.stock = p1.stock;

product.price = p1.price;

product.brand = p1.brand;

product.productCategory = p1.productCategory;

db.SaveChanges();

return RedirectToAction("INDEX);

}

**Listing 3.7: Product Information Updating**

**Product Validation Controlling**

A sample from codes, the user just can enter the 30 characters and also users can not this field be blank. And an example of a piece of code is shown in Listing 3.8.

<div>

<label>Product Name</label>

<inputtype="text"class="form-control"name="productName"required=""maxlength="30"/>

</div>

**Listing 3.8: Product Validation Controlling**

**Implementing Paging**

For a better and professional opinion the paging process was used. PagedList. MVC plugin was installed from the platform. After the installation, a paged list was defined for the project. The corresponding view of the project is shown in Figure 6.14. This code provides 3 values for each page. And an example of a piece of code is shown in Listing 3.9.

var values = db.Table\_Category.ToList().ToPagedList(page,3);

**Listing 3.9: Implementing Paging Code Sample**

This Code providing to users dividing pages all information. And an example of a piece of code is shown in Listing 3.10.

@Html.PagedListPager((IPagedList)Model, page =>Url.Action("Index", new{ page }))

**Listing 3.10: Implementing Paging another Code Sample**

**Implementing Alert**

The new Customer section was created under the script field and a button ID was written to this function. The following is a code example. And the front end side is shown in Figure 6.15.And an example of a piece of code is shown in Listing 3.11.

<script>

$('#btn1').click(function () {

alert('CustomerAdded');

});

</script>

**Listing 3.11: Implementing Alert**

**Deleting with Alert**

In this section, an ActionLink named Delete Button was created when the user clicked on this button program that would give a confirmation message. The message section is written into the overlay function, as seen from the front end shown in Figure 6.16. And an example of a piece of code is shown in Listing 3.12.

<td>@Html.ActionLink("DeleteButton", "RemoveCustomer", new { id=c.customerID }, new { @class = "btnbtn-warning", onclick = "returnconfirm('Are Sure For That?')" })</td>

**Listing 3.12: Deleting with Alert**

**Implementing Search Panel**

One of the most important things for dynamic websites is the Search Panel, so I decided to use this function in the project. Microsoft articles help when applying this option. A sample of view about Search Panel shown Figure 6.16. And an example of a piece of code is shown in Listing 3.13.

public ActionResult Index(string p)

{

var values = from val in db.Table\_Customer select val;

if (!string.IsNullOrEmpty(p))

{

values = values.Where(m => m.customerName.Contains(p));

}

return View(values.ToList());

}

**Listing 3.13: Implementing Search Panel**

* 1. Limitations and Experience Gained

During my time working with Tiposoft, I gained a lot of valuable knowledge in the computer engineering field. While there were a lot of errors and problems that were faced during my summer training internship, I realized that with each error, there was more knowledge to be gained. This idea was always kept in mind. With every problem that was encountered, it was tried and solved. The research started in places that were missing from the beginning and not seen enough. Usually, there are codes to solve the problem within a few hours. When solving a problem, anyone can fail. It was learned that he should never give up. Search and analyze all options, then find the best and shortest path.

In summary, new techniques have been learned to solve computer problems. Analyze the problem and order to an emergency level. This experience will help me when it comes to Software Architecture in the future. Many problems were solved by using different solutions in the training company. So almost all of them were taken care of and the department worked efficiently.

* + 1. Limited Time

Summer internship period and the number of the internship is not enough. If the duration of the Summer Internship will be longer or the number of internships will increase, students will gain much more knowledge and experience during the summer training programme.

* + 1. What Was Lacking?

The general deficiency seen was the lack of respect for individuals studying in our country. Generally, companies give more value and importance to knowledgeable people. That was a very sad situation for me.

* + 1. How to Gain More Experience?

Since the summer training period that is thirty-two days was not enough, in thirty-two days I had the opportunity to learn new technology, apply them, and an observation about business life was experienced. I always tried to learn something and tried to make good use of my time to improve myself, but time was insufficient.

