NAME – SURNAME: BETÜL BERNA SOYLU

UNIVERSITY: ADNAN MENDERES UNIVERSITY

BACHELOR DEGREE: COMPUTER ENGINEERING

POSTGRESQL DATABASE

- User ID and Password must be changed in DAL (Data Access Layer) layer to connect to PostgreSQL database.

```
pgSqlConnection1.Host = "localhost";
pgSqlConnection1.Port = 5432;
pgSqlConnection1.UserId = "postgres";
pgSqlConnection1.Password = "237233";
pgSqlConnection1.Database = "Homework";
```

- The script of the Log folder in our PostgreSQL database will be as follows:

DROP TABLE IF EXISTS public."Logs";

CREATE TABLE IF NOT EXISTS public."Logs"

(

   "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),

   "Method" text COLLATE pg_catalog."default",

   "Path" text COLLATE pg_catalog."default",

   "Query" text COLLATE pg_catalog."default",

   "CreatedTime" date,

   CONSTRAINT "Logs_pkey" PRIMARY KEY ("Id")

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Logs"

   OWNER to postgres;

GRANT ALL ON TABLE public."Logs" TO postgres;

GRANT ALL ON TABLE public."Logs" TO PUBLIC;-- Table: public.public.Logs

-- DROP TABLE IF EXISTS public."public.Logs";

CREATE TABLE IF NOT EXISTS public."public.Logs"

(

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."public.Logs"

  OWNER to postgres;

SQL SERVER DATABASE

- In order to connect to the SQL Server database, the Server and Initial catolog values must be changed in the DAL (Data Access Layer) layer.

```
string connectionString = "Server=.;Initial catalog=deneme;Integrated Security=True";
SqlConnection connection;
SqlCommand command;
List<Log> logList;
```
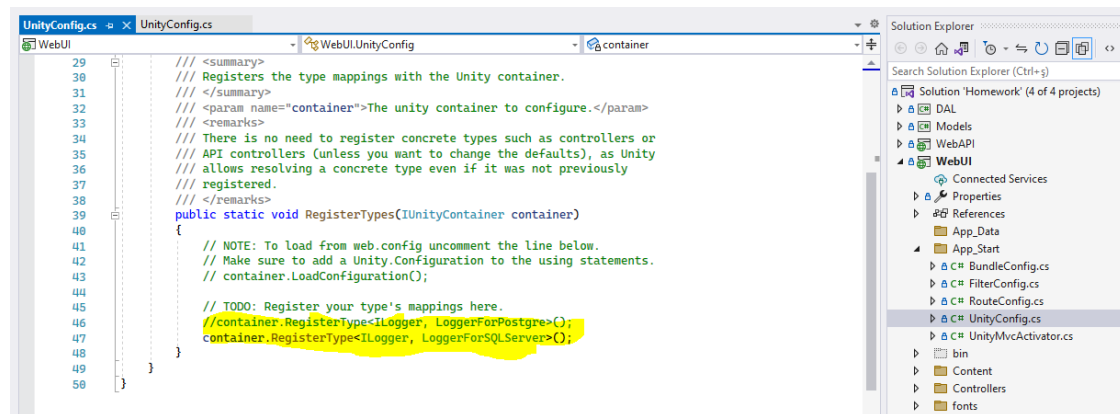
- Our screen output is as follows:

| Homework | Home | Category | Order | Supplier | Product | Shipper | Employee | Customer |

Search : [_____] [Go]

Id	Method	Path	Query	Created Time
1	GET	/		17.05.2022 11:55:05
2	GET	/		17.05.2022 12:48:23
3	GET	/	query=	17.05.2022 12:49:40
4	GET	/		17.05.2022 15:38:46
5	GET	/	query=query	17.05.2022 15:38:57
6	GET	/		17.05.2022 15:39:02
7	GET	/		17.05.2022 15:44:02

- Unity Mvc Framework was used to inject MVC. On the Unity.mvc page, we can choose both PostgreSQL and SQL Server databases for Web Api and Web interfaces.

