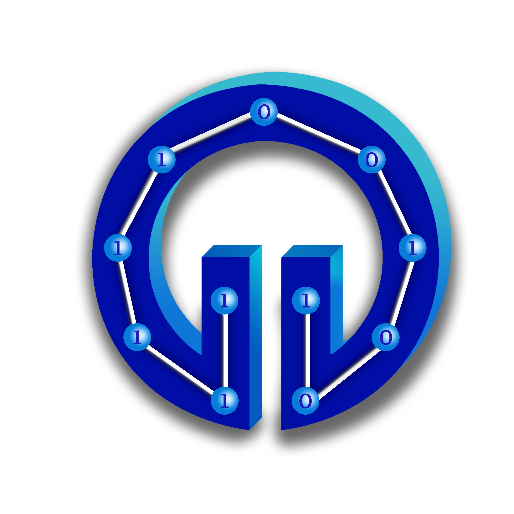
**KARADENİZ TEKNİK ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

****

**WEB TABANLI BULUT DEPOLAMA SİSTEMİ**

**BİTİRME PROJESİ**

**Furkan KAYA**

**Fazlı Furkan KOÇ**

**Burak Taha MARANGOZ**

**2021-2022 BAHAR DÖNEMİ**

**KARADENİZ TEKNİK ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**WEB TABANLI BULUT DEPOLAMA SİSTEMİ**

**BİTİRME PROJESİ**

**Furkan KAYA**

**Fazlı Furkan KOÇ**

**Burak Taha MARANGOZ**

**2021-2022 BAHAR DÖNEMİ**

|  |  |  |
| --- | --- | --- |
| ieee.jpg | **IEEE Etik Kuralları**  **IEEE Code of Ethics** | **ieee.jpg** |

Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayırımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990’da onaylanmıştır.

**ÖNSÖZ**

“WEB TABANLI BULUT DEPOLAMA SİSTEMİ” isimli bu çalışma, Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü’nde bahar dönemi bitirme projesi olarak hazırlanmıştır.

Projenin başlangıcından itibaren bize yol gösteren, projenin yapılmasında fikirleriyle bize destek olan çok değerli hocamız Doç. Dr. Hüseyin PEHLİVAN’a teşekkür ederiz.

Furkan KAYA

Fazlı Furkan KOÇ

Burak Taha MARANGOZ

Trabzon 2022

**İÇİNDEKİLER**

|  |  |
| --- | --- |
|  | Sayfa No |
| [IEEE ETİK KURALLARI](#ieee) | II |
| [ÖNSÖZ](#önsöz) | III |
| İÇİNDEKİLER | IV |
| [ÖZET](#özet) | V |
| 1. [GENEL BİLGİLER](#genelbilgiler) | 1 |
| * 1. [Giriş](#giriş) | 1 |
| * 1. [Bulut Depolama Nedir?](#bdnedir) | 1 |
| * 1. [Bulut Depolamanın Avantajları](#avantajdezavantaj) | 1 |
| 1.3.1. [Avantajlar](#avantaj) | 2 |
| 1.3.2. [Dezavantajlar](#dezavantaj) | 2 |
| 2.YAPILAN ARAŞTIRMALAR | 3 |
| 2.1. HTML VE EJS | 3 |
| 2.2. CSS | 4 |
| 2.3. BOOTSTRAP | 5 |
| 2.4. MongoDB | 5 |
| 2.5. Mongoose | 5 |
| 2.6. Node.js | 5 |
| 2.7. Express.js | 5 |
| 3. PROJE TASARIMI | 6 |
| 3.1. Gereksinim Analizi | 6 |
| 3.1.1. Projenin Amacı | 6 |
| 3.1.2. Hedef Kitle | 6 |
| 3.1.3. Proje Fonksiyonları | 6 |
| 3.1.4. Teknoloji ve Donanım | 6 |
| 3.1.5. Arayüz Tasarımı | 6 |
| 3.2. Mimari Tasarım | 7 |
| 3.3. UML Nesne Modeli | 9 |
| 4. YAPILAN ÇALIŞMALAR | 10 |
| 4.1. IDE Kurulumu | 10 |
| 4.2. Node.js Kurulumu | 11 |
| 4.3. Npm Paket Yöneticisinin Kurulumu | 11 |
| 4.4. Express.js Kurulumu | 12 |
| 4.5. Multer Paketi Kurulumu | 13 |
| 4.6. Mongoose Paketi Kurlumu | 13 |
| 4.7. MongoDB Veri Tabanı Oluşturma | 14 |
| 4.8. Front-End Kodlama | 15 |
| 4.8.1. Karşılama Sayfası | 15 |
| 4.8.2. Giriş Sayfası | 16 |
| 4.8.3. Kayıt Ol Sayfası | 17 |
| 4.8.4. Ana Sayfa Tasarımı | 18 |
| 4.9. Back-End Kurulumu | 20 |
| 4.9.1. Mongoose İle Şema Oluşturma | 20 |
| 4.9.2. Şifreleme | 21 |
| 4.9.3. Multer İle Dosya Yükleme | 22 |
| 4.9.4. Yönlendirmeler | 22 |
| 5.KAYNAKLAR | 23 |
| [STANDARTLAR ve KISITLAR FORMU](#standartlar) | 24 |

**ÖZET**

Bu projede web tabanlı, kolay erişilebilir ve kullanılabilir bir bulut depolama sistemi yapılması amaçlanmıştır. Kullanıcılar üye olduktan sonra giriş yaparak depolama alanlarına erişebilir ve verilerini yükleme ve indirme işlemlerini gerçekleştirebilirler. Verilerin güvenliği gözetilerek, kullanıcı bilgileri şifreli bir şekilde saklanır.

Projenin front-end kodlamasında *HTML*, *CSS* işaretleme dilleri kullanılmıştır. Back-end kodlamasında ise sunucu taraflı projeler geliştirmeye olanak sağlayan, *JavaScript Runtime* platformu olan *Node.js* kullanılması amaçlanmıştır.