1. CONCLUSION

The timing has never been better for using technology to enable and improve learning. All levels, in all places, and for people of all backgrounds. The key practices necessary to realize best the transformations made possible by technology in education are in place with summer training.

The purpose of the internship is to acquire practical, technical and managerial knowledge and experience in order to contribute to their education outside the academic curriculum and to define their work-life and employee-employer relations. It is a time of great possibility and progress for the use of technology to support learning.

One of the most important benefits of doing a conscious internship is learning what competencies are necessary and valid in business life.

In this internship, I learned how to start a project, make documentation, plan the project and take responsibility. Furthermore, analytical thinking was experienced. In the end, this internship was a start for business.

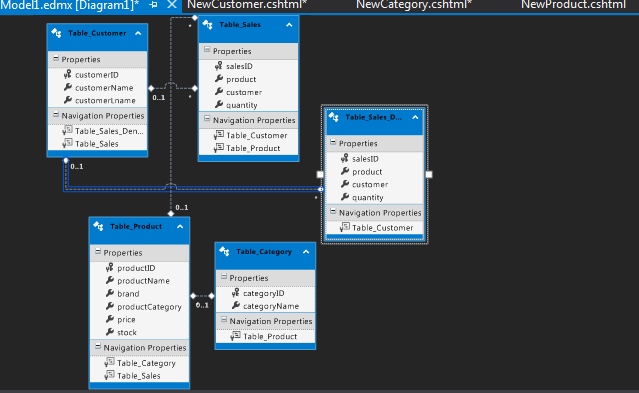
1. REFERENCES

This section must include all references listed as follows;

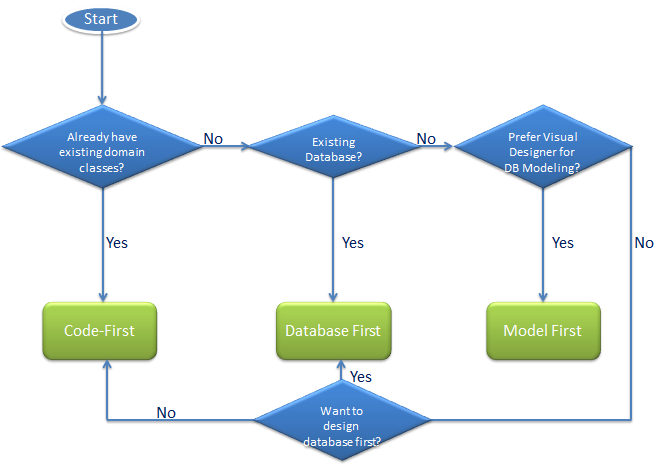
1. TrygveReenska, http:// heim.ifi.uio.no / ~trygver/themes/mvc/mvc-index.html,2019.

2. Steve Burbeck. Applications Programming in Smalltalk-80(TM): How to use Mo-del-View-Controller (MVC).http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html,2019.