WEB API:

```
using Models;
using System.Web.Http;
using Unity;
using Unity.WebApi;
using DAL;

namespace WebAPI
{
    public static class UnityConfig
    {
        public static void RegisterComponents()
        {
            var container = new UnityContainer();

            // register all your components with the container here
            // it is NOT necessary to register your controllers

            container.RegisterType<ILogger, LoggerForPostgre>();
            //container.RegisterType<ILogger, LoggerForSQLServer>();

            GlobalConfiguration.Configuration.DependencyResolver = new UnityDependencyResolver(container);
        }
    }
}
```

WEBUI:

```
        /// <summary>
        /// Registers the type mappings with the Unity container.
        /// </summary>
        /// <param name="container">The unity container to configure.</param>
        /// <remarks>
        /// There is no need to register concrete types such as controllers or
        /// API controllers (unless you want to change the defaults), as Unity
        /// allows resolving a concrete type even if it was not previously
        /// registered.
        /// </remarks>
        public static void RegisterTypes(IUnityContainer container)
        {
            // NOTE: To load from web.config uncomment the line below.
            // Make sure to add a Unity.Configuration to the using statements.
            // container.LoadConfiguration();

            // TODO: Register your type's mappings here.
            //container.RegisterType<ILogger, LoggerForPostgre>();
            container.RegisterType<ILogger, LoggerForSQLServer>();
        }
    }
}
```