**1. GENEL BİLGİLER**

**1.1. Giriş**

Günümüzde bilgisayarlarımızdaki ya da telefonlarımızdaki dosyalara her gün yenileri ekleniyor. E-posta kutumuz, telefonumuzun depolama alanı biraz daha doluyor. Videolar, fotoğraflar, önemli iş dosyaları toplanıp kocaman bir alanı kaplayabilecek hale geliyor. Hal bu olunca tüm bu önemli dosyaları hem istenildiği zaman erişilebilecek hem de güvende olmalarını sağlayacak şekilde depolamak gerekiyor. Verilerimizi depolamak için ise bulut depolama sistemleri kullanılabiliyor.

**1.2. Bulut Depolama Nedir?**

Bulut depolama, verilerin sunucular tarafından ağ üzerinde sanal olarak oluşturulan havuzlarda depolamasıdır. Özellikle artan veri, fotoğraf, video ve dosya gibi ihtiyaçların sanal ortamda muhafaza edilmesi ve kullanıcıların internet erişimi olan her yerde verilerine erişebilmesi için büyük bir ihtiyacı karşılayan sistemlerdir. Telefon ve bilgisayarların depolama alanları yeterli gelmediğinde ya da daha kalıcı bir çözüm arandığında bulut depolama sistemlerine başvurulabilir.

****

**Şekil 1.1.** Bulut depolama

**1.3.** **Bulut Depolamanın Avantajları/Dezavantajları**

Teknolojinin gelişmesi, çoğu zaman istediğimiz veriye hemen ulaşabilmemizi de gerekli hale getirdi. Örneğin; ofis dışından çalıştığımızı ya da tatilde olduğumuzu ve bilgisayarımıza o an için erişebilecek bir durumda olmadığımızı düşünün. Göndermemiz gereken acil bir belge varsa, bunu istediğimiz an istediğimiz yerden erişim sağlayabileceğimiz bulut sistemi üzerinden dosyayı hızlıca gönderebiliriz.

**1.3.1.** **Avantajlar**

* **Her yerden erişim**: Dosyalarınıza internete bağlı olduğunuz her yerden erişebilirsiniz.
* **Uygulama verilerinin senkronizasyonu**: Şifre vb. veriler, uygulama deneyimini kolaylaştırmak için tüm cihazlarda buluta senkronize edilebilir.
* **Entegre yedekleme**: Dosyalarınızın kopyaları onları bozulmaya karşı koruyan birden fazla yerde bulunur.
* **Herkese açık bağlantılar gönderme**: Kullanıcılar senkronizasyon klasörlerinde bir dosyanın genel bağlantısını oluşturabilir ve bunu e-posta yoluyla paylaşabilir.
* **Fotoğrafların otomatik yüklenmesi**: Mobil cihazda bulut depolama uygulamalarını kullanmak, kullanıcıların fotoğraflarına daha fazla koruma sağlar. Kamera rulosu, fotoğrafların otomatik olarak yüklemelerine olanak tanır.
* **Sabit disk alanından tasarruf**: Senkronizasyon ile bilgisayarınızda sabit disk alanı boşaltmanız mümkündür. Cömert depolama kotaları ile fiziksel depolama alanından tasarruf edilmesine olanak tanır.

**1.3.2.** **Dezavantajlar**

* **Hız sorunları**: Büyük dosyaların internet üzerinden aktarılması çok zaman alır. Video dosyaları, büyük *CAD* ya da *.psd* verileriyle uğraşıyorsanız bulut depolama baş ağrılarına neden olabilir.
* **Güvenlik**: Bulut depolama alanını kullanırken en önemli endişelerden biri de güvenlik ve şifrelemedir. Bilgisayarınızdan ayrılmadan önce verileri şifreleyen bulut depolama önerilerinden birini seçmezseniz, güvenlikte sıkıntılar yaşayabilirsiniz.
* **Uyumluluk**: Etkili bir şekilde iş birliği yapmak ve aynı klasördeki dosyaları paylaşmak için, insanların bir bulut depolama hizmetine kaydolmaları gerekir. Birkaç ekiple iş birliği yapıyorsanız, bazı kullanıcılar birden çok bulut depolama hizmetine kaydolmak zorunda kalabilir.

**2. YAPILAN ARAŞTIRMALAR**

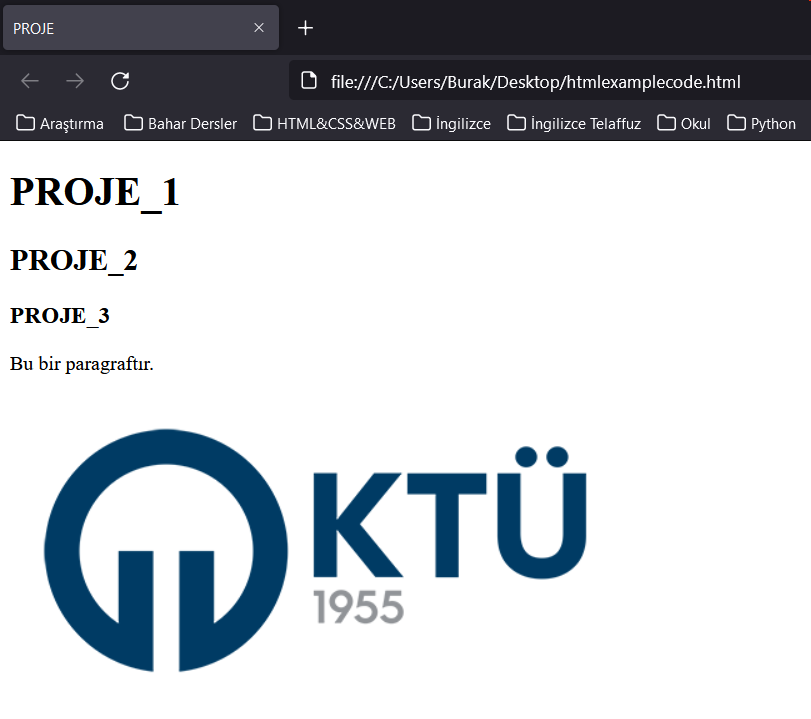
**2.1. HTML VE EJS**

HTML açılımı Hyper Text Markup Language Türkçesi ise Hiper Metin İşaretleme Dili’dir. Sayfaların temel tasarımı için gerekli olan HTML işaretleme dilinde yazı, video veya görsel öğelerin internet sayfaları içerisinde yerleştirilmesini ve istenildiği yere konumlandırılmasını en uygun şekilde yapmak için araştırmalar yapıldı. Yapılan araştırmalar sonucunda HTML dosyaları içerisinde JavaScript kodlarını çalıştırabilmek için EJS modülünü kullanmamız gerektiğini tespit ettik.

