**NESNE YÖNELİMLİ YAZILIM MÜHENDİSLİĞİ DERSİ**

**ANALİZ RAPORU**

**Projenin Adı :** İNÖNÜ BLOG (İBMB)

**Hazırlayanlar :** Emine GÜL

Gülistan DALBUDAK

Rıdvan ZUGURLİ

Ahmet Cihat ÖZBEY

Zeyneddin ÖZ

Esengül BAYAZİT

Aslıhan TUNA

**Projenin Amacı**

InonuBlog.com; İnönü Üniversitesi Bilgisayar Mühendisleri öğrencilerinin ve adayların online ortamda bir araya gelmesini sağlayarak öğrencilerin teknolojik haberlerden ve yazılımsal konularda kaynak bulmasını sağlayan bir blog sitesidir.

Kullanımı kolay, hızlı altyapısı ve sadece İnönü Üniversitesi Bilgisayar Mühendisleri Öğrencilerine hitap etmeyerek; üniversite öğrencileri, teknoloji meraklıları ve yazılım eğitimcilerinin ihtiyacına yönelik oluşu ile diğer blog sitelerinden sıyrılıp farkını ortaya koymak istemektedir.

InonuBlog.com olarak hedefimiz daha geniş kitlelere ulaşarak blog sitemizin büyümesine katkı sağlamaktır.

**Projenin Analizi**

Ürün olarak ortaya çıkartmak istediğimiz “İNÖNÜ BLOG (İBMB)” projesi, bir makale paylaşım sitesidir. Web sitemizde, birden çok admin bulunmaktadır. Üye id’si 1 olanlar admin, üye id’si 2 olanlar normal kullanıcılardır. Kullanıcıların adı, soyadı ve fotoğraf bilgileri Üye Profil Sayfasından görülebilmektedir.

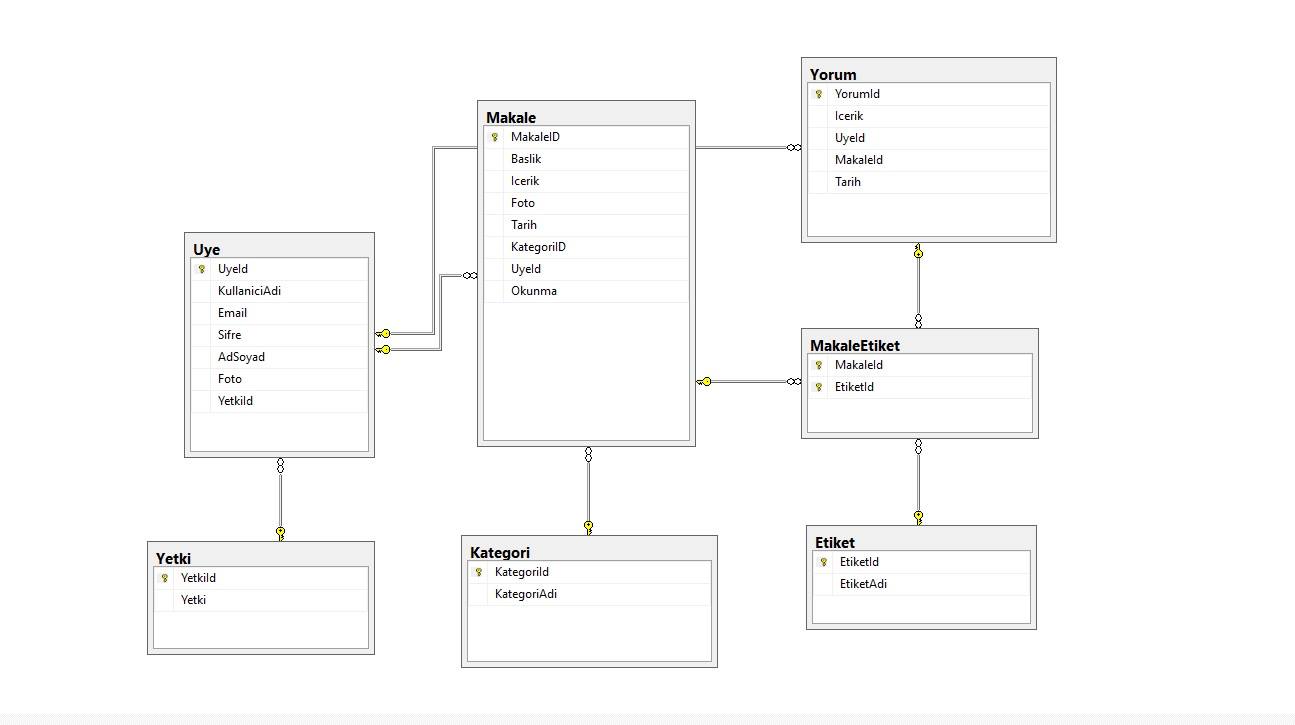
Projemizde makalelerin kategorileri bulunmaktadır. Hangi kategoride kaç makale yayınlandığını görebilmekteyiz. En çok okunan makaleleri okunma sayılarına göre çekebilmekteyiz. Son yapılan 5 yorum sayfada görülmektedir. Site içinde rahatlıkla arama yapmamızı sağlayan bir arama butonu bulunmaktadır. Makalelerin, okunma sayısı, yorum sayısı, hangi kategoride olduğu ve yayınlanma tarihi bilgileri bulunmaktadır.

