

Emlak Website

osman sebsebiy

18290746

Proje tanıtımı

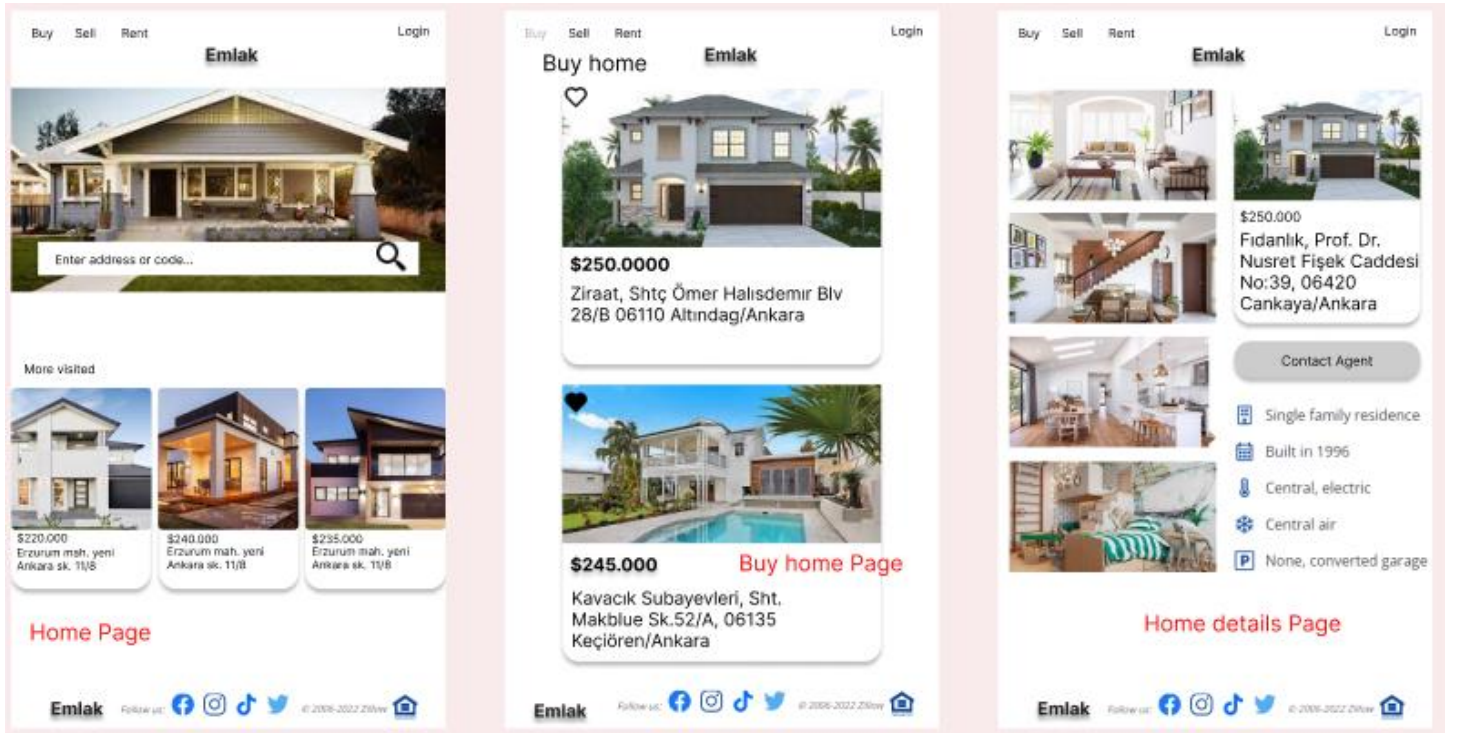
Web sitesi gayrimenkulleri görüntüler ve konut mülkleri ve kiracıların satıcı ve alıcılarına geniş bir hizmet yelpazesi sunar. Ayrıca, kullanıcıların her eve bazı ayrıntıların bir açıklamasını ve bir dizi resim sağlamaları için kolay ve basitleştirilmiş bir arayüz sağlar. Ve satıcılar ve alıcılar arasındaki iletişim sürecini kolaylaştırır.