EJS(embedded javascript) HTML dosyaları içinde JavaScript kodları yazmamızı sağlayan bir kütüphanedir. JavaScript kodlarını basitçe HTML dosyalarını içine eklememizi sağlar. Dosya uzantıları .html yerine .ejs uzantısı yapılır.

Burada HTML’nin temel kullanımı iki adet ana etiketten oluşur:

1.Head Etiketi: <head></head> etiketi içerisinde yer alan sayfamızın baş ve üst kısmı yani sayfa hakkında bilgiler verir ve arama motorlarına açıklar(SEO-Search Engine Optimization diye duyduğunuz kavram burayla ilgilidir), sayfanın title denen isim kısmını, kaynak dosyalarının tanıtıldığı ve yüklendiği yapılar ve daha fazlası buraya yazılır. Bu head yapısını içerisindeki şeylerin çoğunu kullanıcı arayüzünde göremez, bunlar arkaplanda body kısmı için ön hazırlık ve tarayıcılar için yardımcı gibi çalışır.

2.Body Etiketi: <body></body> etiketi içerisine ise aslında şu anda herhangi bir web sitesine girdiğinizde gördüğünüz her şey yazılabilir diyebiliriz. Liste, paragraf, link, resim, video, blog sayfa tasarımları vb. her şeyi buraya yazabiliriz.metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2.1.** Temel düzeyde bir html kod parçası ve çıktısı.

**2.2. CSS**

CSS (Cascading Style Sheets - Basamaklı Stil Sayfaları), web sayfalarının tarayıcıda nasıl görüneceğini kontrol eden bildirimsel bir dildir. CSS, HTML ve JavaScript ile birlikte web’in temelini oluşturan üç temel teknolojiden biridir. Genellikle HTML ögelerine biçim vermek için kullanılır.

CSS ile bir HTML dosyasında yer alan tüm ögelere biçim verebilir, sayfadaki yerleşim düzenlerini kontrol edebilir. CSS dili bahsedilen fonksiyonlardan ibaret değildir fakat temel olarak bunları söyleyebiliriz.

Şekil 1’deki HTML dosyasına aşağıdaki CSS dosyasını <link href="htmlexamplecode.css" type="text/css" rel="stylesheet"/> kodu ile ekliyoruz. İki dosyada aynı klasör altında bulunmaktadır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2.2.** Css kod örneği.



**Şekil 2.3.** CSS ile düzenlenmiş çıktı.

Çıktıda görüldüğü üzere bu CSS kodu HTML dosyası içerisinde bulunan h1 etiketi içindeki yazıyı renk ve boyut olarak değiştirmemize yarıyor. Bu CSS ile yapılabilen en basit işlemlerden biridir.

**2.3. BOOTSTRAP**

Bootstrap, Kurduğumuz web sitesini farklı ekran boyutlarına uyarlayabilmek için (tablet, telefon, bilgisayar vb.) kullanılan bir framework’tür. Tüm cihazlara uygun tasarıma “responsive web design” denilmektedir.

Bootstrap’ın diğer bir özelliği ise bir web sitesini şekillendirmek için gereken zamanı kısaltmaya, öğeleri düzenlemenin karmaşıklığını basitleştirmeye ve birden çok tarayıcıda (Chrome, Safari, Firefox, Internet Explorer, Opera, Yandex Browser gibi) çalışmaya yarar.

**2.4. MongoDB**

C++ ile geliştirilmiş açık kaynak kodlu veritabanı uygulaması. NoSQL yapıdadır, verileri JSON şeklinde saklamaktadır. İlişkisel veritabanlarındaki table yapısının yerini burada collection, row yapısının yerini document, column yapısının yerini field alır.

**2.5. Mongoose**

MongoDB veritabanı ile ilgili nesne ilişkilendirme işlemlerini kolaylaştıran bir ODM(Object Document Mapper) modülüdür. İlişkisel veritabanı sistemindeki ORM’in NoSQL versiyonu olarak düşünebiliriz. Avantaj olarak esnek, şema bazlı, zengin özellikleriyle geliştiriciye kullanım kolaylığı sağlar, JSON nesneleri ile çalışmamızı ve bu nesneler üzerinden CRUD işlemleri gerçekleştirmemize olanak sağlar.

**2.6. Node.JS**

Temel olarak JavaScript komutlarının sunucu tarafında çalışması diyebiliriz. Google’ın V8 motoru için geliştirilmiş olan Node.JS açık kaynaklı Javascript runtime environment’dır. Node.JS ile ölçeklendirilebilir web uygulamaları hızlı bir şekilde geliştirilebilmektedir. Sonuç olarak uygulamanın(proje) sunucu tarafında gerçeklenmesi için kullanılan teknojilerden bir tanesi de Node.JS teknolojisidir

Node.JS’in en büyük farkı modüler yapısı ve bu yapıyı destekleyen paket yönetim sistemi olan npm (node.js package manager) dir. Npm’i Node.js projelerinde kullanılmak üzere içerisinde bir çok modül barındıran bir depo gibi düşünebiliriz. Bu modüller open-source geliştiricileri tarafından yazılarak npmjs.com üzerine yüklenmektedir.

**2.7. Express.js**

Express Node.js tabanlı web uygulama sunucu çatısıdır. HTTP’de bulunan bütün komutlarını içerir. Açık kaynak kodludur. MySQL, MongoDB veritabanı teknolojilerini destekler. Aslında bir Node.js web uygulaması geliştirme framework’üdür. Express.js’in sunmuş olduğu avantajlardan birisi de küçük boyutlu ve esnek çalışma imkanı sunmasıdır. Barındırdığı çok sayıda HTTP komut çalıştırma yöntemleriyle beraber API oluşturmak oldukça kolaydır.

CRUD(create,read,update,delete) modeline destek verir. Hiçbir Node.js özelliğini engellemeden hafif çalışma imkanıyla sistemi yormadan uygulama geliştirmeye yardımcı olur.

Yukarıda bahsedilen teknojiler projede Backend ve Frontend kısmında kullanılacak olan teknolojilerin araştırılıp sade ve anlaşılır örnekler ile açıklanmaya çalışılmıştır.