Makalelere yorum yapabilmek için üye olmak gerekmektedir. Kayıt olma işlemi gerçekleştikten sonra giriş yaparak yorum yapılabilmektedir. Yorumumuzu yazıp yorum yap dediğimizde, site “Yorumunuz gönderildi…” mesajını vermektedir. Kullanıcılar kendi yaptıkları yorumları, yorumlarının altında bulunan sil butonu sayesinde silebilmektedir. Kullanıcılar profil sayfalarından bilgilerini güncelleyebilmektedir.

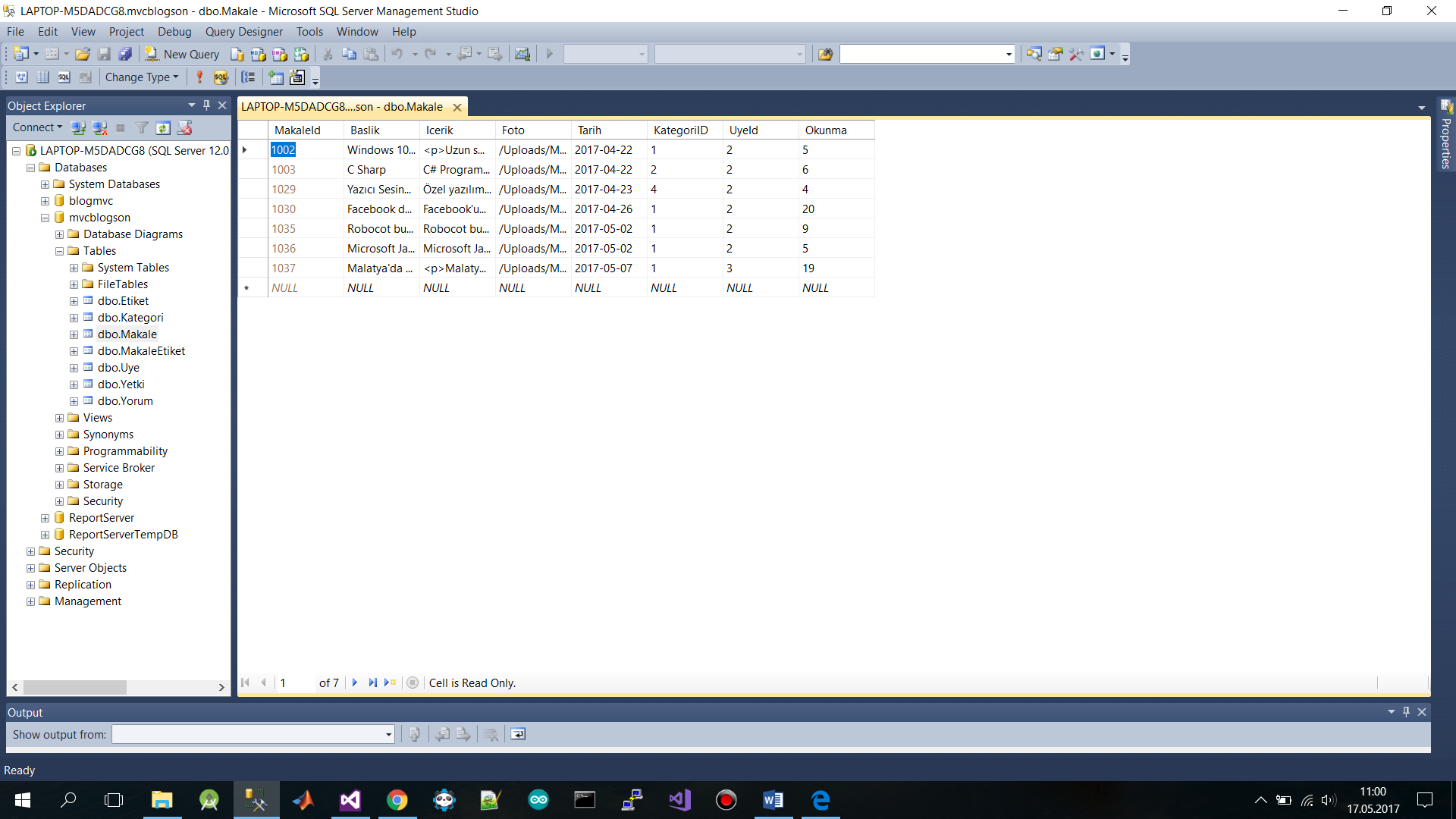
Yalnızca admin olan kişiler admin paneline erişebilmektedir. Adminler, admin panelinde kategorileri yönetebilmektedir. Yani yeni kategori oluşturabilir, var olan kategoriyi düzenleyebilir veya var olan kategoriyi silebilir.

