

## BLM 4531 RAPOR

Bu proje kapsamında görev listenizin yapılma aşamasını 3 kısma ayırarak (to do,in progress ve done ) şeklinde iş akışlarının takip edilmesini kolaylaştırmayı amaçladım. Aynı zamanda günlük haberleri api yoluyla kullanıcıya aktararak kullanıcının haberleri takip etmesini sağladım. Bu projeyi iş tanımına uygun olarak hem görevlerin zamansal anlamda takip edilmesini, görev tanımlanmasını ve iş tanımına uygun olarak çıkan önemli makale ve haberleri kullanıcıya ileten bir uygulama yaptım.

Veri Tabanı:

Uygulamam da list, progress, done şeklinde aynı database içerisinde 3 farklı table oluşturdum. Bu tabloların her biri içerisinde ayrı coloumns olarak id,title ve description oluşturdum. Id'yi primary key olarak işaretleyerek ve sequence oluşturarak gelen bir görev için id'yi arttırdım. Böylece gelen her bir özellik için id farklı olacaktır.

BackEnd:

Gelen bilgileri veri tabanına eklemek, silmek, güncellemek için her bir bahsedilen table için ayrı fonksiyonlar tanımladım. Database connection sağladıktan sonra ilgili id üzerinden verileri çekerek üzerinde ekleme ya da silme işlemlerini gerçekleştirdim. Aşağıda örnek olarak edit aşamasının kodunu görebilirsiniz.

```
[HttpPost]
0 başvuru
public ActionResult Edit(List edit_liste)
{
    todoEntities db = new todoEntities();
    List Saved_db = db.List.FirstOrDefault(x => x.id == edit_liste.id);
    Saved_db.title=edit_liste.title;
    Saved_db.description = edit_liste.description;

    db.SaveChanges();
    return RedirectToAction("show_list");
}

[HttpPost]
0 başvuru
public ActionResult Progress_Edit(List edit_liste)
{
    todoEntities db = new todoEntities();
    var Saved_db = db.Progress.FirstOrDefault(x => x.id == edit_liste.id);
    Saved_db.title = edit_liste.title;
    Saved_db.description = edit_liste.description;

    db.SaveChanges();
    return RedirectToAction("show_in_progress");
}
```

Ayrıca tablolar arasında veri geçişini sağlamak amacıyla ayrı fonksiyonlar yazarak iş akış aşamalarında geçiş sağlamayı amaçladım. Örneğin aşağıdaki kodda progress

table aldığım veriyi Done table içerisine eklemek için Done table tipinde bir nesne oluşturarak o nesneyi Done tabelına ekledim.

```
0 basvuru
public ActionResult Send_do_to_done(int id)
{
    todoEntities db = new todoEntities();
    List Send_data = db.List.FirstOrDefault(x => x.id == id);
    var model = new Done { description = Send_data.description, id = Send_data.id, title = Send_data.title };
    db.Done.Add(model);
    db.List.Remove(Send_data);
    db.SaveChanges();
    return RedirectToAction("show_list");
}

0 basvuru
public ActionResult Send_Done(int id)
{
    todoEntities db = new todoEntities();
    var model = db.Progress.FirstOrDefault(x => x.id == id);
    var done_model=new Done {id=model.id,title=model.title,description=model.description};
    db.Done.Add(done_model);
    db.Progress.Remove(model);
    db.SaveChanges();
    return RedirectToAction("show_in_progress");
}

0 basvuru
```

Bu aşamalar arasındaki her bir geçişi iş akış tanımına göre tanımladım.

