Veritaban Yönetim sistemi
Projesi

Konu :

Gayrimenkul Satış sitesi

Öğrenci Adı Soyadı : Burak Özdemir

ÖğrenciNo: s211210060

Mail:obada.alhourani@std.izu.edu.tr

Proje Tanımı :

Kısa özeti

Kolay bir kullanıcı arayüzü ile gayrimenkul satış ve kiralama sürecini kolaylaştırmayı amaçlayan bir gayrimenkul satış sitesi
Satıcı-müşteri ilişkisi, iletişim ve satıcı tarafından tedarik fikrinin ve kullanıcı tarafından talep edilmesinin basitleştirilmesi açısından daha kolay hale gelmiştir.
Kullanıcının önündeki seçeneklerin çeşitliliğinin yanı sıra, baskı hissetmeden sayfalar arasında istediği gibi serbestçe hareket edebilir ve sitede sergilenen Emlak danışmanlığı hakkında bilgi alması için iki yolu var iletişim kısmından irtibat kurması veya mail adresi ile abone olmak üzere girilen mail adresine yeni ilanlar hakkında bilgilendirme mesajı almakta
Site ayrıca yatırım özellikleri ve ortakların varlığını da sağlar.
Bu, iş kaynaklarının çeşitliliği ve yalnızca bir modele bağlı kalmama açısından satıcıya daha fazla kar katmaktadır.

İş kuralları :

Satıcı tanıtmak istediği hizmet türü kategori kısmına ekler

Ör: kategori içinde kıralama – satış gibi unsurler bulunmakta (post)

Her kategori içinde subkategori belirleyebilir

Ör: daire – Residence – villa – iş ofisleri

Her subkategori içinde şirketin Gayrimenkul ilanlarını sergilebilir

Resim – Ünvan – Açıklaması ekleyerek berlirler (post)

Home kısmına slider ekleyebilir

Title - açıklama – resim

Social media öğelerini ekleye bilir

Ortaklar kısmına ortaklık veya işbirliği yapılan şirketin logosunu ekleyebilir

Eklenen mail adresler Mail kısmında listelenir ve gerekli bilgilendirmeler gönderilir.

Konum ekleyebilir ve değiştirebilir .

Ziayretçi ise   Home page – Kategori – about Us sayırfalarına tıklayabilir ve görüntüleyebilir

İki dilden birini seçebilir

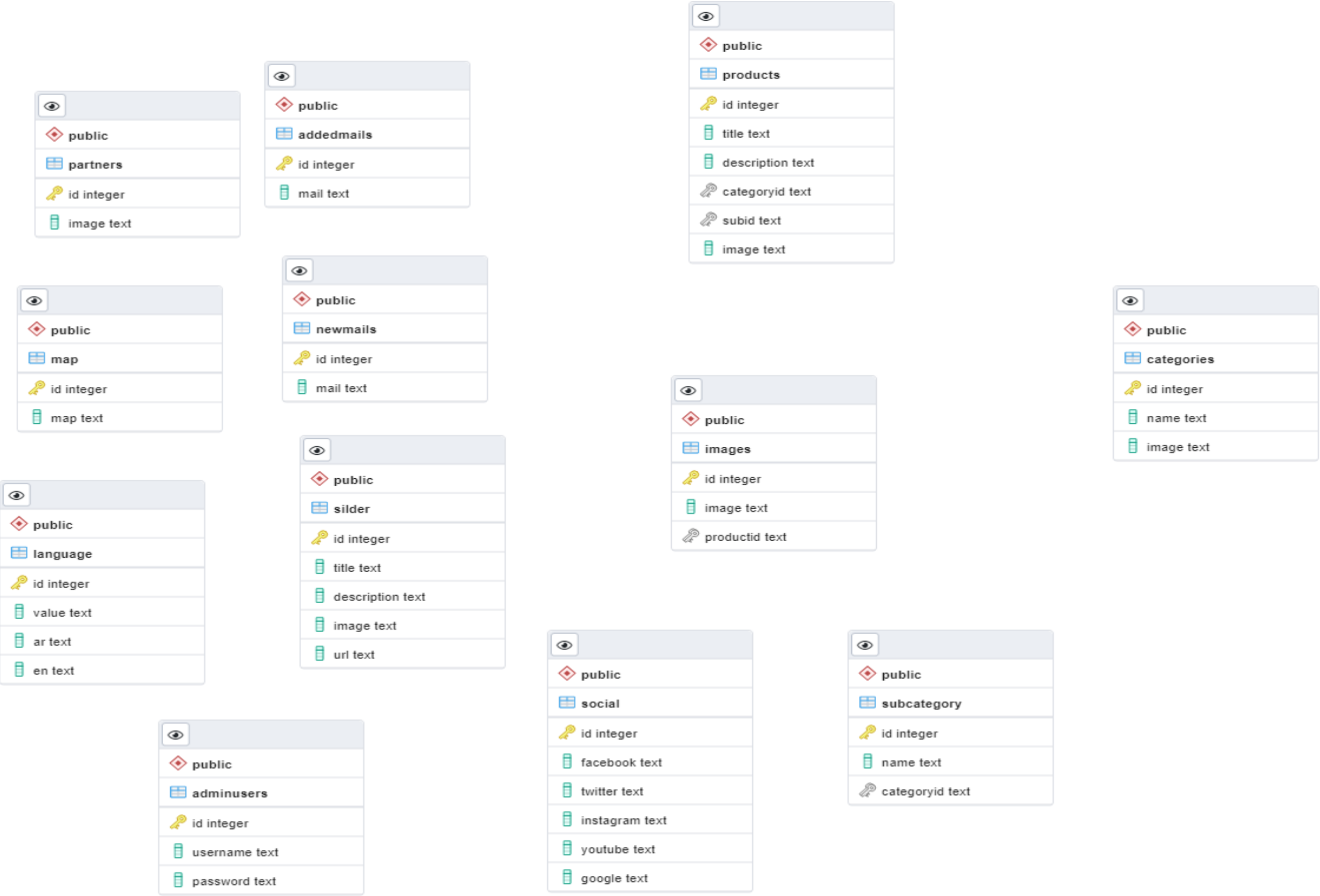
Takipte olmak için mail adresini ekleyebilir

Contact Us kısmından iletişim ve irtibat kurabilir

Ve sosyal medya hesablarına ulaşabilir

Son eklenenler sayıfasını görüntüleyebilir

Varlık Bağıntı Modeli \ ERD :



public
partners
- id integer
- image text

public
addedmails
- id integer
- mail text

public
products
- id integer
- title text
- description text
- categoryid text
- subid text
- image text