Yeni bir makale eklemek istendiğinde öncelikle kategori seçilmelidir. Daha sonra bir makale başlığı eklenmelidir. Yazı tipi vs. düzenlenebilmektedir. Makale için fotoğraf ve etiketler eklenip kaydet denildiğinde makale eklenmektedir. Makale eklendikten sonra her tıklandığında okunma sayısı artmaktadır. Eklenen makaleler daha sonra düzenlenebilmektedir. Detay butonuna basıldığında makalenin detayı görülmektedir. Sil butonuna basıldığında da makale silinmektedir.

**Veri Tabanı Diyagramı ve Açıklamaları**



Veritabanımız 7 tablodan oluşmaktadır. Blog sayfamızda kullanıcılar 2 türlüdür. Admin olan üyeler ve sadece üyeler. Yetki tablomuzdaki YetkiId si 1 olan kişiler Adminlerdir. 2 olanlar ise sadece üye olan kullanıcılardır. Sadece Admin kullanıcılara admin paneli açılmaktadır. Admin olmayan üyeler saddece makalelere yarum yapabilmektedeir. Uyelerin KullanıcıAdi, Email, Sifre, bilgileri sisteme giriş için kullanılmaktadır. Diğer bilgiler ise kullanıcı profil sayfasında gösterilmek için kullanılmaktadır.



Makalelerin tablosu MakaleId, Baslik, Icerik, Foto, Tarih,KategoriID,UyeId,Okuma bilgileirni tutmaktadır. Bu bilgiler makale detay ve admin makale yönetim panelinde gösterilmektedir.

MakaleId leri otomatik arttırılmakta ve Okunma sayısı da ilk makale eklendiğinde default olarak 1 kaydedilmekte. Makale detayı sayfası her çağrıldığında otomatik olarak arttırılmaktadır.

Makale etiket ve Etiket tablosu Google SEO için tasarlanmıştır. Her makalenin birden fazla etiketi olmaktadır.

**Methodlar ve Açıklamaları**

HomeController : Controller

{

mvcblogDB db = new mvcblogDB();//database nesnesi oluşturuyoruz

// GET: Home

public ActionResult Index(int Page=1)//1.Sayafadan başlıyoruz makale listelemeye

{

var makale = db.Makales.OrderByDescending(m => m.MakaleId).ToPagedList(Page,5);

//Her sayfada 5 makale listeleniyor.

return View(makale);

}

public ActionResult BlogAra(string Ara=null)

{//Blogda arama yapılınca başlıklar içinde arama yapılıyor

var aranan = db.Makales.Where(m => m.Baslik.Contains(Ara)).ToList();

//Başlık içinde aranan varsa tarihe göre sıralı getirir.