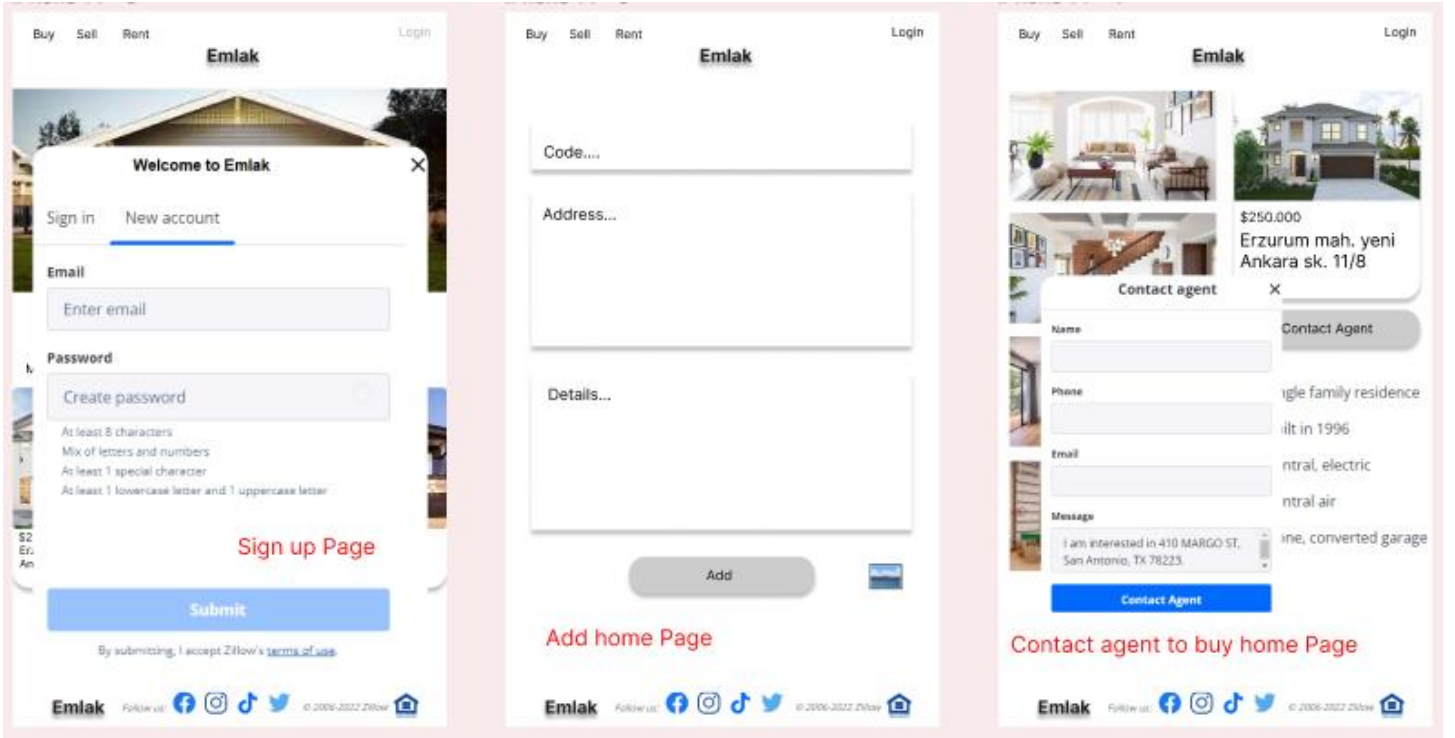
Proje Açıklaması

Siteyi herkes gezebilir ve hizmetlerini görebilir, sadece satmak, satın almak veya kiralamak isteyenlerin hesap oluşturup siteye giriş yapması ve ayrıca hizmet vermek isteyenlerin (kiralık veya satılık ev) hesap oluştur gerekir. . Kullanıcı siteden ev satın almak, satmak veya kiralamak istediğinde, sağlayıcıya bir e-posta göndererek hizmet sağlayıcı (ev sahibi) ile iletişime geçmelidir, burada site hizmet sağlayıcılar için iletişim bilgilerini kaydeder ve ayrıca iletişime geçebilir.