public
map
- id integer
- map text

public
newmails
- id integer
- mail text

public
categories
- id integer
- name text
- image text

public
images
- id integer
- image text
- productid text

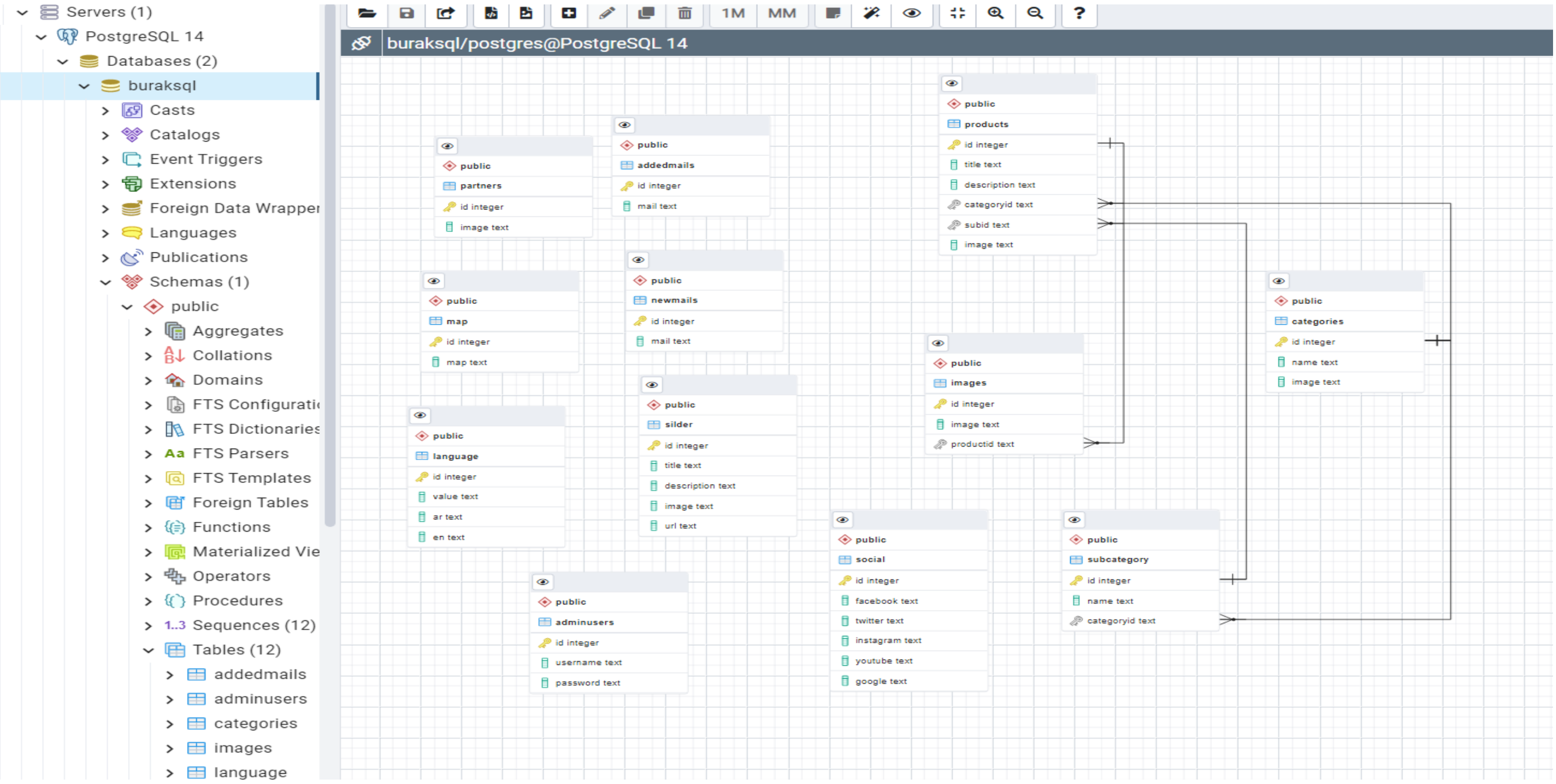
public
language
- id integer
- value text
- ar text
- en text

public
silder
- id integer
- title text
- description text
- image text
- url text

public
social
- id integer
- facebook text
- twitter text
- instagram text
- youtube text
- google text

public
subcategory
- id integer
- name text
- categoryid text

public
adminusers
- id integer
- username text
- password text



Functions:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
```

```
namespace Masra_Group
{
    public class Functions
    {

        public static DataTable TestLogin(string
user ,string pass)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from AdminUsers where UserName ='{0}' and
Password ='{1}'",user,pass));
        }
        public static DataTable GetCategory()
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Categories"));
        }
        public static DataTable GetSocial()
        {
            return MSSQL.ExecSelect(String.Format("select * from Social order by ID DESC"));
        }
        public static DataTable GetSlider()
        {
            return MSSQL.ExecSelect(String.Format("select * from Silder order by ID DESC"));
        }
        public static DataTable GetMap()
        {
            return MSSQL.ExecSelect(String.Format("select * from Map order by ID DESC"));
        }
        public static DataTable GetCategory(int CID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Categories where ID= {0}",CID));
        }
        public static DataTable
GetCategory(String CID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Categories where ID= {0}",CID));
        }
        public static DataTable
GetSliderByID(String ID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Silder where ID= {0}",ID));
        }
        public static int deletecategory(int ID)
        {
            return
MSSQL.ExecQuery(String.Format("DELETE
FROM Categories WHERE ID = {0}",ID));


        }
        public static int deletepartneer(string ID)
        {
            return
MSSQL.ExecQuery(String.Format("DELETE
FROM Partners WHERE ID = {0}", ID));


        }
        public static int deletSlider(string ID)
        {
            return
MSSQL.ExecQuery(String.Format("DELETE
FROM Silder WHERE ID ={0}", ID));
```

```
        }
        public static int deletpost(string ID)
        {
            return
MSSQL.ExecQuery(String.Format("DELETE
FROM Products WHERE ID = {0}", ID));


        }
        public static int deletelang(string
TitleAR)
        {
            return
MSSQL.ExecQuery(String.Format("DELETE
FROM Language where Value
=N'{0}'",TitleAR));


        }
        public static int AddCategory(string
categoryName,string Image)
        {
            return
MSSQL.ExecQuery(String.Format("insert into
Categories (Name,Image) values
(N'{0}',N'{1}')", categoryName,Image));


        }
        public static int AddPartner(string Image)
        {


            return
MSSQL.ExecQuery(String.Format("insert into
Partners (Image) values (N'{0}')",Image));


        }
        public static int UpdateCategory(string
oldcategoyName , string categoryNameAR,
string categoryNameEN ,int id)
        {
```

```
MSSQL.ExecQuery(string.Format("update
Language set Value= N'{0}' , AR= N'{1}' ,
EN= N'{2}' where Value=
N'{3}'",categoryNameAR,categoryNameAR,c
ategoryNameEN,oldcategoyName));
            return
MSSQL.ExecQuery(String.Format("update
Categories set Name=(N'{0}') where ID={1}",
categoryNameAR,id));


        }
        public static int AddLanguage(string
ValueAR,string ValueEN)
        {
            return
MSSQL.ExecQuery(String.Format("insert into
Language (Value ,AR,EN ) values
(N'{0}',N'{1}',N'{2}')",
ValueAR,ValueAR,ValueEN));
```

```
    }

    public static DataTable
GetSubCategory()
    {
      return
MSSQL.ExecSelect(String.Format("select *
from SubCategory order by ID DESC"));
    }
    public static DataTable Getpartners()
    {
      return
MSSQL.ExecSelect(String.Format("select *
from Partners order by ID DESC"));
    }
    public static DataTable GetLang(string
value)
    {
      return
MSSQL.ExecSelect(String.Format("select *
from Language where Value =
N'{0}'",value));
        }
        public static DataTable
GetSubCategory(int ID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from SubCategory where ID={0} order by ID
DESC",ID));
        }
        public static DataTable
GetSubCategoryOrderByCategory(string CID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from SubCategory where CategoryID={0}
order by ID DESC", CID));
        }
        public static DataTable
GetSubCategory(string CID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from SubCategory where ID={0} order by ID
DESC",CID));
        }

        public static int AddSubCategory(string
SubcategoryName ,string categoryid)
        {

            return
MSSQL.ExecQuery(String.Format("insert into
SubCategory (Name,CategoryID) values
(N'{0}',N'{1}')",SubcategoryName,categoryid
));

        }
        public static int
UpdateSubCategory(string
oldSubcategoyName, string
SubcategoryNameAR, string
SubcategoryNameEN, string categoryid ,int
SubCID)
        {



MSSQL.ExecQuery(string.Format("update
Language set Value= N'{0}' , AR= N'{1}' ,
EN= N'{2}' where Value= N'{3}'",
SubcategoryNameAR , SubcategoryNameAR,
SubcategoryNameEN, oldSubcategoyName));
            return
MSSQL.ExecQuery(String.Format("update
SubCategory set Name=(N'{0}')
,CategoryID=(N'{1}') where ID={2}",
SubcategoryNameAR,categoryid,SubCID));