Bu teknolojilerin tezimizin yapılan çalışmalar kısmında proje gerçeklendikten sonra nasıl kullanıldığı detaylı olarak açıklanacaktır.

**3. PROJE TASARIMI**

**3.1. Gereksinim Analizi**

**3.1.1. Projenin Amacı**

Bu projede, kullanıcıların cep telefonlarından ve bilgisayarlarında erişebildiği bir bulut depolama sistemi geliştirilmesi amaçlanmıştır.

**3.1.2. Hedef Kitle**

Bu proje internet erişimi olan her cihazdan dosyalarına ulaşmak isteyen, dosyalarını güvenli şekilde saklamak isteyen ve cihazları arasında kolay bir şekilde dosya aktarımı yapmak isteyen kullanıcılara hitap ediyor.

**3.1.3. Proje Fonksiyonları**

Sitem bir web uygulamasından oluşmaktadır. Sisteme bir internet tarayıcısı ile siteye erişilebilir. Kullanıcılar kaydolmalı ve üye girişi yapmalıdır. Giriş yapan kullanıcılar şu fonksiyonlara erişebilir;

* Dosya Yükleme
* Dosya İndirme
* Dosya Silme
* Klasör Oluşturma
* Klasör Silme

**3.1.4. Teknoloji ve Donanım**

Back-end programlama için *Node.js*, Front-end için *HTML*, *CSS* ve *JS* dilleri kullanılması amaçlanmıştır.

**3.1.5. Arayüz Tasarımı**

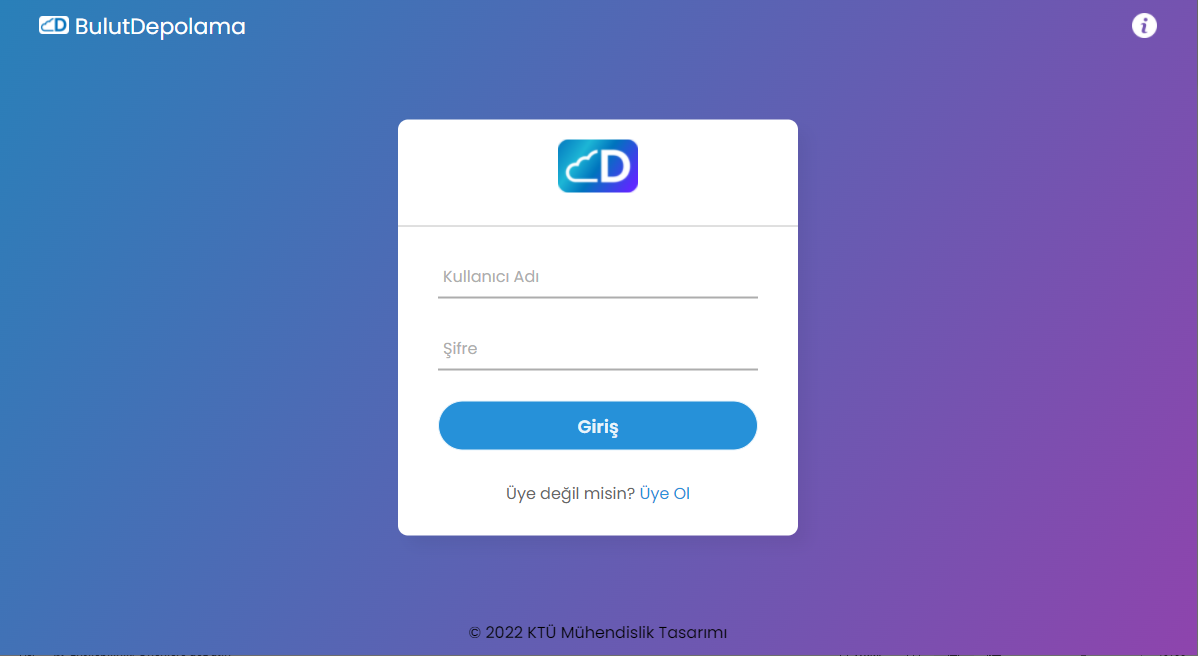
Web sitesinin tasarımı sade ve kolay kullanılabilir olacaktır. “Giriş yap” ve “Kayıt Ol” butonu kullanıcıların kolayca erişebileceği şekilde konumlandırılacaktır. Giriş yaptıktan sonra kullanıcılar ana sayfada dosyalarını görüntüleyecektir. “Dosya Yükle” seçeneği ile siteme dosya yükleyip, Dosyaların yanında bulunan “İndir” seçeneği ile dosyalarını indirebilecektir.

**3.2. Mimari Tasarım**

*HTML*, *CSS* ve *JS* kullanarak web uygulamasının Front-End kısmının kodlanması planlanmıştır. Web uygulaması şu sayfalardan oluşmaktadır;

* Giriş Yap
* Kayıt Ol
* Şifre Değiştir
* Ana Sayfa
* Yardım

Giriş sayfasında kullanıcılar kullanıcı adı ve şifrelerini kullanarak giriş yapabilecek ve üyelik oluşturma sayfana ulaşarak e-postaları ile kendilerine kullanıcı adı ve şifre belirleyerek sisteme kaydolabileceklerdir.

****

**Şekil 3.1.** Giriş sayfası

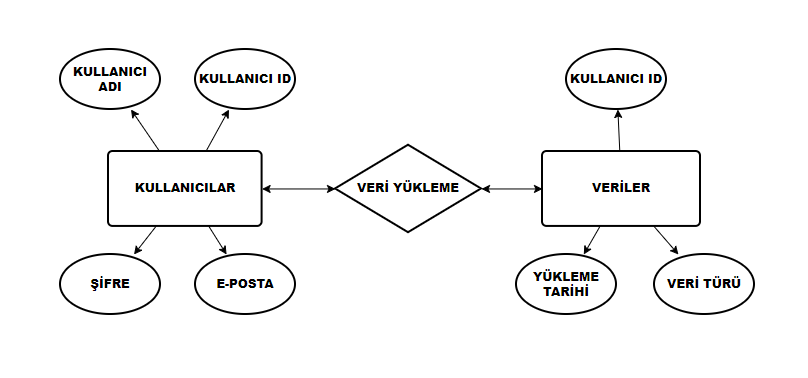
Ana sayfada yüklü dosyalarına erişebilecek ve üst kısımda bulunan “Dosya Seç” ve “Yükle” butonları ile sisteme seçtikleri dosyalarını yükleyebileceklerdir. Aynı zamanda bu ekrandan yüklü olan dosyaların yanında bulunan seçenekler ile dosyalarını silebilecek ve indirebileceklerdir. Sağ üst kısımda konumlandırılacak “Bilgi” butonu ile kullanıcılar uygulamanın nasıl kullanılacağı ve nasıl üye olunacağı hakkındaki bilgilere erişebileceklerdir.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