Çalışma planı

1- Figma aracını kullanarak Web Sitesi Sayfaları Tasarlama..





<https://www.figma.com/file/ytep2ukQ0YLbTp8hgSx6tm/Emlak?node-id=14%3A6&t=vpbZCrbx4j1o75Tr-1>

2- Microsoft Sql Server kullanarak veritabanı tasarımı ve yapımı.

Veritabanı şunları içerecektir:

Bir veritabanı oluşturdum ve adını “emlakDB” koydum, aşağıdaki tablolardan oluşuyor:

Ev Tablosu: Ev detaylarını, ev sahibini ve ziyaret sayısını içerir.

	Column Name	Data Type	Allow Nulls
🔑	houseId	int	<input type="checkbox"/>
	houseCode	nvarchar(10)	<input checked="" type="checkbox"/>
	houseAddress	nvarchar(50)	<input checked="" type="checkbox"/>
	housePrice	float	<input checked="" type="checkbox"/>
	houseBuiltDate	int	<input checked="" type="checkbox"/>
	houseHeating	nvarchar(10)	<input checked="" type="checkbox"/>
	houseConditioning	nvarchar(10)	<input checked="" type="checkbox"/>
	houseGarage	nvarchar(10)	<input checked="" type="checkbox"/>
	houseVisitCount	int	<input checked="" type="checkbox"/>
	houseMainImg	nvarchar(MAX)	<input checked="" type="checkbox"/>
	vendorId	int	<input checked="" type="checkbox"/>

Görüntü Tablosu: Evin iç fotoğraflarını içerir.

	Column Name	Data Type	Allow Nulls
🔑	imageld	int	<input type="checkbox"/>
	imagePath	nvarchar(MAX)	<input checked="" type="checkbox"/>
	houseId	int	<input checked="" type="checkbox"/>

Her evin birden fazla fotoğrafı olabilir.

Kullanıcı Tablosu: Kullanıcının oturum açma ve iletişim bilgilerini içerir.

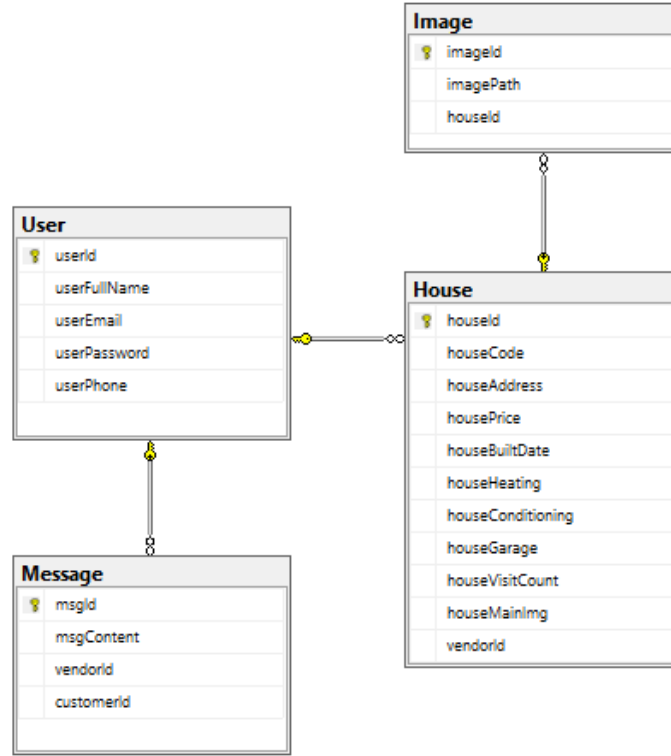
	Column Name	Data Type	Allow Nulls
🔑	userId	int	<input type="checkbox"/>
	userFullName	nvarchar(15)	<input checked="" type="checkbox"/>
	userEmail	nvarchar(15)	<input checked="" type="checkbox"/>
	userPassword	nvarchar(15)	<input checked="" type="checkbox"/>
	userPhone	nvarchar(15)	<input checked="" type="checkbox"/>

Kullanıcı bir satıcı (ev sahibi) veya müşteri (alıcı veya kiracı) olabilir.