        }
        public static int UpdaLang(string
oldValue, string VAlueAR, string ValueEN)
        {
          return
MSSQL.ExecQuery(string.Format("update
Language set Value= N'{0}' , AR= N'{1}' ,
EN= N'{2}' where Value=
N'{3}'",VAlueAR,VAlueAR,ValueEN,oldVal
ue));
        }
        public static int UpdateSocial(string
Facebook, string Youtube, string Instagram
,string Google ,string Twitter)
        {
          return
MSSQL.ExecQuery(string.Format("update
Social set Facebook= N'{0}' , Twitter = N'{1}'
,Instagram= N'{2}',Youtube= N'{3}',Google=
N'{4}'",Facebook,Twitter,Instagram,Youtube,
Google));
        }
        public static int UpdateMap(string
MapUrl)
        {
           return
MSSQL.ExecQuery(string.Format("update
Map set Map= N'{0}'", MapUrl));
        }
        public static int Updateproduct(string
Title, string Description, string pID)
        {
           return
MSSQL.ExecQuery(string.Format("update
Products set Title =N'{0}',Description= N'{1}'
where ID= {2} ",Title,Description,pID));
        }
        public static int deleteSubcategory(int
ID)
        {
            return
MSSQL.ExecQuery(String.Format("DELETE
FROM SubCategory WHERE ID = {0}",
ID));

        }
        public static DataTable GetProducts()
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Products order by ID DESC"));
        }
        public static DataTable
GetProductsBYCID(string CID)
        {
            return
MSSQL.ExecSelect(String.Format("select top
3* from Products where CategoryID='{0}'
order by ID DESC", CID));
        }
        public static DataTable
GetProducts(string RowNum)
```

```
        {
            return
MSSQL.ExecSelect(String.Format("select
Top {0}* from Products order by ID
DESC",RowNum));
        }
        public static DataTable GetMails()
        {
            return
MSSQL.ExecSelect(String.Format("select *
from NewMails order by ID DESC"));
        }
        public static DataTable GetAddedMails()
        {
            return
MSSQL.ExecSelect(String.Format("select *
from AddedMails order by ID DESC"));
        }
        public static DataTable
GetMailByID(string ID)
```

```
        {
            return
MSSQL.ExecSelect(String.Format("select *
from NewMails where ID={0} order by ID
DESC",ID));
        }
        public static DataTable
GetProductbyID(string ID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Products where ID='{0}' order by ID
DESC",ID));
        }
        public static DataTable
GetProductbySubID(int SubID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Products where SubID ='{0}' order by ID
DESC", SubID));
```

```
        }
        public static DataTable
GetProductbyCategoryORSubID(int CID)
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Products where CategoryID='{0}' order
by ID DESC", CID));
        }
        public static DataTable
GetProductbyCategoryORSubID()
        {
            return
MSSQL.ExecSelect(String.Format("select *
from Products order by ID DESC"));
        }
        public static DataTable
GetProductImagesByID(string ID)
        {
            return
MSSQL.ExecSelect(String.Format("SELECT
Image FROM Products where
Products.ID={0} UNION SELECT  Image
from Images where
Images.ProductID={1}",ID,ID));
        }
        public static int AddPosts(string
Category, string SubCategory, string
TitleAR,string TitleEN ,string DescriptionAR,
string DescriptionEN ,string Image)
        {

MSSQL.ExecQuery(string.Format("insert into
Language (Value ,AR,EN ) values
(N'{0}',N'{1}',N'{2}')",TitleAR,TitleAR,Title
EN));

MSSQL.ExecQuery(string.Format("insert into
Language (Value ,AR,EN ) values
(N'{0}',N'{1}',N'{2}')",DescriptionAR,Descri
ptionAR,DescriptionEN));
```

```
          return
MSSQL.ExecQuery(String.Format("insert into
Products(CategoryID, SubID, Title,
Description, Image) values(N'{0}', N'{1}',
N'{2}',
N'{3}',N'{4}')",Category,SubCategory,TitleA
R,DescriptionAR,Image));


       }
       public static int SetImage(string Image,
string ProductID)
       {
          return
MSSQL.ExecQuery(String.Format("insert into
Images (Image,ProductID) values(N'{0}',
N'{1}')", Image,ProductID));


       }
       public static int SetImail(string mail)
       {
            return
MSSQL.ExecQuery(String.Format("insert into
AddedMails(Mail) values(N'{0}')", mail));


        }
        public static int SetNewImail(string mail)
        {
            return
MSSQL.ExecQuery(String.Format("insert into
NewMails(Mail) values(N'{0}')", mail));


        }
        public static int SetSlider(string TitleAR
,string TitleEN,string DescriptionAR, string
DescriptionEN, string URl , string Image)
        {

MSSQL.ExecQuery(string.Format("insert into
Language (Value ,AR,EN ) values
(N'{0}',N'{1}',N'{2}')", TitleAR, TitleAR,
TitleEN));

MSSQL.ExecQuery(string.Format("insert into
Language (Value ,AR,EN ) values
(N'{0}',N'{1}',N'{2}')", DescriptionAR,
DescriptionAR, DescriptionEN));
            return
MSSQL.ExecQuery(String.Format("insert into
Silder(Title,Description,URL,Image)
values(N'{0}',N'{1}',N'{2}',N'{3}')",
TitleAR,DescriptionAR,URl,Image));


        }
        public static int DeleteImail(string ID)
        {
            return
MSSQL.ExecQuery(String.Format("Delete
NewMails where ID ={0}", ID));


        }
```

```
    }
}
```

SQL İfadeleri :

```
USE [master]
GO
/****** Object:  Database [MasraGroup]    Script Date: 24.08.2022 18:40:43 ******/
CREATE DATABASE [MasraGroup]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'MasraGroup', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\MasraGroup.mdf' , SIZE = 3072KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 1024KB )
 LOG ON
( NAME = N'MasraGroup_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\MasraGroup_log.ldf' , SIZE = 1024KB , MAXSIZE =
2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [MasraGroup] SET COMPATIBILITY_LEVEL = 120
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [MasraGroup].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [MasraGroup] SET ANSI_NULL_DEFAULT OFF
```

```
GO
ALTER DATABASE [MasraGroup] SET ANSI_NULLS OFF
GO
ALTER DATABASE [MasraGroup] SET ANSI_PADDING OFF
GO
ALTER DATABASE [MasraGroup] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [MasraGroup] SET ARITHABORT OFF
GO
ALTER DATABASE [MasraGroup] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [MasraGroup] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [MasraGroup] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [MasraGroup] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [MasraGroup] SET CURSOR_DEFAULT  GLOBAL
GO
ALTER DATABASE [MasraGroup] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [MasraGroup] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [MasraGroup] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [MasraGroup] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [MasraGroup] SET  DISABLE_BROKER
GO
ALTER DATABASE [MasraGroup] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
```

```
ALTER DATABASE [MasraGroup] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [MasraGroup] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [MasraGroup] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [MasraGroup] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [MasraGroup] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [MasraGroup] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [MasraGroup] SET RECOVERY SIMPLE
GO
ALTER DATABASE [MasraGroup] SET  MULTI_USER
GO
ALTER DATABASE [MasraGroup] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [MasraGroup] SET DB_CHAINING OFF
GO
ALTER DATABASE [MasraGroup] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [MasraGroup] SET TARGET_RECOVERY_TIME = 0 SECONDS
GO
ALTER DATABASE [MasraGroup] SET DELAYED_DURABILITY = DISABLED
GO
USE [MasraGroup]
GO
/****** Object:  User [MasraGroup]    Script Date: 24.08.2022 18:40:43 ******/
CREATE USER [MasraGroup] FOR LOGIN [MasraGroup] WITH DEFAULT_SCHEMA=[dbo]
GO
```