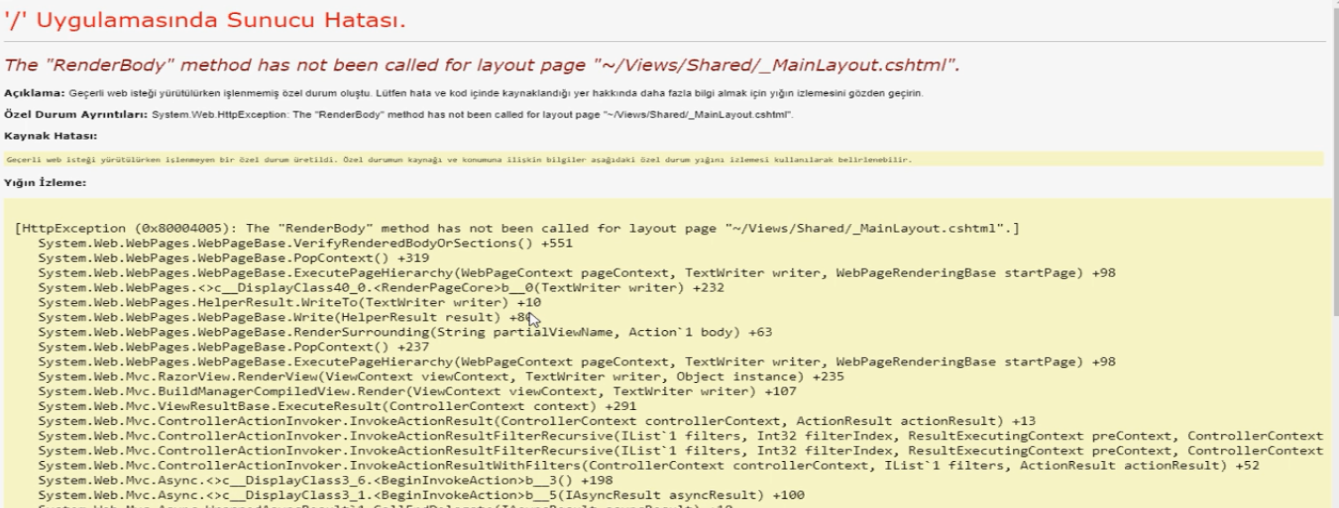
1. APPENDIX



**Figure 6.1: Database Diagram**



**Figure 6.2: Entity Frameworks**



**Figure 6.3: RenderBody Error**

using System.Web;

using System.Web.Mvc;

using StockMVC.Models.Entity;//Defined Model Entity

using PagedList;

using PagedList.Mvc;

namespace StockMVC.Controllers

{

**Listing 6.1: Defining Model Entity**

MvcDbStockEntities db = new MvcDbStockEntities();

public ActionResult Index()