Mesaj Tablosu: Bayilere gönderilen mesajları içerir.

	Column Name	Data Type	Allow Nulls
🔑	msgId	int	<input type="checkbox"/>
	msgContent	nvarchar(100)	<input checked="" type="checkbox"/>
	vendorId	int	<input checked="" type="checkbox"/>
	customerId	int	<input checked="" type="checkbox"/>

Veritabanı Seması



3- Asp.net kullanarak web sitesi sayfaları oluşturmak ve uygulamak.

Site, en çok ziyaret edilen evleri gösteren ve kullanıcının satılık, satın alınan veya kiralık evleri görüntüleme arasında seçim yapmasını sağlayan bir liste içeren bir ana sayfadan oluşur.

Seçeneklerden birini seçerek, evlerin listesini görüntüleyen yeni bir sayfa (Ana sayfası) görünür.

Listeden evlerden birini seçerek ev bilgilerinin ve resimlerinin bulunduğu ev detayları sayfasına gidecektir.

Evin sahibiyle iletişime geçmek için, kullanıcının ev sahibi için bir mesaj girmesini sağlayan bir sayfa (İletişim sayfası) görünür.

Ayrıca, iletişim bilgilerini ve giriş bilgilerini içeren bir hesap oluşturmak için bir sayfa vardır.

Ana Sayfa:

[Home](#) [Login](#) [Account](#) [Sell](#) [Rent](#) [Search](#) [About](#) [Contact](#)

Emlak



Most visited



\$22000
Address for third house
R
[Details](#)



\$25000
address
R
[Details](#)



\$1000
Erzurum mah.
R
[Details](#)

© 2023 - Emlak

Satılık/Kiralık Sayfası:

Houses for Sale

2Rah



\$27000
Apaydin / Antakya / Hatay-Turkey
[Details](#)

1SaN



Ayrıntılar Sayfası:



2Rah

\$27000

Apaydın / Antakya /Hatay-Turkey
S

[Contact Agent](#)



Built in 2003



propane

Yeni Hesap Oluştur Sayfası:

Create New ccount

Full Name

Email

Password

Phone

Create

[Back Home](#)

© 2023 - Emlak

Hesap Sayfası:

Account

Full Name Osman
Email osman@gmail.com
Password 000
Phone 098765432

[Inbox](#) | [Sent](#) | [Houses](#) | [Home](#)

© 2023 - Emlak

Gelen Kutusu Sayfası:

Inbox

new

From Ahmad
Date 12/19/2022
Time 10:14 PM
new message
[Reply](#)

Reply :tttitle

From Ahmad
Date 12/19/2022
Time 11:38 PM
message content from Ahmad
[Reply](#)

Arama Sayfası:

SearchForm

HouseModel

Search in the name

Search

[Back to Home](#)

© 2023 - Emlak

Sayfa Hakkında:

[Home](#) [Login](#) [Account](#) [Sell](#) [Rent](#) [Search](#) **[About](#)** [Contact](#)

About.

Emlak Group, Inc., is a Turkish tech real-estate marketplace company that was founded in 2022, and was created by OSMAN SEBSEBIY, web developer.

© 2023 - Emlak

İletişim sayfası:

[Home](#) [Login](#) [Account](#) [Sell](#) [Rent](#) [Search](#) [About](#) **[Contact](#)**

Contact.

Apaydin / Antakya /Hatay-Turkey

P: +905525636787

Support: rromanrr05@gmail.com

Marketing: rromanrr05@gmail.com

© 2023 - Emlak

4- Veritabanına bağlanmak:

Web sitemi veritabanına bağlamak için bir .Net web Api oluşturdum.