```
/****** Object:  Table [dbo].[AddedMails]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AddedMails](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Mail] [nvarchar](max) NULL,
 CONSTRAINT [PK_AddedMails] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[AdminUsers]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AdminUsers](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[UserName] [nvarchar](max) NULL,
	[Password] [nvarchar](max) NULL,
 CONSTRAINT [PK_AdminUsers] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

```
GO
/****** Object:  Table [dbo].[Categories]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categories](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NULL,
	[Image] [nvarchar](max) NULL,
 CONSTRAINT [PK_Categories] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[Images]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Images](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Image] [nvarchar](max) NULL,
	[ProductID] [nvarchar](max) NULL,
 CONSTRAINT [PK_Images] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]


GO

/****** Object:  Table [dbo].[Language]    Script Date: 24.08.2022 18:40:43 ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Language](

	[ID] [int] IDENTITY(1,1) NOT NULL,

	[Value] [nvarchar](max) NULL,

	[AR] [nvarchar](max) NULL,

	[EN] [nvarchar](max) NULL,

 CONSTRAINT [PK_Language] PRIMARY KEY CLUSTERED

(

	[ID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]


GO

/****** Object:  Table [dbo].[Map]    Script Date: 24.08.2022 18:40:43 ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Map](

	[ID] [int] IDENTITY(1,1) NOT NULL,

	[Map] [nvarchar](max) NULL,

 CONSTRAINT [PK_Map] PRIMARY KEY CLUSTERED

(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[NewMails]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[NewMails](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Mail] [nvarchar](max) NULL,
 CONSTRAINT [PK_NewMails] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[Partners]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Partners](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Image] [nvarchar](max) NULL,
 CONSTRAINT [PK_Partners] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[Products]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Products](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Title] [nvarchar](max) NULL,
	[Description] [nvarchar](max) NULL,
	[CategoryID] [nvarchar](max) NULL,
	[SubID] [nvarchar](max) NULL,
	[Image] [nvarchar](max) NULL,
 CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[Silder]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Silder](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Title] [nvarchar](max) NULL,
	[Description] [nvarchar](max) NULL,
	[Image] [nvarchar](max) NULL,
	[URL] [nvarchar](max) NULL,
 CONSTRAINT [PK_Silder] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[Social]    Script Date: 24.08.2022 18:40:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Social](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[Facebook] [nvarchar](max) NULL,
	[Twitter] [nvarchar](max) NULL,
	[Instagram] [nvarchar](max) NULL,
	[Youtube] [nvarchar](max) NULL,
	[Google] [nvarchar](max) NULL,
 CONSTRAINT [PK_Social] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

```sql
GO

/****** Object:  Table [dbo].[SubCategory]    Script Date: 24.08.2022 18:40:43 ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[SubCategory](

	[ID] [int] IDENTITY(1,1) NOT NULL,

	[Name] [nvarchar](max) NULL,

	[CategoryID] [nvarchar](max) NULL,

 CONSTRAINT [PK_SubCategory] PRIMARY KEY CLUSTERED

(

	[ID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]


GO

USE [master]

GOs

ALTER DATABASE [MasraGroup] SET  READ_WRITE

GO
```

Kullanılan fonksiyonlar AboutUS :

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Mail;
using System.Web;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class about : System.Web.UI.Page
    {
        protected DataTable tblmap = new DataTable();
        protected void Page_Load(object sender, EventArgs e)
        {
            tblmap = Functions.GetMap();

        }

        protected void Submit(object sender, EventArgs e)
        {
            try
            {
                MailMessage mail = new MailMessage();
                SmtpClient client = new SmtpClient("relay-hosting.secureserver.net");
                mail.From = new MailAddress("user@masragroup.com");
                mail.To.Add("maha.alturk@gmail.com");
                mail.Subject = "Masra Group User";
                mail.Body = " Name:" + Name.Text + "<br> Email:  " + Email.Text +
"<br> Telephone number: " + Phone.Text + "<br> Country: " + Country.Text + "<br>
Convenience time to call you: " + Time.Text + "<br> Type of service requested:  " +
Messege.Text;
                mail.IsBodyHtml = true;
                client.Port = 25;
                client.Credentials = new
System.Net.NetworkCredential("user@masragroup.com", "mohammad@33");
                client.EnableSsl = false;
                mail.IsBodyHtml = true;
                client.Send(mail);

                client.Dispose();
                Name.Text = "";
                Email.Text = "";
                Phone.Text = "";
                Country.Text = "";
                Time.Text = "";
                Messege.Text = "";
            }
            catch (Exception ex)
            {

                sucses.Text = Masra_Group.Language.getText("notadded");
            }
        }
    }
}
```

Kullanılan fonksiyonlar Kategori :

```
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
```

```
{
    public partial class addcategory : System.Web.UI.Page
    {
        protected DataTable tblCategory = new DataTable();


        protected void Page_Load(object sender, EventArgs e)
        {
            if (Cache["User"] == null)
                Response.Redirect("./admin.aspx");

                tblCategory = Functions.GetCategory();
        }

        protected void Addcategory(object sender, EventArgs e)
        {
            Guid myimgname = Guid.NewGuid();
            string filename = Path.GetFileName(image.FileName);
            image.SaveAs(Server.MapPath("images/" + myimgname + filename));
            string MyImage = "images/" + myimgname + filename;
            Functions.AddCategory(categoryAR.Text,MyImage);
            Functions.AddLanguage(categoryAR.Text,CategoryEN.Text);
            Response.Redirect("./addcategory.aspx");


        }
    }
}
```

Kullanılan fonksiyonlar Mail :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class addmail : System.Web.UI.Page
    {
        protected DataTable tblmail = new DataTable();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Cache["User"] == null)
                Response.Redirect("./admin.aspx");
            tblmail = Functions.GetMails();
        }
    }
}
```

Kullanılan fonksiyonlar Admin :

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class admin : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void Submit(object sender, EventArgs e)
        {
            string email;
            string pass;
            email = MSSQL.temizle(Email.Text);
            pass = MSSQL.temizle(password.Text);
            DataTable temp = new DataTable();
            temp = Functions.TestLogin(email,pass);

            if (temp != null && temp.Rows.Count > 0)
            {
                Cache["User"] = temp.Rows[0]["UserName"].ToString();
                Response.Redirect("./posts.aspx");
            }

            else
            {
                Response.Redirect("/admin.aspx");
            }
        }
    }
}
```

Kullanılan fonksiyonlar Default :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class _default : System.Web.UI.Page
    {
        protected DataTable tblgetslider = new DataTable();
        protected DataTable tblposts = new DataTable();
        protected DataTable tblgetCategory = new DataTable();
        protected DataTable tblmap = new DataTable();
        protected DataTable tblgetpartners = new DataTable();
        protected void Page_Load(object sender, EventArgs e)
        {
           tblgetslider = Functions.GetSlider();
            tblmap = Functions.GetMap();
            tblposts = Functions.GetProducts("4") ;
            tblgetCategory = Functions.GetCategory();
            tblgetpartners = Functions.Getpartners();
        }
        protected void Sendmai(object sender, EventArgs e)
        {
            string mail = MSSQL.temizle(mymail.Text);
            Functions.SetNewImail(mail);
            mymail.Text = "";
        }
    }
}
```

Kullanılan fonksiyonlar Products :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class products : System.Web.UI.Page
    {
        protected DataTable tblgetproducts = new DataTable();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!string.IsNullOrEmpty(Request.QueryString["sub"]))
            {
                string subID = MSSQL.temizle(Request.QueryString["sub"]);
                int sbid = int.Parse(subID);
                tblgetproducts = Functions.GetProductbySubID(sbid);
            }
           else if (!string.IsNullOrEmpty(Request.QueryString["cid"]))
            {string cid = MSSQL.temizle(Request.QueryString["cid"]);
                int CID = int.Parse(cid);

                tblgetproducts = Functions.GetProductbyCategoryORSubID(CID);

            }
            if (string.IsNullOrEmpty(Request.QueryString["cid"]) &&
string.IsNullOrEmpty(Request.QueryString["sub"]))
            {
                tblgetproducts = Functions.GetProductbyCategoryORSubID();

            }




        }
    }
}
```