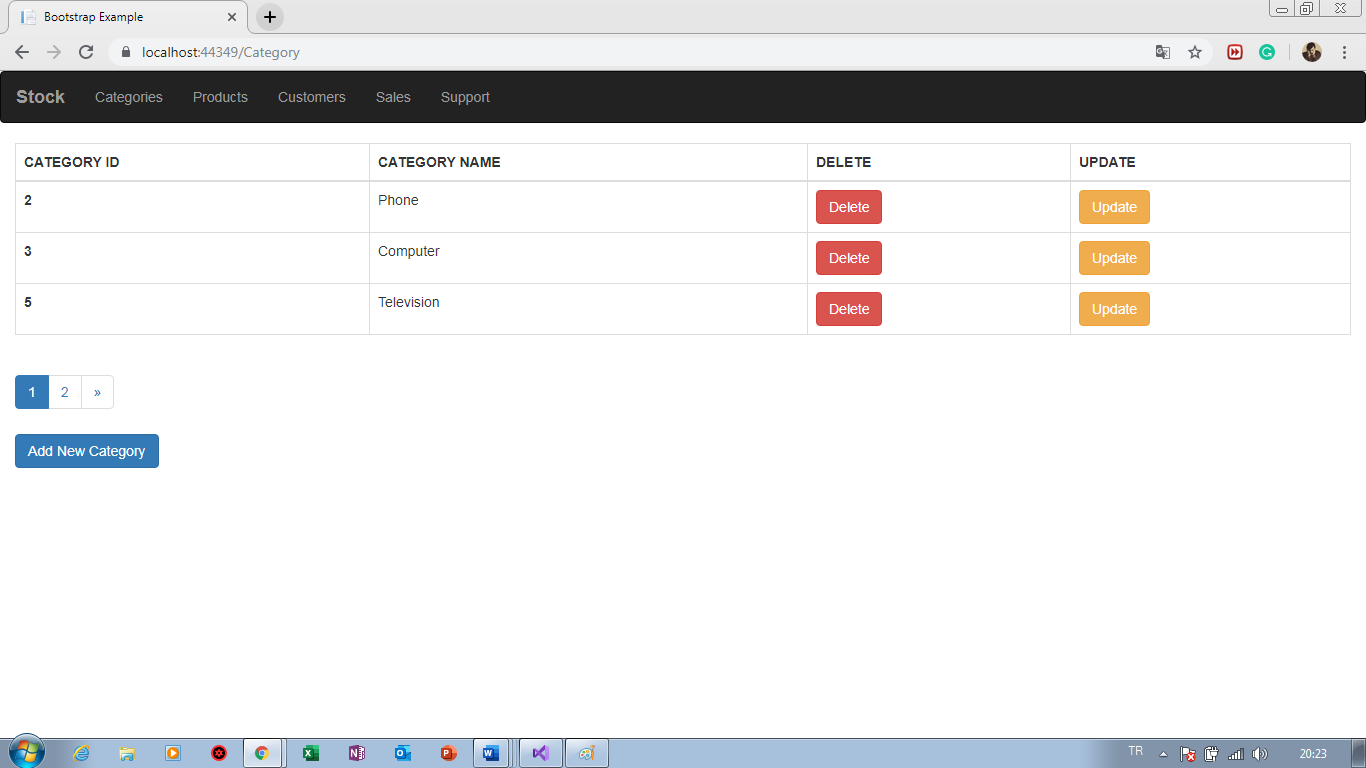
{

var values = db.Table\_Category.ToList();

return View(values);

}

**Listing 6.2: Index of Category Controller Codes**



**Figure 6.4: Page of Category Section**

@model StockMVC.Models.Entity.Table\_Category

@{

ViewBag.Title = "NewCategory";

Layout = "~/Views/Shared/\_MainLayout.cshtml";

}

<form class="form-group" method="post">

<div>

<label>Category Name</label>

@Html.TextBoxFor(m => m.categoryName, new { @class = "form-control" })

@Html.ValidationMessageFor(m => m.categoryName, "",new {@style="color:red" })

</div>

<div style="margin-top:5px ">

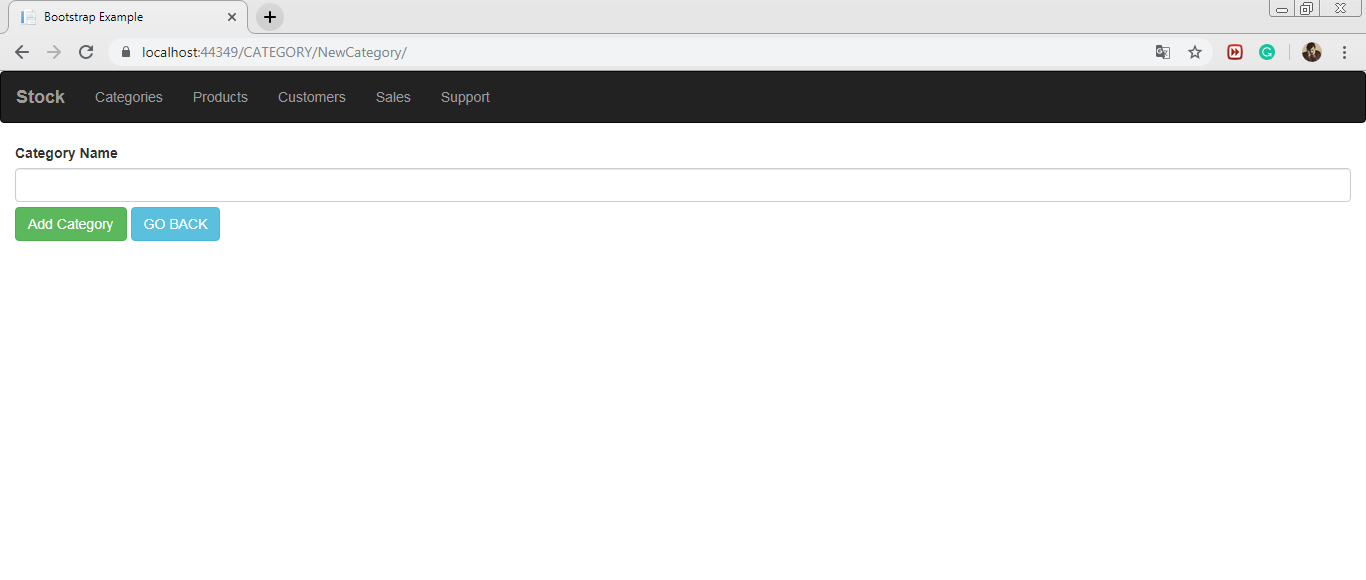
<button class="btn btn-success">Add Category </button>