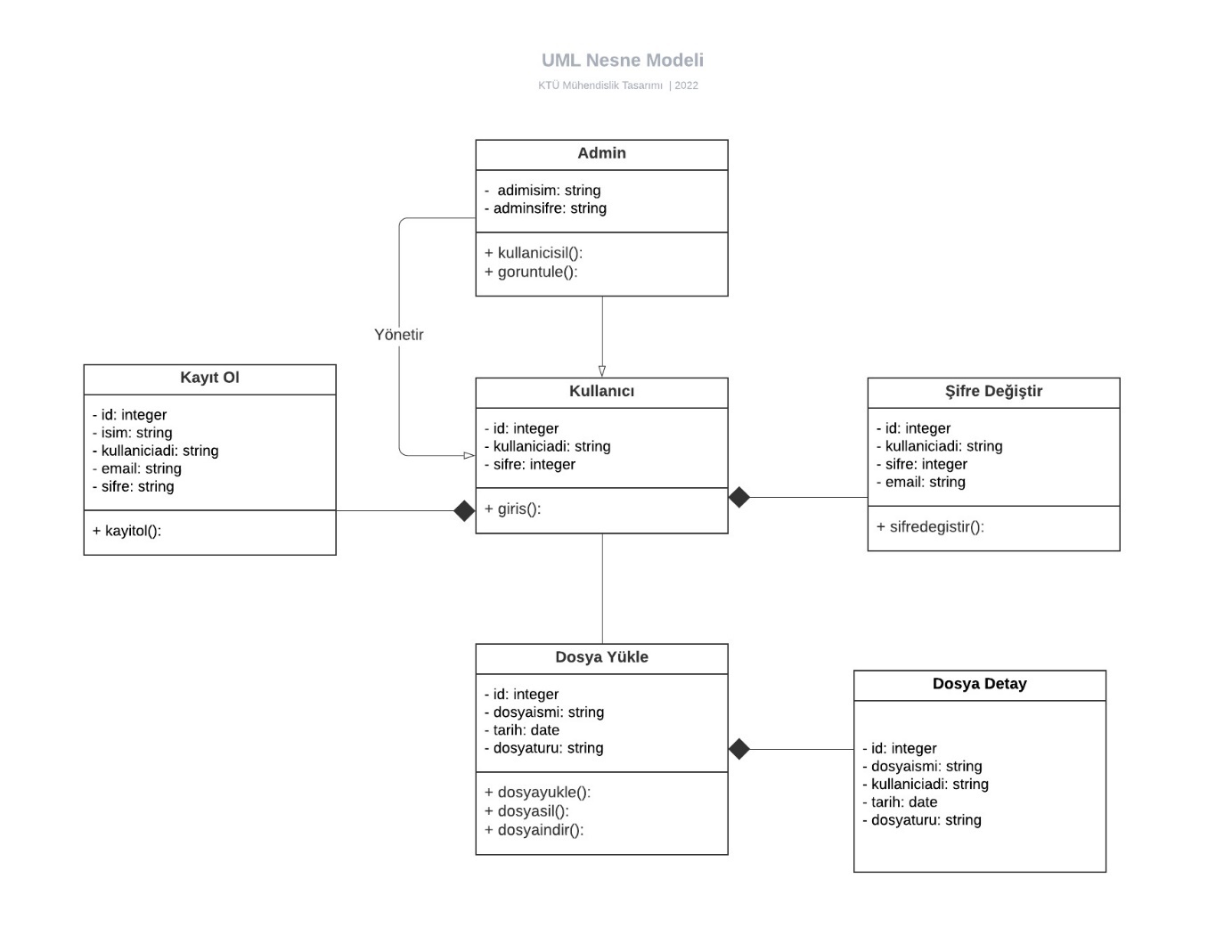
**Şekil 3.2.** Ana sayfa

*Node.js* ve gerekli paketleri kullanılarak projenin Back-End kısmı geliştirilecek ve veri tabanı bağlantıları yapılacaktır. Web uygulaması için gerekli olacak veri tabanı tabloları oluşturulacaktır. Güvenlik amacıyla kullanıcıların şifre bilgileri veri tabanında şifreli bir şekilde tutulacaktır.