return View(aranan.OrderByDescending(m=>m.Tarih));

}

public ActionResult SonYorumlar()

{//Son 5 yorum anasayfada listleniyor.

return View(db.Yorums.OrderByDescending(y=>y.YorumId).Take(5));

}

public ActionResult PopulerMakaleler()

{//Okunma sayısına göre en çok okunan 5 makale anasayfada sıralanıyor

return View(db.Makales.OrderByDescending(m => m.Okunma).Take(5));

}

public ActionResult KategoriMakale(int id)

{ //Kategoriye tıklanınca tüm makaleleri listeliyor

var makaleler = db.Makales.Where(m => m.Kategori.KategoriId == id).ToList();

return View(makaleler);

}

public ActionResult MakaleDetay(int id)

{//Makaleye tıklanınca makale detayına gidiyor.

var makale = db.Makales.Where(m => m.MakaleId == id).SingleOrDefault();

if (makale == null)

{

return HttpNotFound();

}

return View(makale);

}

public ActionResult Hakkimizda()

{

return View();

}

public ActionResult Iletisim()

{

return View();

}

public ActionResult KategoriPartial()//Kategori listesi oluşturuluyor

{

return View(db.Kategoris.ToList());

}

public JsonResult YorumYap(string yorum,int Makaleid)

{

var uyeid = Session["uyeid"];

if (yorum == null)

{

return Json(true, JsonRequestBehavior.AllowGet);

}

db.Yorums.Add(new Yorum { UyeId = Convert.ToInt32(uyeid), MakaleId = Makaleid, Icerik = yorum, Tarih = DateTime.Now });

db.SaveChanges();

return Json(false, JsonRequestBehavior.AllowGet);

}

public ActionResult YorumSil(int id)

{//

var uyeid = Session["uyeid"];

var yorum = db.Yorums.Where(y => y.YorumId == id).SingleOrDefault();

var makale = db.Makales.Where(m => m.MakaleId == yorum.MakaleId).SingleOrDefault();

//Yoruma yapan üye kendi yorumunu silebilir

if (yorum.UyeId == Convert.ToInt32(uyeid))

{

db.Yorums.Remove(yorum);

// Değişiklikler veritabanına kaydediliyor

db.SaveChanges();

return RedirectToAction("MakaleDetay", "Home",new {id=makale.MakaleId});

}

else

{

return HttpNotFound();

}

}

public ActionResult OkunmaArttir(int Makaleid)

{//Makaleye her tıklandığında veritabanında makale ounma sayısını arttırıyor.

var makale = db.Makales.Where(m => m.MakaleId == Makaleid).SingleOrDefault();

makale.Okunma += 1;

db.SaveChanges();

return View();

AdminController : Controller

ViewBag ile verileri çekip sayıyoruz.

ViewBag.MakaleSayisi = db.Makales.Count();

ViewBag.YorumSayisi = db.Yorums.Count();

ViewBag.KategoriSayisi = db.Kategoris.Count();

ViewBag.UyeSayisi = db.Uyes.Count();

AdminKategoriController : Controller

public ActionResult Details(int? id)// Kategorilerin id'sine göre detayına bakıyoruz.

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Kategori kategori = db.Kategoris.Find(id);

if (kategori == null)

{

return HttpNotFound();

}

return View(kategori); //Kategorileri döndürüyoruz.

}

public ActionResult Create()

{

return View();

}

// POST: AdminKategori/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "KategoriId,KategoriAdi")] Kategori kategori)

{

if (ModelState.IsValid)

{

db.Kategoris.Add(kategori);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(kategori);

}

// GET: AdminKategori/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Kategori kategori = db.Kategoris.Find(id);

if (kategori == null)

{

return HttpNotFound();

}

return View(kategori);

}

// POST: AdminKategori/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "KategoriId,KategoriAdi")] Kategori kategori)

{

if (ModelState.IsValid)

{

db.Entry(kategori).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(kategori);

}

// GET: AdminKategori/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Kategori kategori = db.Kategoris.Find(id);

if (kategori == null)

{

return HttpNotFound();

}

return View(kategori);

}

// POST: AdminKategori/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Kategori kategori = db.Kategoris.Find(id);//Kategori id yi bulup kategoriden de makaleyi siliyor.