Kullanılan fonksiyonlar Map :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class setmap : System.Web.UI.Page
    {
        protected DataTable tblgetmap = new DataTable();
        protected void Page_Load(object sender, EventArgs e)
        {
            tblgetmap = Functions.GetMap();
            if (Cache["User"] == null)
                Response.Redirect("./admin.aspx");

            if (!Page.IsPostBack)
            {
                Map.Text = tblgetmap.Rows[0]["Map"].ToString();
            }
            }
        protected void Submit(object sender, EventArgs e)
        {
            if (Page.IsPostBack)
            {
                Functions.UpdateMap(Map.Text);
                Response.Write("<script language='javascript'>alert('Updated
successfully');</script>");
            }
        }
    }
}
```

Kullanılan fonksiyonlar Slider :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;

namespace Masra_Group
{
    public partial class slider : System.Web.UI.Page
    {
        protected DataTable tblslider = new DataTable();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Cache["User"] == null)
                Response.Redirect("./admin.aspx");
            tblslider = Functions.GetSlider();
        }
        protected void Submit(object sender, EventArgs e)
        {
            Guid myimgname = Guid.NewGuid();
            string filename = Path.GetFileName(image.FileName);
            image.SaveAs(Server.MapPath("images/slides/" + myimgname + filename));
            string MyImage = "images/slides/"+ myimgname + filename;
            Functions.SetSlider(TitleAR.Text, TitleEN.Text, DescriptionAR.Text, 
DescriptionEN.Text, URL.Text,MyImage);
            Response.Redirect("./slider.aspx");
        }
    }
}
```

Kullanılan fonksiyonlar subkategori :

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class subcategory : System.Web.UI.Page
    {
        protected static DataTable tblSubCategory = new DataTable();
        protected static DataTable tblCategory = new DataTable();


        protected void Page_Load(object sender, EventArgs e)
        {
            if (Cache["User"] == null)
            {
                Response.Redirect("./admin.aspx");
            }
            else {
            tblSubCategory = Functions.GetSubCategory();
            tblCategory = Functions.GetCategory();

            if (!Page.IsPostBack)
            {
                tblSubCategory = Functions.GetSubCategory();
                SelectSubCategory.DataSource =tblCategory;
                SelectSubCategory.DataTextField = "Name";
                SelectSubCategory.DataValueField = "ID";
                SelectSubCategory.DataBind();
                SelectSubCategory.Items.Insert(0, new ListItem("--Select Name--", 
"0"));

            }
            }
        }
        protected void AddSubcategory(object sender, EventArgs e)
        {
Functions.AddSubCategory(SubcategoryAR.Text,SelectSubCategory.SelectedValue);
            Functions.AddLanguage(SubcategoryAR.Text,SubCategoryEN.Text);
            Response.Redirect("./subcategory.aspx");


        }

    }
}
```

Kullanılan fonksiyonlar Post :

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Masra_Group
{
    public partial class posts : System.Web.UI.Page
    {
        protected DataTable tblProducts = new DataTable();
        protected DataTable tblcategory = new DataTable();
        protected DataTable tblliscategory = new DataTable();
        protected DataTable tblSubcategory = new DataTable();


        protected void Page_Load(object sender, EventArgs e)
        {
            if (Cache["User"] == null)
                Response.Redirect("./admin.aspx");
            if (!Page.IsPostBack)
            {
                tblProducts = Functions.GetProducts();

            }
        }

    }
}
```

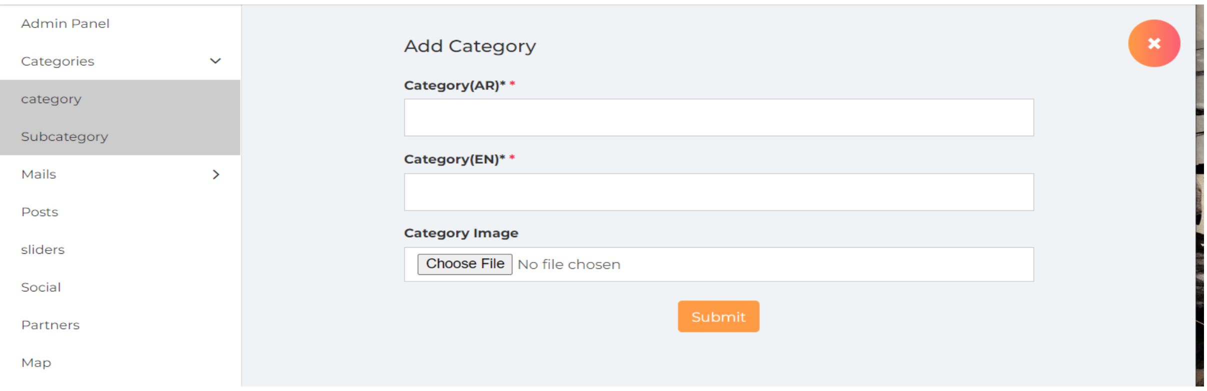
Triggers :

```
CREATE OR REPLACE FUNCTION
partners()
  RETURNS trigger AS
$$
BEGIN
         INSERT INTO partners(col1,col2)

VALUES(NEW.col1,NEW.col2,NEW.col3);


    RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION
products()
  RETURNS trigger AS
$$
```

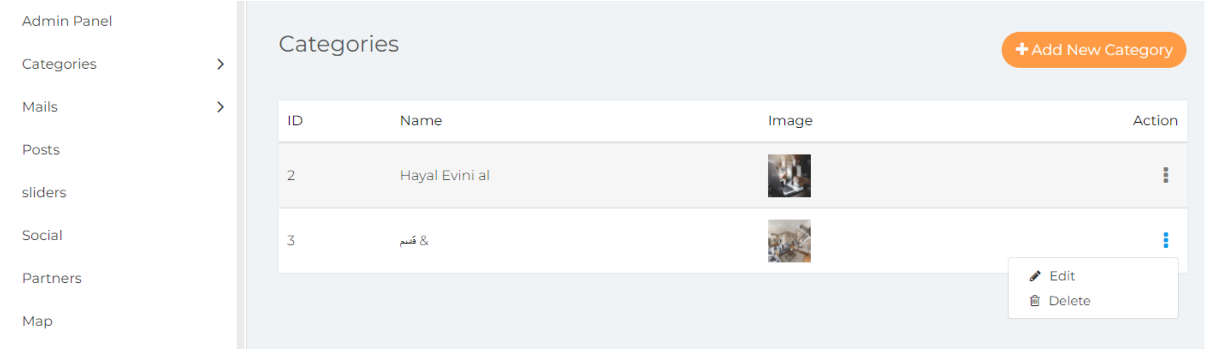
```
BEGIN
        INSERT INTO products(col1,col2,col3)

VALUES(NEW.col1,NEW.col2,NEW.col3);


    RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';
CREATE OR REPLACE FUNCTION
createcategoyr()
  RETURNS trigger AS
$$
BEGIN
        INSERT INTO categories(col1,col2)
        VALUES(NEW.col1,NEW.col2);


    RETURN NEW;
END;
```

```
$$
LANGUAGE 'plpgsql';
CREATE OR REPLACE FUNCTION test()
  RETURNS trigger AS
$$
BEGIN
         INSERT INTO
adminusers(col1,col2,col3)

VALUES(NEW.col1,NEW.col2,NEW.col3);


    RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';
```

Kategori Ekleme :

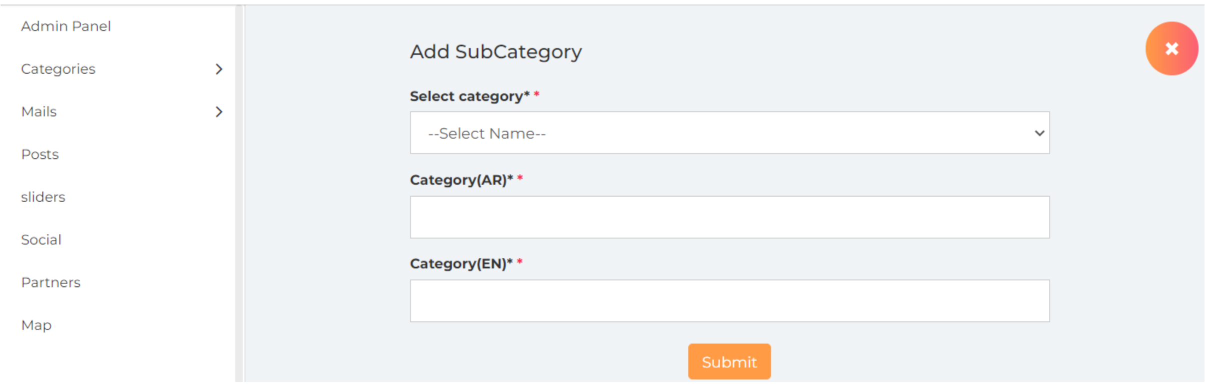


Kategori Silme :