Kod Örnekleri:

Modeller:

```
public class HouseModel
{
    //References
    public int houseId { get; set; }
    //1reference
    public string houseCode { get; set; }
    //References
    public string houseAddress { get; set; }
    //References
    public float housePrice { get; set; }
    //References
    public int houseBuiltDate { get; set; }
    //References
    public string houseHeating { get; set; }
    //References
    public string houseConditioning { get; set; }
    //References
    public string houseGarage { get; set; }
    //References
    public int houseVisitCount { get; set; }
    //References
    public string houseMainImg { get; set; }
    //References
    public int vendorId { get; set; }
    //References
    public List<string> images { get; set; }
    //References
    public string houseImg1 { get; set; }
    //References
    public string houseImg2 { get; set; }
    //References
    public string houseImg3 { get; set; }
    //References
    public string houseImg4 { get; set; }
}
```

```

public class UserModel
{
    21 references
    public int userId { get; set; }
    [DisplayName("Full Name")]
    [Required]
    15 references
    public string userFullName { get; set; }
    [DisplayName("Email")]
    [Required]
    14 references
    public string userEmail { get; set; }
    [DisplayName("Password")]
    [Required]
    7 references
    public string userPassword { get; set; }
    [DisplayName("Phone")]
    [Required]
    11 references
    public string userPhone { get; set; }
}

```

HTML Kodu:

Araştırma

```

<h2>SearchForm</h2>

@using (Html.BeginForm("SearchForName", "Home"))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>HouseModel</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <label for="searchPhrase">Search in the name </label>
        <div class="form-group">
            <div class="col-md-10">
                <input type="text" name="searchPhrase" id="searchPhrase" class="form-control"/>
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Search" class="btn btn-default" />
            </div>
        </div>
    </div>
}

```

Kayıt olmak:

```
<div class="form-horizontal">
  <hr />
  @Html.ValidationSummary(true, "", new { @class = "text-danger" })
  <div class="form-group">
    @Html.LabelFor(model => model.userEmail, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.EditorFor(model => model.userEmail, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.userEmail, "", new { @class = "text-danger" })
    </div>
  </div>

  <div class="form-group">
    @Html.LabelFor(model => model.userPassword, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
      @Html.PasswordFor(model => model.userPassword, new { htmlAttributes = new { @class = "form-control" } })
      @Html.ValidationMessageFor(model => model.userPassword, "", new { @class = "text-danger" })
    </div>
  </div>

  <div class="form-group">
    <div class="col-md-offset-2 col-md-10">
      <input type="submit" value="Login" class="btn btn-default" />
    </div>
    <div class="col-md-offset-2 col-md-10">
      <div style="margin-top:10px"> @Html.ActionLink("Create new account" , "NewAccount", "Home")</div>
    </div>
  </div>
</div>
```

C# Kodu:

```
public ActionResult Create()
{
    if (SecurityDAO.loginUser.userId == 0)
    {
        Response.Write("<script>alert('You have to login!')</script>");

        houses = houseDAO.fetchMostVisited();

        return View("Index",houses);
    }
    else
        return View("HouseForm");
}
```

```

public ActionResult Delete(int houseId)
{
    HouseModel house = houseDAO.fetchOne(houseId);
    if (SecurityDAO.loginUser.userId == 0)
    {
        Response.Write("<script>alert('You have to login!')</script>");
        houses = houseDAO.fetchMostVisited();
        return View("Index",houses);
    }
    else
    {
        if (SecurityDAO.loginUser.userId == house.vendorId)
            houseDAO.Delete(houseId);
        else
            Response.Write("<script>alert('You do not have permission to delete!')</script>");

        if (house.houseType == "R")
        {
            List<HouseModel> houses = houseDAO.fetchAllRent();
            return View("Rent", houses);
        }
        else
        {
            List<HouseModel> houses = houseDAO.fetchAllSell();
            return View("Sell", houses);
        }
    }
}
}