Api:

Haberlerin ve makalelerin takip edilmesini sağlamak için newsapi.orgdan connection sağlayarak ilgili habeleri json objesi olarak çektim. Daha sonra bu verileri Deserialize

Etmek amacıyla rootjson classını oluşturdum. Dönen veri içerisinde source, id olarak bir json array dönmektedir. Dönen json arrayın içindekiler içinde ayrı class tanımlamam gerekti. Yani bir article classı ve source classlarında oluşturdum. Bu durumda rootjsono classı list tipinde articlar olacak ve bunun yanında id si ve kaynağı da olacaktır. Aşağıda tanımlaması verilmiştir.

```

public class Article
{
    0 başvuru
    public Source source { get; set; }
    0 başvuru
    public string author { get; set; }
    0 başvuru
    public string title { get; set; }
    0 başvuru
    public string description { get; set; }
    0 başvuru
    public string url { get; set; }
    0 başvuru
    public string urlToImage { get; set; }
    0 başvuru
    public DateTime publishedAt { get; set; }
    0 başvuru
    public string content { get; set; }
}

1 başvuru
public class RootJson
{
    0 başvuru
    public string status { get; set; }
    0 başvuru
    public int totalResults { get; set; }
    0 başvuru
    public List<Article> articles { get; set; }
}

1 başvuru
public class Source
{
    0 başvuru
    public string id { get; set; }
    0 başvuru
    public string name { get; set; }
}

```

```

///
var url = "https://newsapi.org/v2/everything?q=cybersecurity&sortBy=popularity&apiKey=97da344f557e483f8e64c69bc0b256e2";

WebClient client = new WebClient();
client.Encoding = Encoding.UTF8;

string json = client.DownloadString(url);

var deserialized = JsonConvert.DeserializeObject<RootJson>(json);

return View(deserialized);

```

Deserialize ettiğim veriyi de ön kısımda her bir makale için ayrı table olacak şekilde ayırarak gösteriyorum. En sonunda ise görünüm şu şekilde oluyor.

Görev Takip Uygulaması						
To do						
In Progress						
Done						
News						
<div><div>Başlık Bilgisi</div><div>Engadget</div><div>LockBit ransomware gang apologizes for SickKids hospital attack and offers free decryptor</div><div>Açıklama Bilgisi</div><div>One of the world's most notorious ransomware gangs has issued a rare apology after claiming that one of its partners was responsible for a cyberattack on Canada's largest pediatric hospital. On December 18th, the Hospital for Sick Children (SickKids) in Tor...</div><div>Yazar</div><div>Igor Bonifacio</div><div>İçerik</div><div>One of the worlds most notorious ransomware gangs has issued a rare apology after claiming that one of its partners was responsible for a cyberattack on Canadas largest pediatric hospital. On Decem... [+1935 chars]</div><div>Uri</div><div>https://www.engadget.com/lock-bit-ransomware-gang-apologizes-sick-kids-attack-224245439.html</div></div>						
<div><div>Başlık Bilgisi</div><div>Wired</div><div>The Worst Hacks of 2022</div><div>Açıklama Bilgisi</div><div>The year was marked by sinister new twists on cybersecurity classics, including phishing, breaches, and ransomware attacks.</div><div>Yazar</div><div>Lily Hay Newman</div><div>İçerik</div><div>With the pandemic evolving into an amorphous new phase and political polarization on the rise around the world, 2022 was an uneasy and often perplexing year in digital security. And while hackers fre... [+7901 chars]</div><div>Uri</div><div>https://www.wired.com/story/worst-hacks-2022/</div></div>						
<div><div>Başlık Bilgisi</div><div>Wired</div></div>						

Tablo kesitleri gelen her bir makale için ayrılıyor.

FrontEnd:

Görev Takip Uygulaması

To do

In Progress

Done

News

To Do

Add New Item

Görev Listeniz

ID	Başlık	Açıklama	Düzenle	Send to In progress	Sil	Send to Done
33	Okul Ödevleri	Bitirme Tezi				
34	Yarışma	CTF write up ekleme				

MVC kullanarak yukarıdaki gibi button ve model tanımlamalarımı yaptım. Add new item diyerek aşağıdaki card yapısı içerisinde bilgilerinin kaydedilmesini sağlıyorum.