db.Kategoris.Remove(kategori);

db.SaveChanges();

return RedirectToAction("Index");

}

AdminMakaleController : Controller

public ActionResult Index(int Page=1)//Pageing işlemi yapılıyor

{

var makales = db.Makales.OrderByDescending(m => m.MakaleId).ToPagedList(Page, 10);

return View(makales);

} public ActionResult Details(int id)

{

return View();

}

public ActionResult Create()

{

ViewBag.KategoriId = new SelectList(db.Kategoris, "KategoriId", "KategoriAdi");

return View();

}

// POST: AdminMakale/Create

[HttpPost]

public ActionResult Create(Makale makale,string etiketler, HttpPostedFileBase Foto)

{

if(ModelState.IsValid)//Model doğrulanıyor.

{

if (Foto != null)

{

WebImage img = new WebImage(Foto.InputStream);

FileInfo fotoinfo = new FileInfo(Foto.FileName);

string newfoto = Guid.NewGuid().ToString() + fotoinfo.Extension;

img.Resize(800,350);

img.Save("~/Uploads/MakaleFoto/"+ newfoto);

makale.Foto = "/Uploads/MakaleFoto/" + newfoto;

}

if (etiketler != null)

{

string[] etiketdizi = etiketler.Split(',');//etiketler arasına token olarak virgül koyuluyor.

foreach (var i in etiketdizi)

{

var yenietiket = new Etiket { EtiketAdi = i };

db.Etikets.Add(yenietiket);

makale.Etikets.Add(yenietiket);

}

}

makale.UyeId = Convert.ToInt32(Session["uyeid"]);

makale.Okunma = 1;

db.Makales.Add(makale);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(makale);

}

// GET: AdminMakale/Edit/5

public ActionResult Edit(int id)

{

var makale = db.Makales.Where(m => m.MakaleId == id).SingleOrDefault();

if (makale == null)

{

return HttpNotFound();

}

ViewBag.KategoriId = new SelectList(db.Kategoris, "KategoriId", "KategoriAdi", makale.KategoriId);

return View(makale);

}

// POST: AdminMakale/Edit/5

[HttpPost]

public ActionResult Edit(int id, HttpPostedFileBase Foto, Makale makale)

{

try

{

var makales = db.Makales.Where(m => m.MakaleId == id).SingleOrDefault();

if (Foto != null)

{

if (System.IO.File.Exists(Server.MapPath(makales.Foto)))

{

System.IO.File.Delete(Server.MapPath(makales.Foto));

}

WebImage img = new WebImage(Foto.InputStream);

FileInfo fotoinfo = new FileInfo(Foto.FileName);

string newfoto = Guid.NewGuid().ToString() + fotoinfo.Extension;

img.Resize(800, 350);

img.Save("~/Uploads/MakaleFoto/" + newfoto);

makales.Foto = "/Uploads/MakaleFoto/" + newfoto;

makales.Baslik = makale.Baslik;

makales.Icerik = makale.Icerik;

makales.KategoriId = makale.KategoriId;

db.SaveChanges();

return RedirectToAction("Index");

}

return View();

}

catch

{

ViewBag.KategoriId = new SelectList(db.Kategoris, "KategoriId", "KategoriAdi", makale.KategoriId);

return View(makale);

}

}

// GET: AdminMakale/Delete/5

public ActionResult Delete(int id)

{

var makale=db.Makales.Where(m => m.MakaleId == id).SingleOrDefault();

if (makale== null)

{

return HttpNotFound();

}

return View(makale);

}

// POST: AdminMakale/Delete/5

[HttpPost]

public ActionResult Delete(int id, FormCollection collection)

{

try

{

// TODO: Add delete logic here

var makales = db.Makales.Where(m => m.MakaleId == id).SingleOrDefault();

if (makales== null)

{

return HttpNotFound();

}

if (System.IO.File.Exists(Server.MapPath(makales.Foto)))

{

System.IO.File.Delete(Server.MapPath(makales.Foto));

}

foreach(var i in makales.Yorums.ToList())

{

db.Yorums.Remove(i);

}

foreach (var i in makales.Etikets.ToList())

{

db.Etikets.Remove(i);

}

db.Makales.Remove(makales);

db.SaveChanges();

return RedirectToAction("Index");

}

catch

{

return View();