<a href="/category" class="btn btn-info">GO BACK</a>

</div>

</form>

**Listing 6.3: Codes of New Category Section**



**Figure 6.5: View of New Category Section**



**Figure 6.6: View of New Customer Section**

@model StockMVC.Models.Entity.Table\_Customer

@{

ViewBag.Title = "NewCustomer";

Layout = "~/Views/Shared/\_MainLayout.cshtml";

}

<form class="form-group" method="post">

<div>

<label>Customer Name</label>

@Html.TextBoxFor(m=>m.customerName,new {@class ="form-control" })

@Html.ValidationMessageFor(m => m.customerName, "", new {@style="color:red" })

<br />

<label>Customer Lastname</label>

@Html.TextBoxFor(m => m.customerLname, new { @class = "form-control" })

@Html.ValidationMessageFor(m=>m.customerLname,"",new {@style="color:red" })

</div>

<div style="margin-top:5px ">

<button class="btn btn-success" id="btn1">Add Customer </button>

<a href="/CUSTOMER" class="btn btn-primary">GO BACK</a>

</div>

<script>

$('#btn1').click(function () {

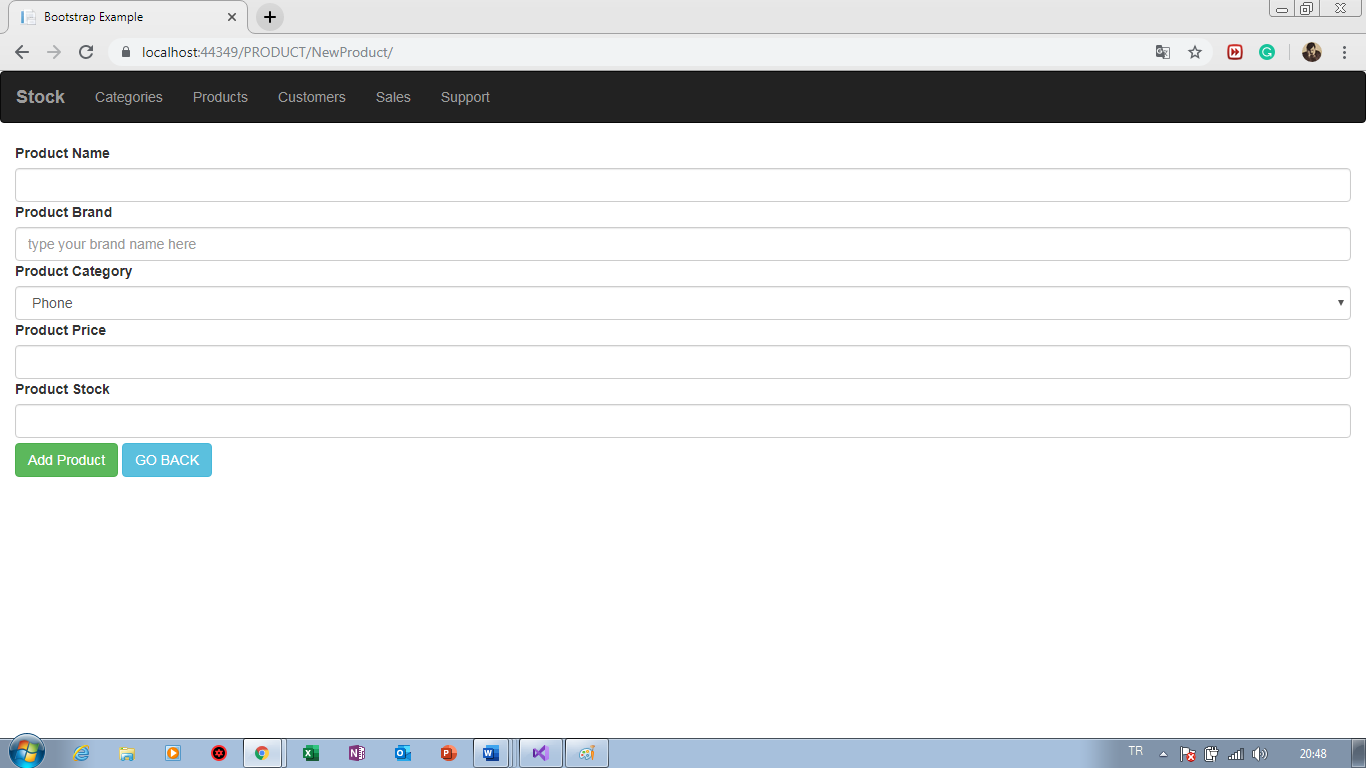
alert('Customer Added');