WEB API

- In order to be able to access the Web API, the port number after localhost in the Controllers layers in the WebUI layer must be changed to suit your own computer.
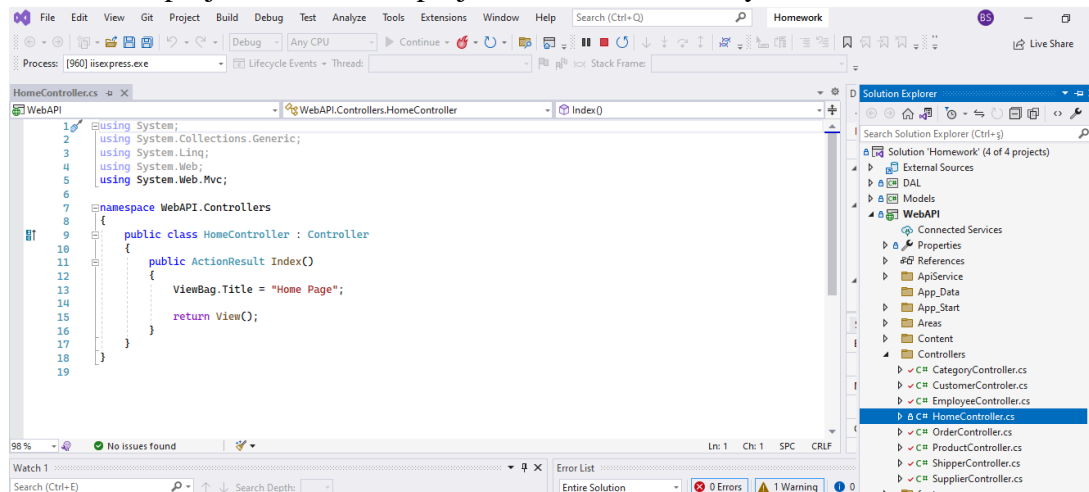
```
namespace Homework.WebUI.Controllers
{
    public class CategoryController : Controller
    {
        private readonly HttpClient httpClient;
        public CategoryController()
        {
            httpClient = new HttpClient();
            httpClient.BaseAddress = new Uri("https://localhost:44313/api/");
        }
    }
}
```
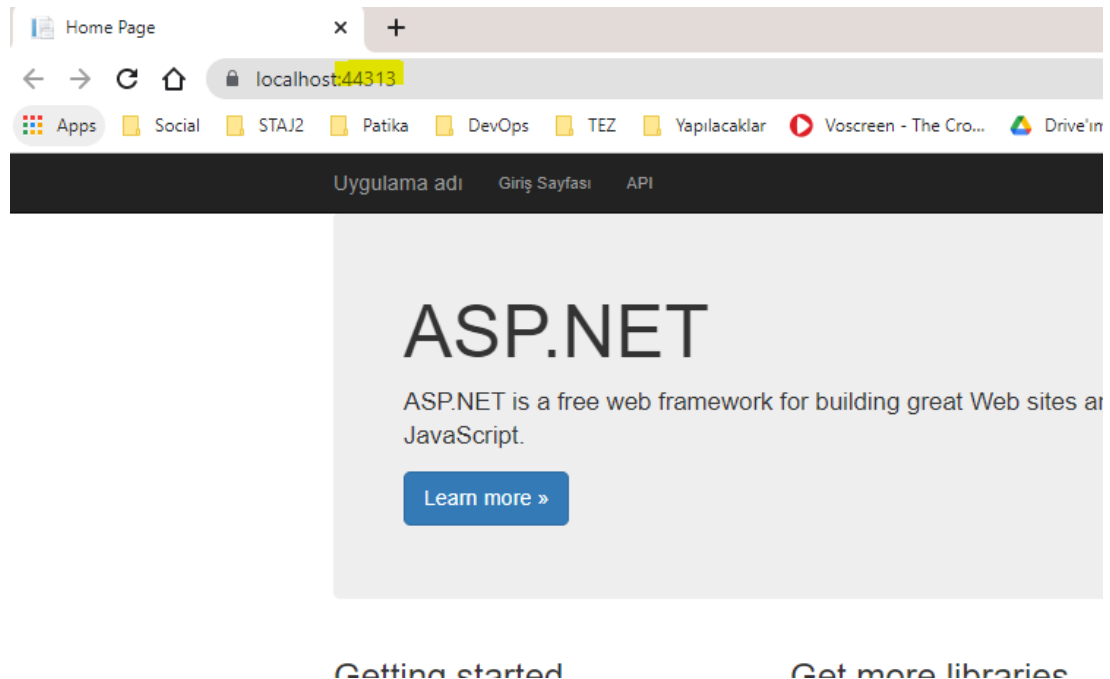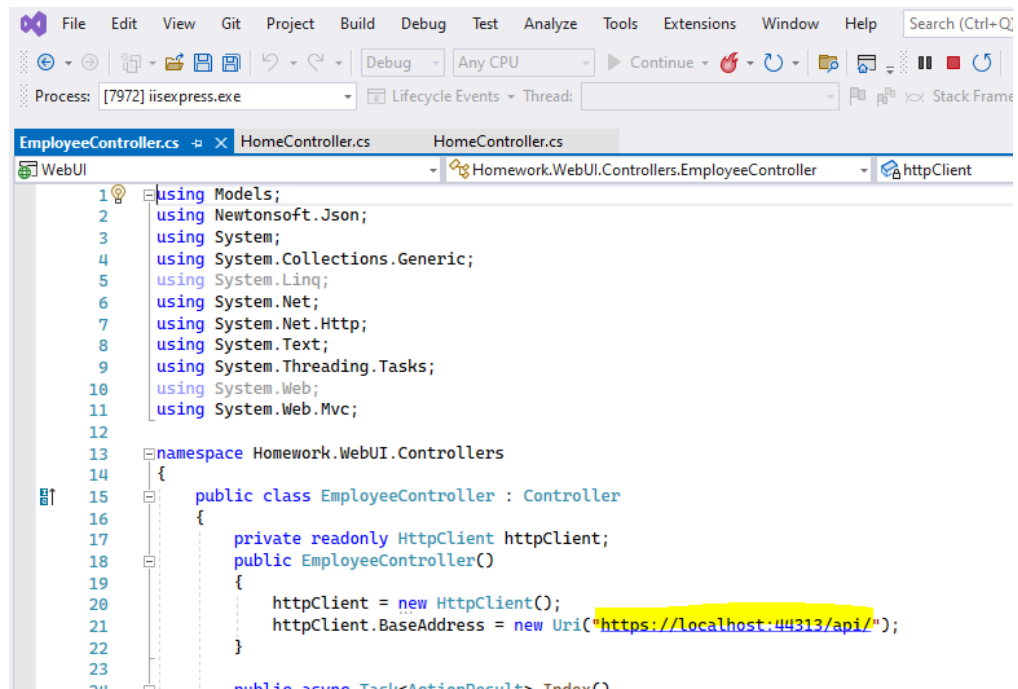
WEB API TEST and RUNNING THE PROJECT

- First of all, start the project by opening 2 visual studios. Run the Web Api layer in 1 visual studio project.In the other project, run the WebUI layer.

- Get the Web Api Port Number carefully.