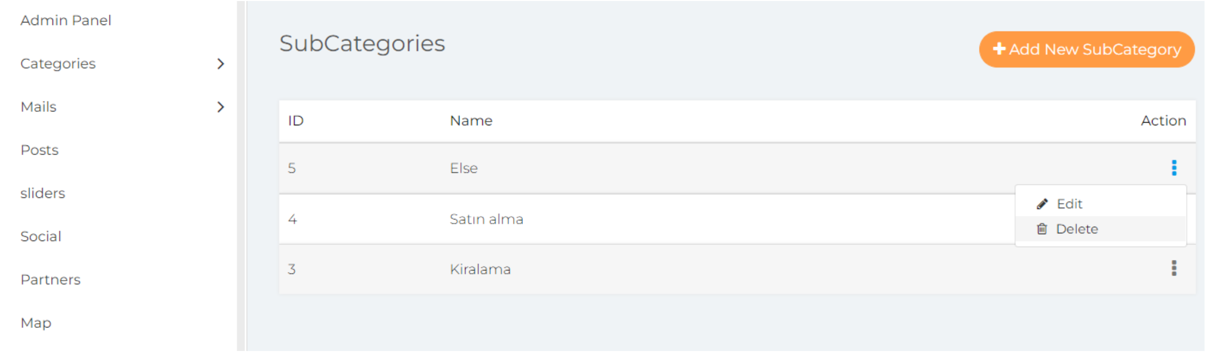


ID	Name	Image	Action
2	Hayal Evini al		⋮
3	قسم &		⋮

Subkategori Ekleme :



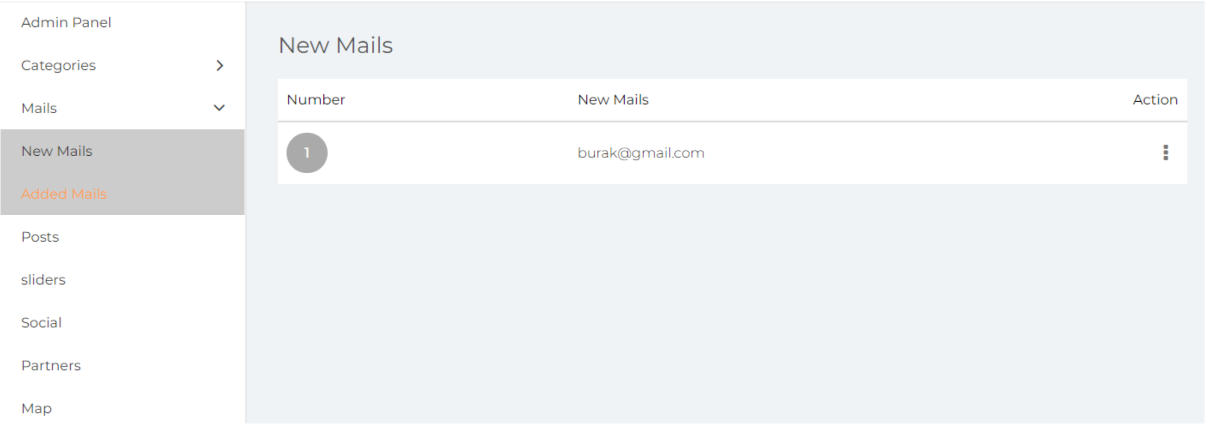
Subkategori Silme :



SubCategories

ID	Name	Action
5	Else	⋮
4	Satın alma	
3	Kiralama	⋮

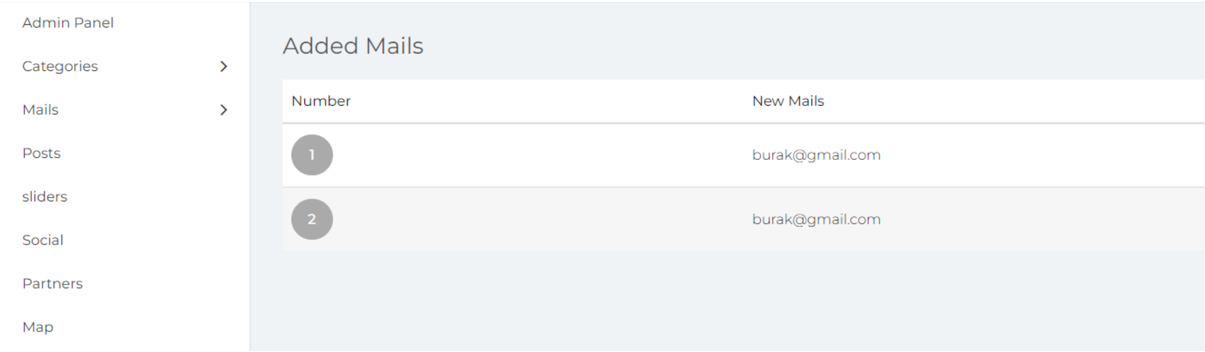
Yeni gelen Mailler :



New Mails

Number	New Mails	Action
1	burak@gmail.com	⋮

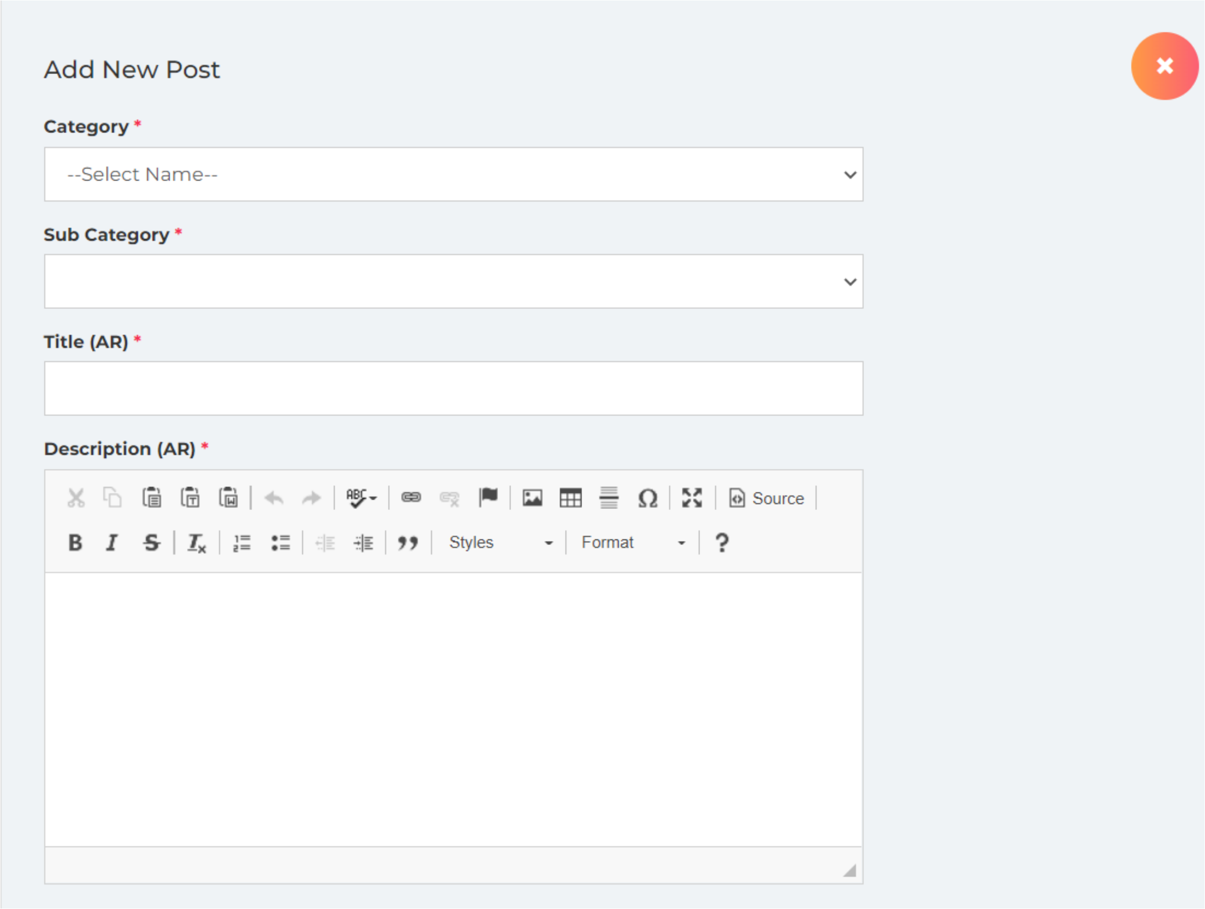
Kayd edilen bütün mailler :