**Şekil 3.3.** Veri tabanı ER diyagramı

**3.3.** **UML Nesne Modeli**

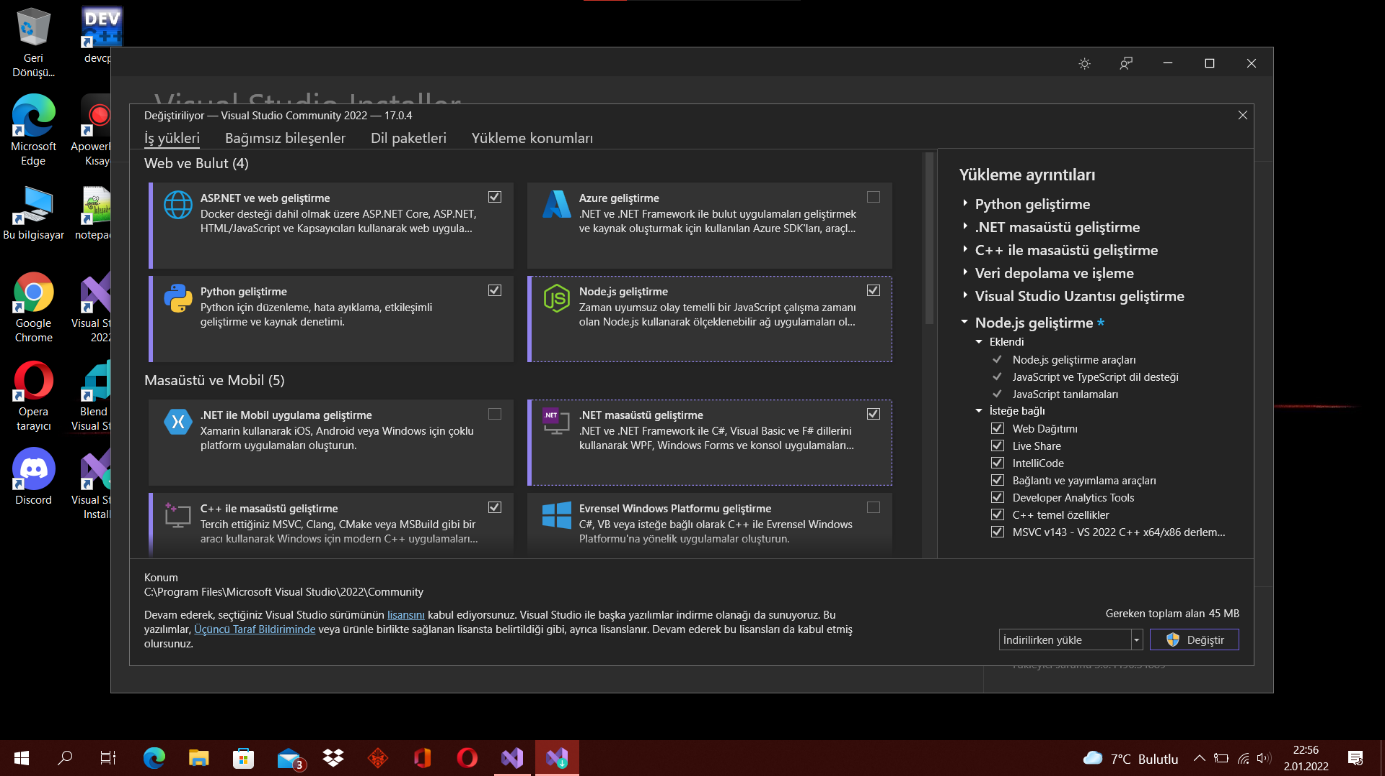
****

**Şekil 3.4.** UML Diyagramı

**4.** **YAPILAN ÇALIŞMALAR**

**4.1. IDE Kurulumu**

Kod düzenleyicisi olarak *Visual Studio* editörü seçildi. Editör kurulumu yapılırken kullanılacak olan gerekli geliştirici ve paketler seçilerek kurulum tamamlandı.



**Şekil 4.1.** Visual Studio kurulumu

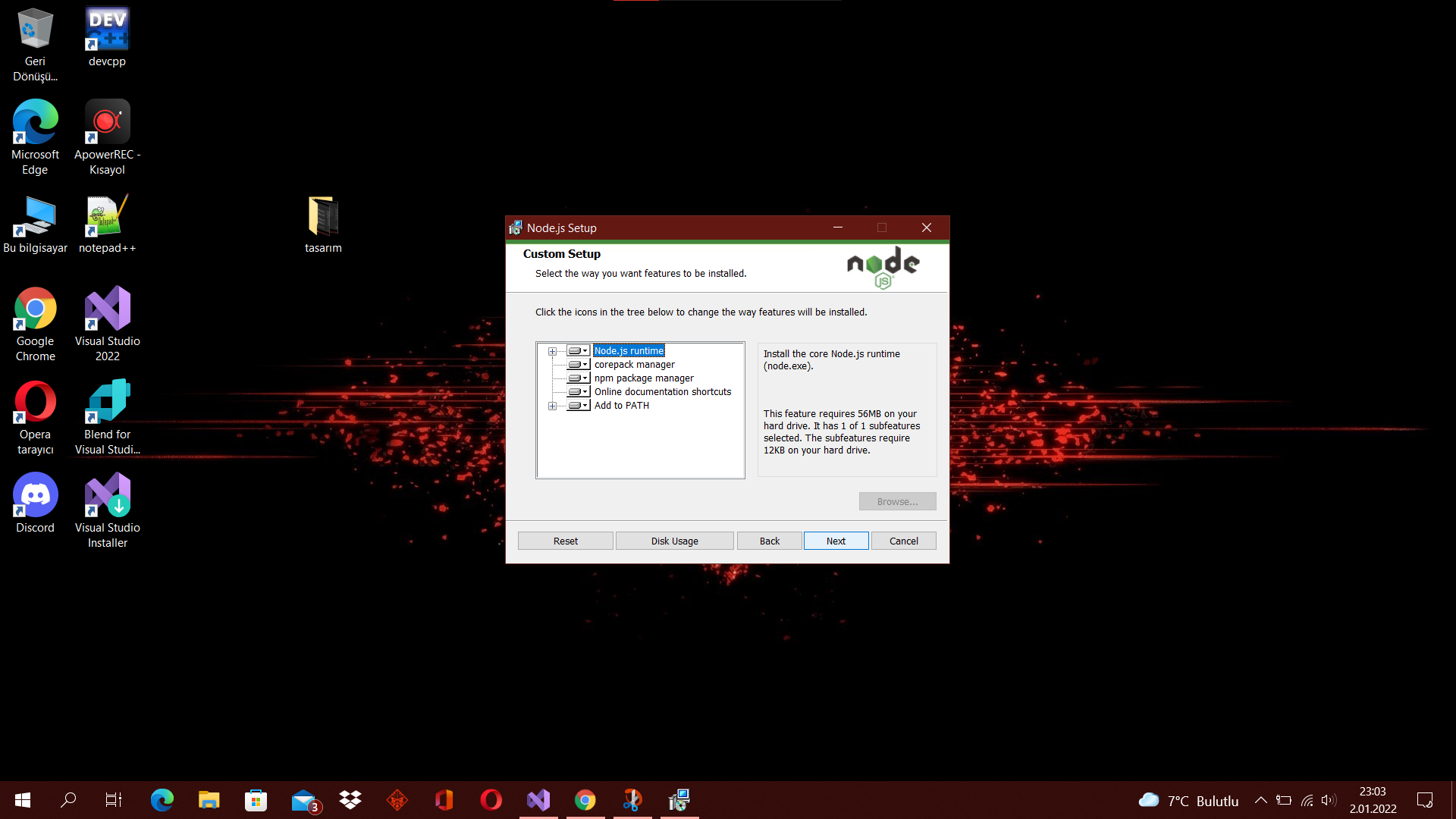
Daha sonra Visual Studio üzerinden, seçilen klasöre bir Node.js projesi oluşturularak adlandırıldı. Node.js ve paketlerinin kurulumu ile ilgili bilgiler alt başlıklarda yer almaktadır.