- Make sure that this Port number is entered correctly in the WebUI controller interface:
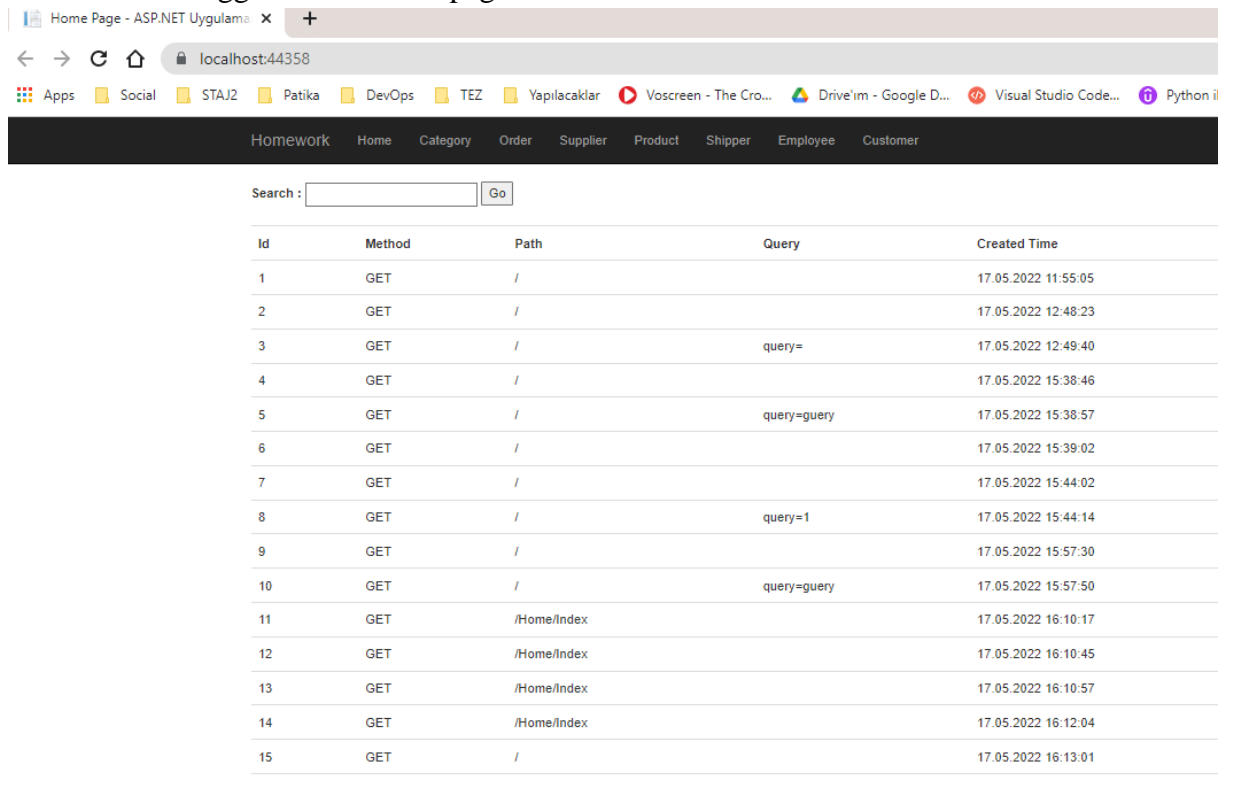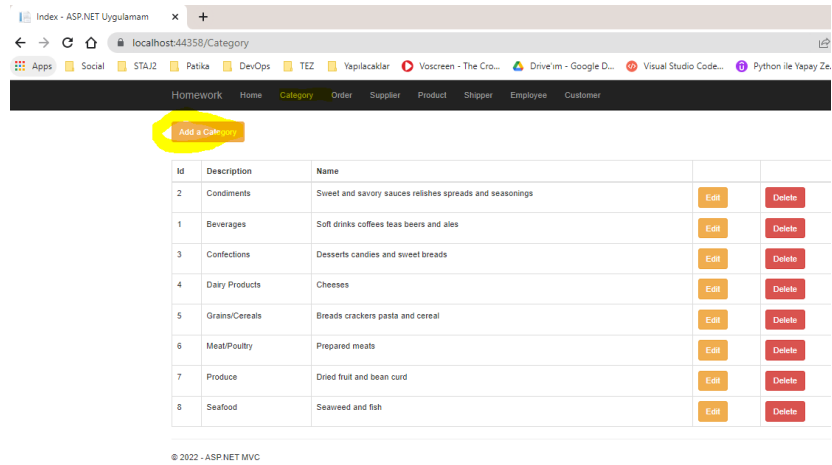

- You can find Logger on the home page.

- For example, to add a new category, you can click the Category Tab button and click the Add Category button.

- You can add a new category.

- The category we added appears in the picture.