Added Mails

Number	New Mails
1	burak@gmail.com
2	burak@gmail.com

Post Ekleme :



Add New Post

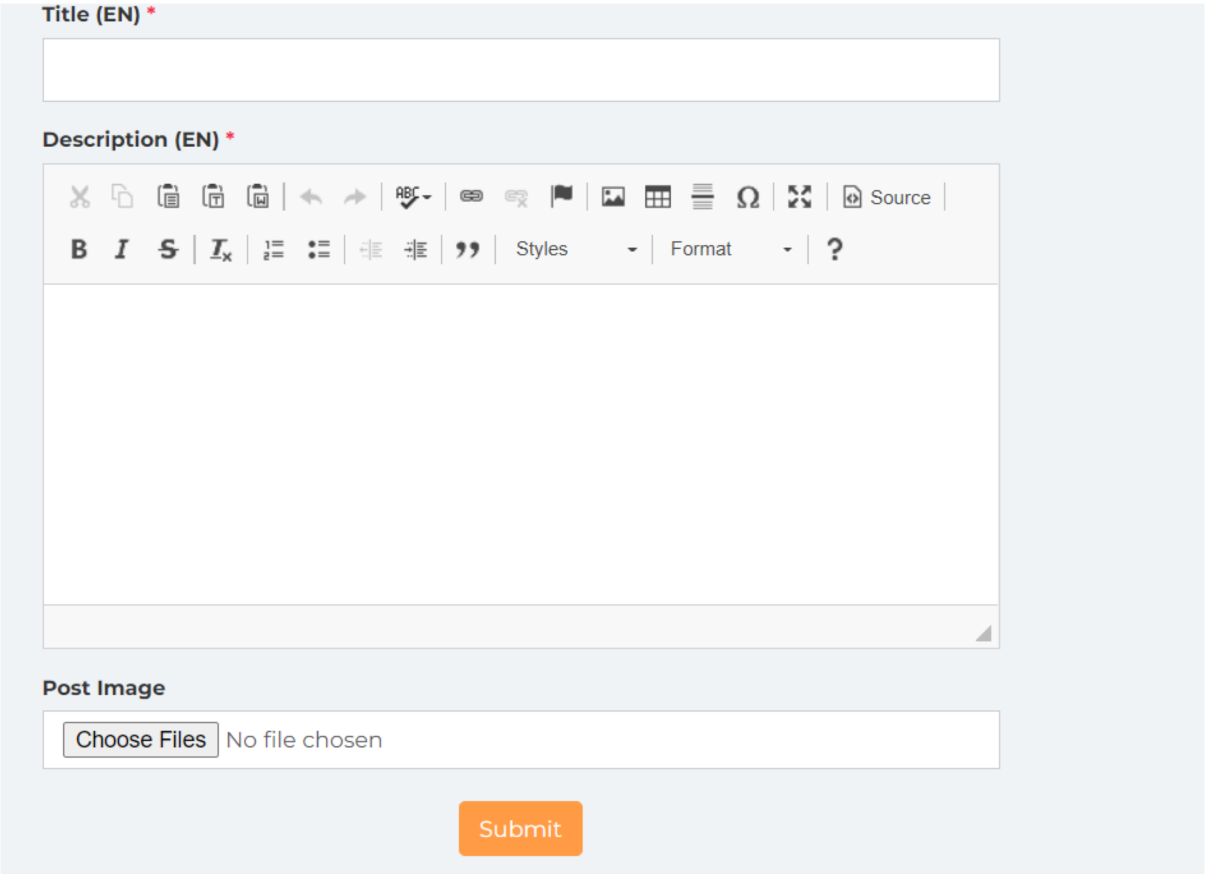
Category *

--Select Name--

Sub Category *

Title (AR) *

Description (AR) *



Title (EN) *

Description (EN) *

Post Image

Choose Files No file chosen

Post Silme :

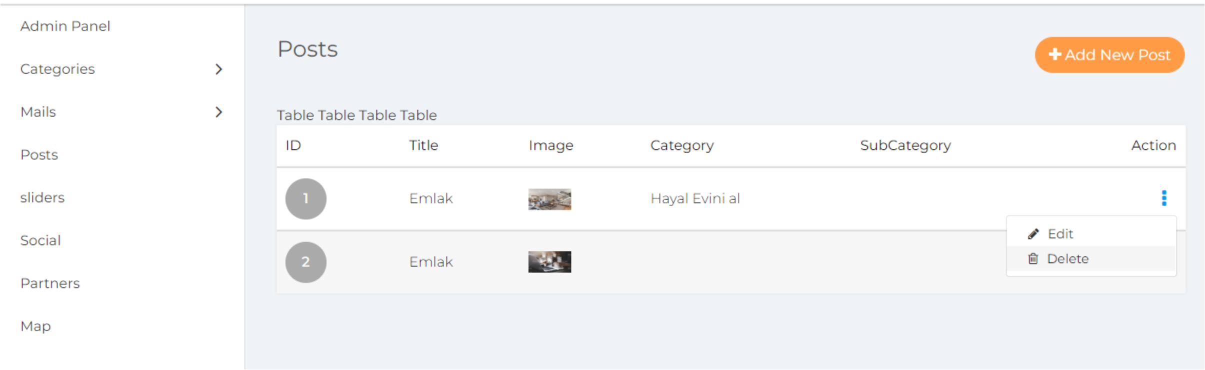
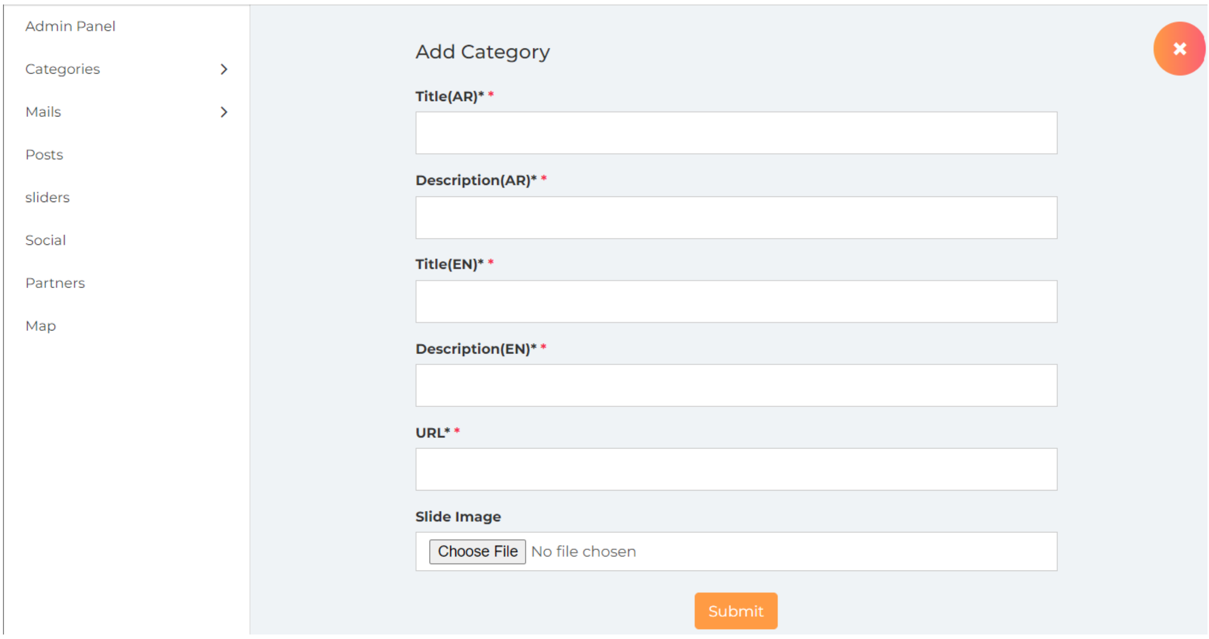