});

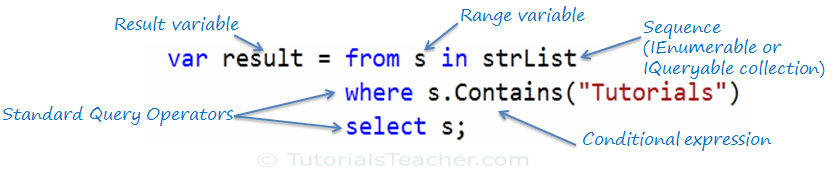
</script>

</form>

**Listing 6.4: Codes of New Customer Section**



**Figure 6.7: View of New Product Section**

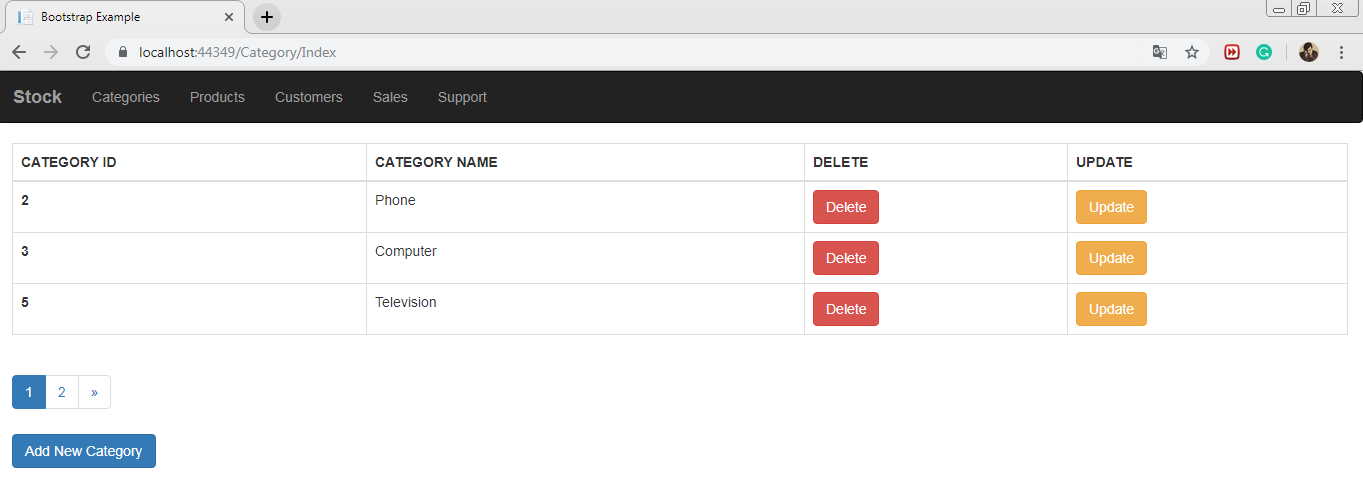


**Figure 6.8: Syntax**

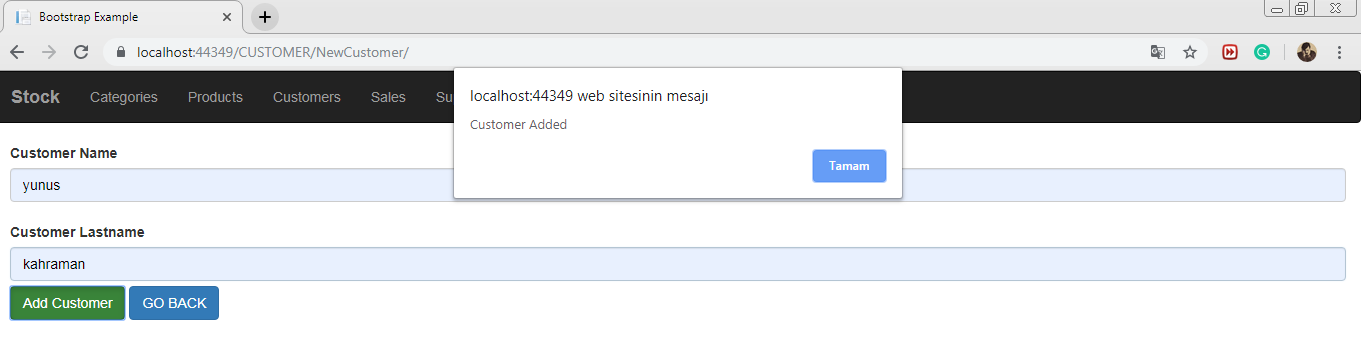
<label>Product Category</label>