**metin, ekran görüntüsü, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**4.2. Node.js Kurulumu**

Tarayıcı üzerinden *Node.js* yükleyicisi indirilerek *Node.js* kurulumu yapıldı. *Visual Studio* üzerinden yeni bir *Node.js Web Application* projesi oluşturuldu.

****

**Şekil 4.2.** Node.js kurulumu

**4.3. Npm Paket Yöneticisinin Kurulumu**

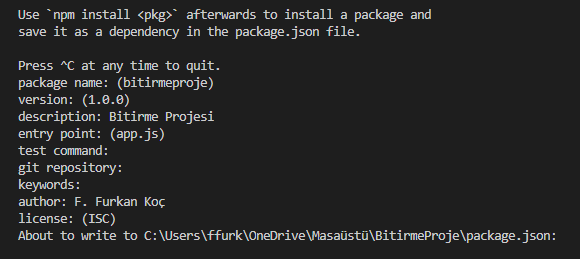
Npm paket yöneticisi, işe yarar birçok paketi projemize çok kolay bir şekilde eklememizi sağlar. Kurulum için Komut İstemi’de kurulacak ilgi dizine gidilir.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.3.** Npm paket yöneticisinin proje dizinine kurulumu.

npm init komutu çalıştırıldığında proje için bir JSON dosyası da oluşturulur. Bu dosyanın içeriğini kendimiz girebiliriz veya boş bırakabiliriz.

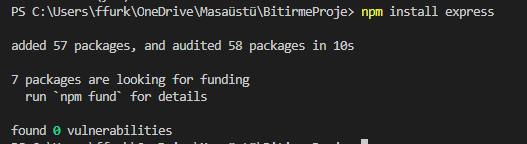


**Şekil 4.4.** JSON dosya ayarları.

Bu package.json dosyasında paketler arası bağımlılıklar, versiyonlar, proje yazarı, lisans vs. bilgiler tutulur.

**4.4. Express.js Kurulumu**

Express, Node.js içerisinde bulunan http modülünü kullanarak web uygulamalarını kolay ve hızlı bir şekilde geliştirmeyi sağlayan pakettir. Projenin olduğu dizine gidilerek kurulum yapılır.

****

**Şekil 4.5.** Express’in proje dizinine kurulumu.

*npm install express* komutu ile npm paket yöneticisi kullanarak Express paketi kurulur. Kurulduktan sonra projeye implement etmek için aşağıdaki kod parçasını kullanılır.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.6.** Express’in projeye eklenmesi.

**4.5.** **Multer Paketi Kurulumu**

Multer dosya işlemleri için kullanılan bir ara katman modülüdür. Birçok dosya türünü kapsar. Kurulum için aynı Express gibi ilgili dizine girilir. *npm install multer* komutuyla yüklenir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 4.7.** Multer’in proje dizinine kurulumu.

Multeri projeye ekledikten sonra dosya yükleme işlemi için aşağıdaki kodlar kullanılır. Bu kod yükleme uploads dizinine yapılır. Yükleme diske veya belleğe yapılacaksa storage motorları kullanılmalıdır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 4.8.** Multer’in projeye eklenmesi.

**4.6. Mongoose Paketi Kurulumu**

Mongoose geliştirilen uygulamaya göre model oluşturarak MongoDB işlemlerini kolaylaştıran bir ODM modülüdür. MongoDB veritabanında veriler karmaşık bir şekilde saklanabilir. Bu verilerin karmaşık saklanması verilerle işlem yaparken çeşitli zorluklar ortaya çıkartır. Bu sorunun önüne geçebilmek ve veri içindeki alanları belirlemek için modelleme yapılır. Mongoose kurulumu için, terminal kullanılarak proje dizinine gidilir ve *npm install mongoose* komutu ile paket yüklenir.



**Şekil 4.9.** Mongoose’un proje dizinine kurulumu.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 4.10.** Mongoose’un projeye eklenmesi.

**4.7. MongoDB Veri Tabanı Oluşturma**

Verileri saklamak için Mongoose modülü ile Node.js’le uyumlu bir şekilde çalışan, açık kaynak kodlu No SQL yapıdaki MongoDB veritabanını kullanılmakta. MongoDB’de öncelikle bir database oluşturuldu. Ardından bu database’e erişim yetkisi olan bir kullanıcı oluşturuldu.

**tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.11.** MongoDB veri tabanı.

Veri tabanı oluşturulduktan sonra, MongoDB tarafından oluşturulan bağlantı kodu ve oluşturulan kullanıcının bilgileri eklenerek aşağıda verilen kod ile veri tabanına bağlantı sağlanmaktadır.

****

**Şekil 4.12.** Veri tabanı bağlantısı.

**4.8. Front-End Kodlama**

**4.8.1. Karşılama Sayfası**

Öncelikle kullanıcıların sisteme ilk girdiklerinde karşılarına çıkacak karşılama sayfasının tasarımı yapıldı. Bu sayfada kullanıcıları giriş yapma ve kaydolma sayfalarına yönlendiren 2 adet buton eklendi.