Table Table Table Table

ID	Title	Image	Category	SubCategory	Action
1	Emlak		Hayal Evini al		⋮
2	Emlak				

Silder ekleme :



Add Category

Title(AR)* *

Description(AR)* *

Title(EN)* *

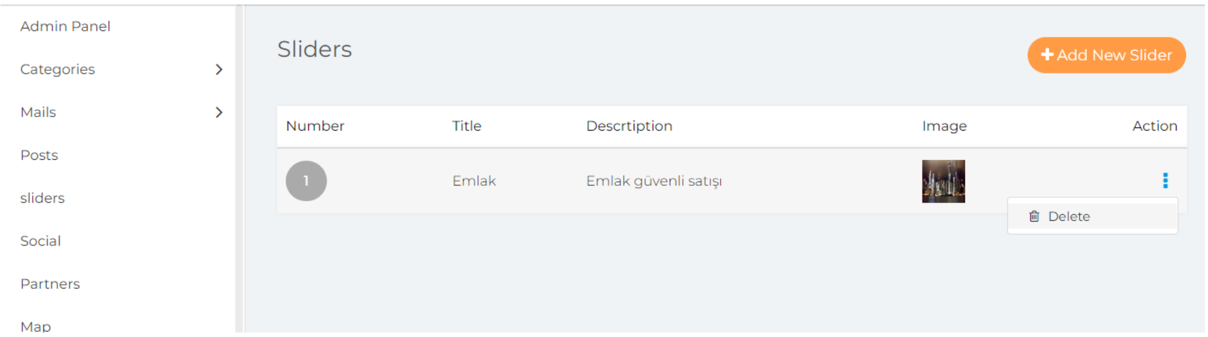
Description(EN)* *

URL* *

Slide Image

No file chosen

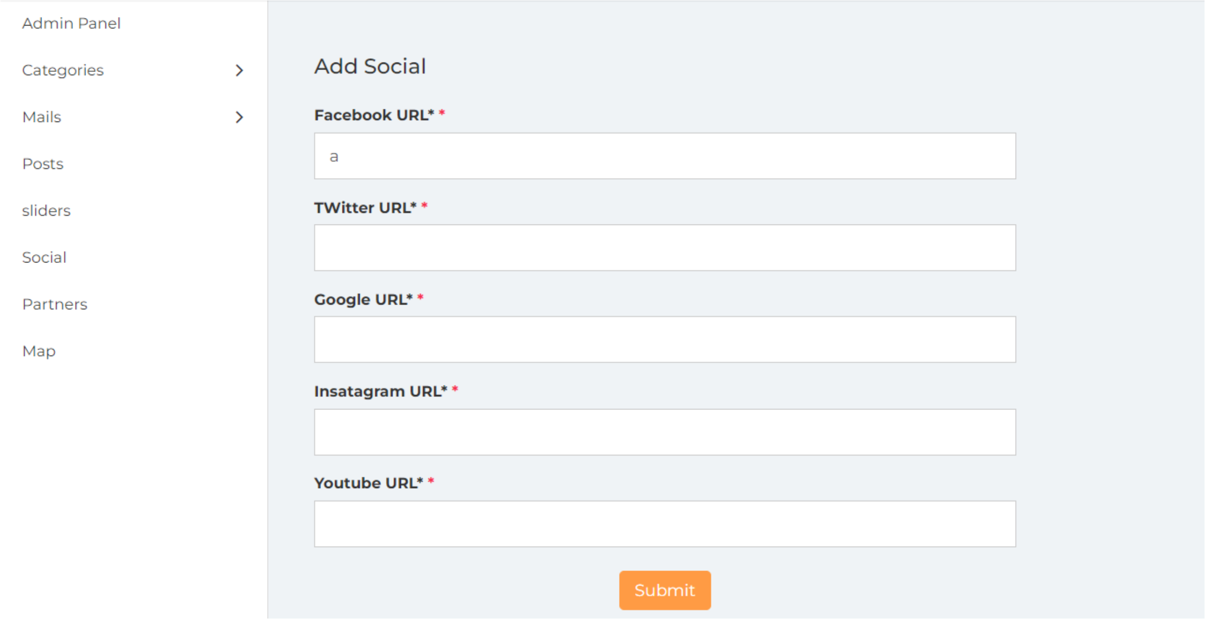
Silder silme :



Sliders

Number	Title	Descrtiption	Image	Action
1	Emlak	Emlak güvenli satışı		

Sosyal medya hesabları ekleme :



Add Social

Facebook URL* *
a

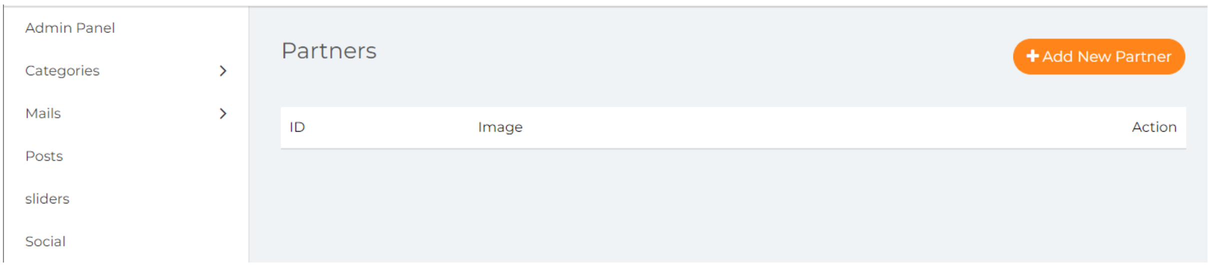
TWitter URL* *

Google URL* *

Insatagram URL* *

Youtube URL* *

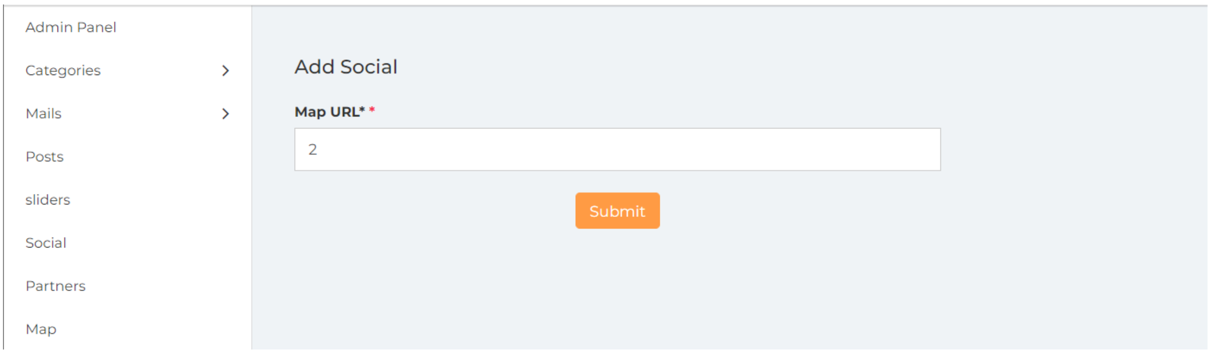
Ortakların file ekleme :



Partners

ID	Image	Action

Harita / Map Ekleme :



Add Social

Map URL* *

2