}

}

AdminUyeController : Controller

public ActionResult Index()

{

var uyes = db.Uyes.Include(u => u.Yetki);

return View(uyes.OrderByDescending(u=>u.UyeId).ToList());

}

// GET: AdminUye/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Uye uye = db.Uyes.Find(id);

if (uye == null)

{

return HttpNotFound();

}

return View(uye);

}

// GET: AdminUye/Create

public ActionResult Create()

{

ViewBag.YetkiId = new SelectList(db.Yetkis, "YetkiId", "Yetki1");

return View();

}

// POST: AdminUye/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "UyeId,KullaniciAdi,Email,Sifre,AdSoyad,YetkiId")] Uye uye)

{

if (ModelState.IsValid)

{

db.Uyes.Add(uye);

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.YetkiId = new SelectList(db.Yetkis, "YetkiId", "Yetki1", uye.YetkiId);

return View(uye);

}

// GET: AdminUye/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Uye uye = db.Uyes.Find(id);

if (uye == null)

{

return HttpNotFound();

}

ViewBag.YetkiId = new SelectList(db.Yetkis, "YetkiId", "Yetki1", uye.YetkiId);

return View(uye);

}

// POST: AdminUye/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "UyeId,KullaniciAdi,Email,Sifre,AdSoyad,YetkiId")] Uye uye)

{

if (ModelState.IsValid)

{

db.Entry(uye).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.YetkiId = new SelectList(db.Yetkis, "YetkiId", "Yetki1", uye.YetkiId);

return View(uye);

}

// GET: AdminUye/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Uye uye = db.Uyes.Find(id);

if (uye == null)

{

return HttpNotFound();

}

return View(uye);

}

// POST: AdminUye/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Uye uye = db.Uyes.Find(id);

db.Uyes.Remove(uye);

db.SaveChanges();

return RedirectToAction("Index");

}

AdminYorumController : Controller

{

private mvcblogDB db = new mvcblogDB();//Veritabanı nesnesi oluşturuyoruz

// GET: AdminYorum

public ActionResult Index()

{

var yorums = db.Yorums.Include(y => y.Makale).Include(y => y.Uye);

//yorumları tarihe göre listeliyoruz

return View(yorums.OrderByDescending(y=>y.Tarih).ToList());

}

// GET: AdminYorum/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Yorum yorum = db.Yorums.Find(id);

if (yorum == null)

{

return HttpNotFound();

}

return View(yorum);

}

// GET: AdminYorum/Create

public ActionResult Create()

{

ViewBag.MakaleId = new SelectList(db.Makales, "MakaleId", "Baslik");

ViewBag.UyeId = new SelectList(db.Uyes, "UyeId", "KullaniciAdi");

return View();

}

// POST: AdminYorum/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "YorumId,Icerik,UyeId,MakaleId,Tarih")] Yorum yorum)

{

if (ModelState.IsValid)

{

db.Yorums.Add(yorum);

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.MakaleId = new SelectList(db.Makales, "MakaleId", "Baslik", yorum.MakaleId);

ViewBag.UyeId = new SelectList(db.Uyes, "UyeId", "KullaniciAdi", yorum.UyeId);

return View(yorum);

}

// GET: AdminYorum/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Yorum yorum = db.Yorums.Find(id);

if (yorum == null)

{

return HttpNotFound();

}

ViewBag.MakaleId = new SelectList(db.Makales, "MakaleId", "Baslik", yorum.MakaleId);

ViewBag.UyeId = new SelectList(db.Uyes, "UyeId", "KullaniciAdi", yorum.UyeId);

return View(yorum);

}

// POST: AdminYorum/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "YorumId,Icerik,UyeId,MakaleId,Tarih")] Yorum yorum)

{

if (ModelState.IsValid)

{

db.Entry(yorum).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.MakaleId = new SelectList(db.Makales, "MakaleId", "Baslik", yorum.MakaleId);

ViewBag.UyeId = new SelectList(db.Uyes, "UyeId", "KullaniciAdi", yorum.UyeId);

return View(yorum);

}

// GET: AdminYorum/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Yorum yorum = db.Yorums.Find(id);

//Makale silinince yorumları da siliniyor.

if (yorum == null)

{

return HttpNotFound();

}

return View(yorum);