Görev Takip Uygulaması

To do

In Progress

Done

News

To Do

Add New Item

Görev Listeniz

ID	Başlık
33	Okul Öd
34	Yarışma

Yeni Görev Bilgileri

Başlık Bilgisi

Açıklama Bilgisi

Kapat

Bilgileri Kaydet

Progress ve Done tabela atılacağı zaman ise aşağıdaki gibi bir card modeli tanımladım.

Görev In Progresse aktarılacak.Son Düzenleme Yapılandırması yapılabilir.

**Başlık Bilgisi**

Okul Ödevleri

**Açıklama Bilgisi**

Bitirme Tezi

Kapat Görevi In Progresse At

Görev kalıcı olarak silineceğinde ise aşağıdaki gibi danger modunda bir card yapısı ile gösterimi sağladım.

Görev Kalıcı Olarak Silinecek

**Başlık Bilgisi**

Okul Ödevleri

**Açıklama Bilgisi**

Bitirme Tezi

Kapat Görevi Sil

Görev Düzenleme ekranında ise aşağıdaki gibi bir editleme yaparak database kaydettim.

Bilgileri Düzenle

**Başlık Bilgisi**

Yarışma

**Açıklama Bilgisi**

CTF write up ekleme Son Düzenleme

Close Save changes

Aşağıda da örnek olarak to do ekranının front end kısmını görebilirsiniz. List tabelında ilgili verileri viewbag yapısına daha önceden atayarak verileri bu şekilde alabiliyoruz.

```
<div class="row">
  <div class="col-lg-12">
    <div class="card">
      <div class="card-header"><i class="fa fa-list"></i> Görev Listeniz</div>
      <div class="card-body">
        <div class="table-responsive">
          <table id="example" class="table table-striped table-hover table-bordered">
            <thead>
              <tr>
                <th scope="col">ID</th>
                <th scope="col">Başlık</th>
                <th scope="col">Açıklama</th>
                <th scope="col">Düzenle</th>
                <th scope="col">Send to In progress</th>
                <th scope="col">Sil</th>
                <th scope="col">Send to Done</th>
              </tr>
            </thead>
            <tbody>
              <@foreach (ToDo.Models.List item in ViewBag.List)>
                <tr>
                  <td scope="row">@item.id</td>
                  <td scope="row">@item.title</td>
                  <td scope="row">@item.description</td>
                  <td scope="row">
                    <button class="btn btn-warning" data-toggle="modal" data-target="#@item.id">
                      <i class="fa fa-edit"></i>
                    </button>
                    <div class="modal fade" id="@item.id">
                      <div class="modal-dialog modal-lg">
                        <form action="/Home/Edit/" method="post">
                          <div class="modal-content border-info">
                            <div class="modal-header bg-info">
                              <h3 class="modal-title text-white">Bilgileri Düzenle </h3>
                              <button type="button" class="close text-white" data-dismiss="modal" aria-label="Close">
                                <span aria-hidden="true"><i></i></span>
                              </button>
                            </div>
                            <div class="modal-body">
                              <input type="text" name="id" value="@item.id" hidden />
                              <div class="form-group col-lg-12">
                                <label>Başlık Bilgisi</label>
                                <input type="text" class="form-control" value="@item.title" name="Title">
                              </div>
                              <div class="form-group col-lg-12">
                                <label>Açıklama Bilgisi</label>
                                <textarea rows="4" class="form-control" id="basic-textarea" name="Description">@item.description</textarea>
                              </div>
                            </div>
                            <div class="modal-footer">
                              <button type="button" class="btn btn-dark" data-dismiss="modal"><i class="fa fa-times"></i> Close </button>
                              <button type="submit" class="btn btn-info"><i class="fa fa-save"></i> Save changes</button>
                            </div>
                          </div>
                        </form>
                      </div>
                    </td>
                </tr>
              </tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```