```

```

public ActionResult Reply(int msgId)
{
    MessageModel message = new MessageModel();
    msgType = "r";
    message = messageDAO.fetchOne(msgId);

    MessageModel repMessage = new MessageModel();
    UserModel vendor = sec.fetchOne(message.customerId);
    UserModel customer = sec.fetchOne(message.vendorId);
    repMessage.customerName = customer.userFullName;
    repMessage.customerId = message.vendorId;
    repMessage.date = DateTime.Now.ToShortDateString();
    repMessage.time = DateTime.Now.ToShortTimeString();
    repMessage.vendorName = vendor.userFullName;
    repMessage.vendorId = message.customerId;
    repMessage.msgType = "r";
    messageId = msgId;
    repMessage.vendorMobile = vendor.userPhone;
    repMessage.vendorEmail = vendor.userEmail;

    return View("MessageForm", repMessage);
}

```

API Yöntemleri:

```
public HouseModel fetchOne(int id)
{
    List<HouseModel> houses = new List<HouseModel>();

    using (SqlConnection connection = new SqlConnection(SecurityDAO.connectionString))
    {
        string sqlQuery = "SELECT * FROM HOUSE WHERE houseId="+id;
        SqlCommand command = new SqlCommand(sqlQuery, connection); connection.Open();
        SqlDataReader reader = command.ExecuteReader();
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                HouseModel house = new HouseModel();
                house.houseId = reader.GetInt32(0);
                house.houseCode = reader.GetString(1);
                house.houseAddress = reader.GetString(2);
                house.housePrice = Convert.ToSingle(reader.GetDouble(3));
                house.houseBuiltDate = reader.GetInt32(4);
                house.houseHeating = reader.GetString(5);
                house.houseConditioning = reader.GetString(6);
                house.houseGarage = reader.GetString(7);
                house.houseVisitCount = reader.GetInt32(8);
                house.houseMainImg = house.houseId + ".jpg";
                house.houseType = reader.GetString(10);
                house.vendorId = reader.GetInt32(11);
                get images

                houses.Add(house);
            }
        }
    }
    return houses[0];
}
```

```
1 reference
public int updateVisitCount(HouseModel houseModel)
{
    int visitCount = houseModel.houseVisitCount+1;
    using (SqlConnection connection = new SqlConnection(SecurityDAO.connectionString))
    {
        string sqlQuery = "UPDATE HOUSE SET houseVisitCount = " +
            "@houseVisitCount WHERE houseId="+houseModel.houseId;

        SqlCommand command = new SqlCommand(sqlQuery, connection);

        command.Parameters.Add("@houseVisitCount", System.Data.SqlDbType.Int).Value
            = visitCount;

        connection.Open();
        int newID = command.ExecuteNonQuery();
    }
    return houseModel.houseId;
}
```

```

public List<MessageModel> fetchSent(int customerId)
{
    List<MessageModel> messages = new List<MessageModel>();
    using (SqlConnection connection = new SqlConnection(SecurityDAO.connectionString))
    {
        string sqlQuery = "SELECT msgId,msgTitle,msgContent,vendorId,customerId,date," +
            "time,msgType,userFullName FROM MESSAGE,AGENT " +
            "WHERE AGENT.userId=MESSAGE.vendorId And customerId=" + customerId;
        SqlCommand command = new SqlCommand(sqlQuery, connection);
        connection.Open();
        SqlDataReader reader = command.ExecuteReader();
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                MessageModel message = new MessageModel();
                message.msgId = reader.GetInt32(0);
                message.msgTitle = reader.GetString(1);
                message.msgContent = reader.GetString(2);
                message.vendorId = reader.GetInt32(3);
                message.customerId = reader.GetInt32(4);
                message.date = reader.GetString(5);
                message.time = reader.GetString(6);
                message.msgType = reader.GetString(7);
                message.vendorName = SecurityDAO.loginUser.userFullName;
                message.vendorMobile = SecurityDAO.loginUser.userPhone;
                message.vendorEmail = SecurityDAO.loginUser.userEmail;
                message.customerName = reader.GetString(8);
                messages.Add(message);
            }
        }
    }
    return messages;
}

```

Kaynaklar

https://www.youtube.com/playlist?list=PLhPyEFL5u-i1u7EJDbg_yismwCAqLn4PD

<https://www.youtube.com/watch?v=8Wv2Q5Kn58w>

<https://www.tutorialsteacher.com/webapi>

GitHub Bağlantısı

<https://github.com/osmansebsebiy/Emlak.git>