@Html.DropDownListFor(m => m.productCategory, (List<SelectListItem>)ViewBag.Categories, new { @class = "form-control" })

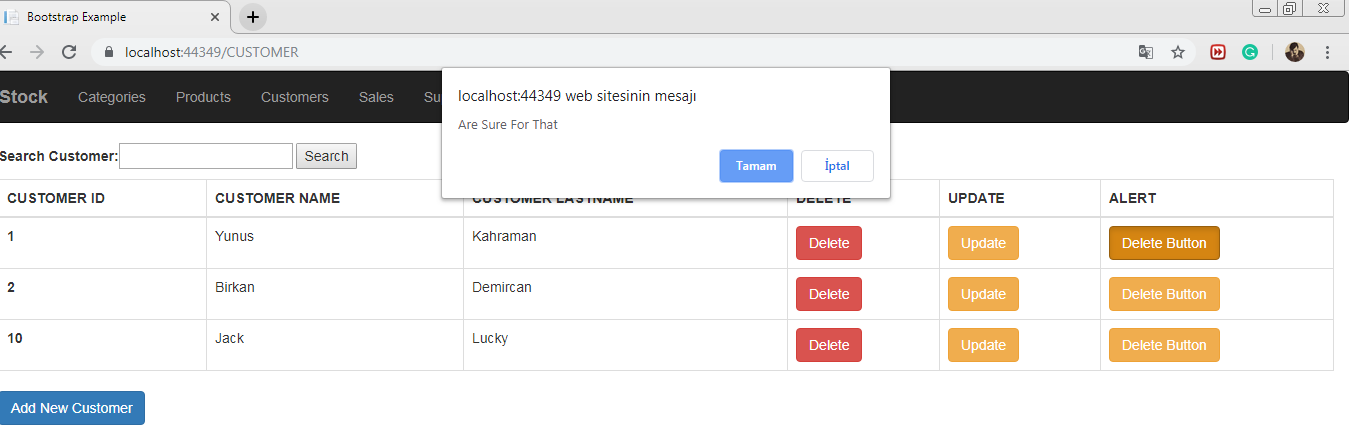
**Listing 6.5: Usage of LINQ**



**Figure 6.9: Paging**



**Figure 6.10: Implementing Alert**



**Figure 6.11: Delete Alert**