}

// POST: AdminYorum/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Yorum yorum = db.Yorums.Find(id);

db.Yorums.Remove(yorum);

db.SaveChanges();

return RedirectToAction("Index");

}

UyeController : Controller

{

public ActionResult Index(int id)

{

var uye = db.Uyes.Where(u => u.UyeId == id).SingleOrDefault();

if (Convert.ToInt32(Session["uyeid"]) != uye.UyeId)

{

return HttpNotFound();

}

return View(uye);

}

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(Uye uye)

{//Giriş yapan kullanıcı adı, email ve şifre veritabanında eşleşiyorsa giriş yapıyor

var login = db.Uyes.Where(u => u.KullaniciAdi == uye.KullaniciAdi).SingleOrDefault();

if(login.KullaniciAdi==uye.KullaniciAdi && login.Email==uye.Email && login.Sifre == uye.Sifre)

{

Session["uyeid"] = login.UyeId;

Session["kullaniciadi"] = login.KullaniciAdi;

Session["yetkiid"] = login.YetkiId;

return RedirectToAction("Index", "Home");

}

else

{

ViewBag.Uyari = "Kullanıcı Adı,Mail ya da Şifrenizi Kontrol Ediniz!";

return View();

}

}

public ActionResult Logout()

{//uyeid Null yapılıp siteden çıkış yapılıp, anasayfa ya geçiyor.

Session["uyeid"] = null;

Session.Abandon();

return RedirectToAction("Index", "Home");

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(Uye uye,HttpPostedFileBase Foto)

{

if (ModelState.IsValid)

{

if (Foto != null)//Fotoğraf seçildiyse işlemleri yapar

{

WebImage img = new WebImage(Foto.InputStream);

FileInfo fotoinfo = new FileInfo(Foto.FileName);

//Fotoğraf oluşturuluyor.

string newfoto = Guid.NewGuid().ToString() + fotoinfo.Extension;

//Gelen her fotoğraf boyutu ayarlanıp öyle gösteriliyor

img.Resize(75,75);

img.Save("~/Uploads/UyeFoto/" + newfoto);

uye.Foto = "/Uploads/UyeFoto/" + newfoto;

uye.YetkiId= 2;

db.Uyes.Add(uye);//Uye veritabanına ekleniyor

db.SaveChanges();

Session["uyeid"] = uye.UyeId;

Session["kullaniciadi"] = uye.KullaniciAdi;

return RedirectToAction("Index", "Home");

}

else

{

ModelState.AddModelError("Fotoğraf", "Fotoğraf Seçiniz.");

}

}

return View(uye);

}

public ActionResult Edit(int id)

{

var uye = db.Uyes.Where(u => u.UyeId == id).SingleOrDefault();

if (Convert.ToInt32(Session["uyeid"]) != uye.UyeId)//Her üye(id lerine bakılır.) kendi profilini düzenler

{

return HttpNotFound();

}

return View(uye);

}

[HttpPost]

public ActionResult Edit(Uye uye,int id,HttpPostedFileBase Foto)

{

if (ModelState.IsValid)

{

var uyes = db.Uyes.Where(u => u.UyeId == id).SingleOrDefault();

if (Foto != null)

{

if (System.IO.File.Exists(Server.MapPath(uyes.Foto)))

{

System.IO.File.Delete(Server.MapPath(uyes.Foto));

}

WebImage img = new WebImage(Foto.InputStream);

FileInfo fotoinfo = new FileInfo(Foto.FileName);

string newfoto = Guid.NewGuid().ToString() + fotoinfo.Extension;

img.Resize(75,75);

img.Save("~/Uploads/UyeFoto/" + newfoto);

uyes.Foto = "/Uploads/UyeFoto/" + newfoto;

}

uyes.AdSoyad = uye.AdSoyad;

uyes.KullaniciAdi = uye.KullaniciAdi;

uyes.Sifre = uye.Sifre;

uyes.Email = uye.Email;

db.SaveChanges();

Session["kullaniciadi"] = uye.KullaniciAdi;

return RedirectToAction("Index", "Home",new { id=uyes.UyeId});

}

return View(uye);

}

public ActionResult UyeProfil(int id)

{//Herkese açık üye profini

var uye = db.Uyes.Where(u => u.UyeId == id).SingleOrDefault();

return View(uye);

}