**metin, ekran, ekran görüntüsü, gümüş içeren bir resim

Açıklama otomatik olarak oluşturuldu**

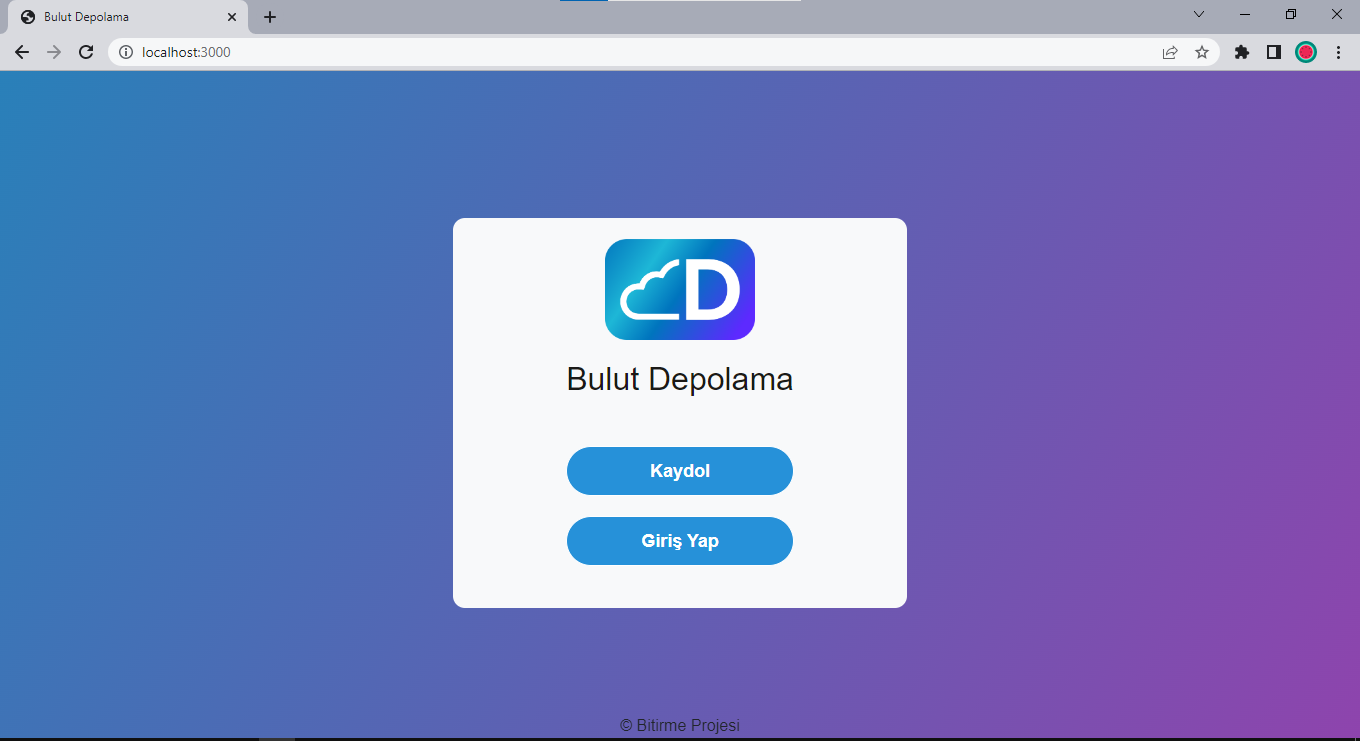
**Şekil 4.13.** Karşılama sayfası HTML kodları.

Daha sonra sayfanın tasarımı CSS kodları kullanılarak düzenlendi. Tasarımda sadelik ve kullanım kolaylığı göz önünde bulunduruldu. CSS kodları aşağıdaki gibidir.

**metin, ekran, ekran görüntüsü, gümüş içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.14.** Karşılama sayfası CSS kodları.

****

**Şekil 4.15.** Karşılama Sayfası

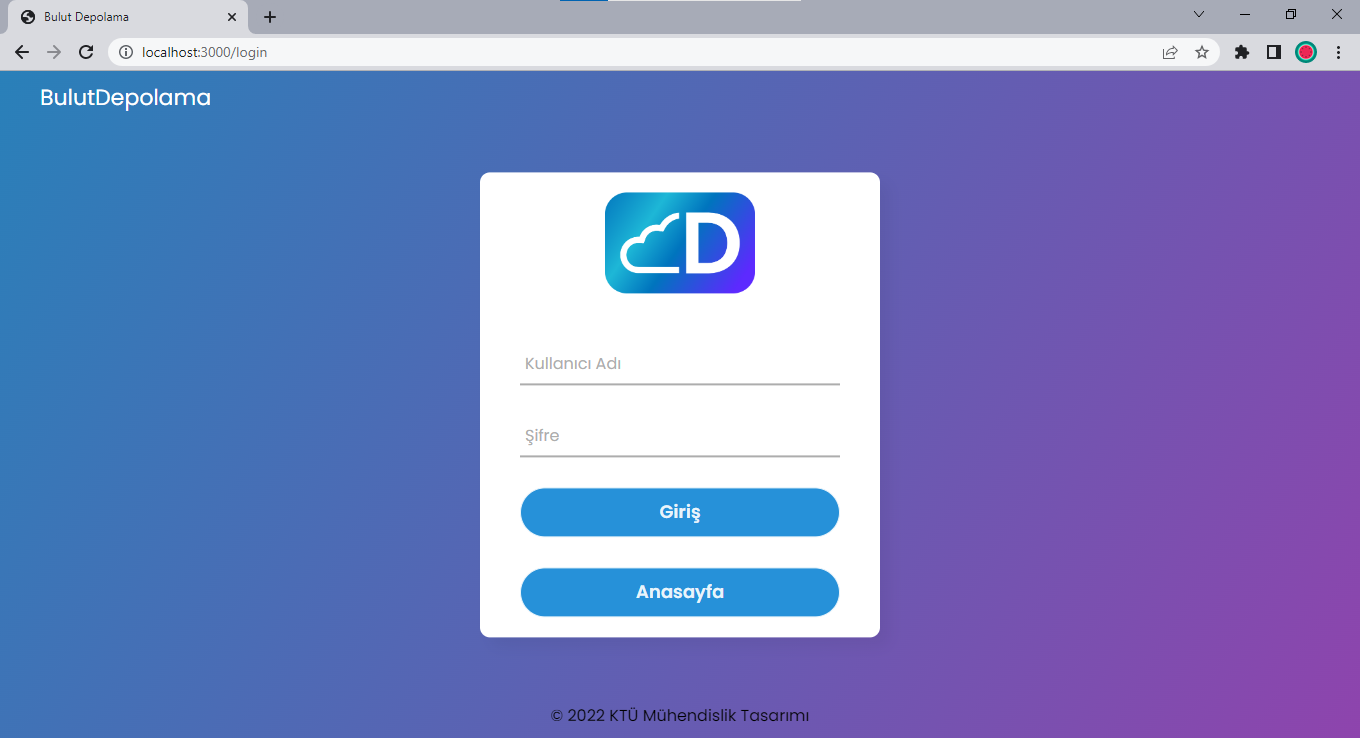
**4.8.2. Giriş Sayfası**

Kayıtlı kullanıcıların kullanıcı adı ve şifreleri ile giriş yapacağı sayfa ve giriş paneli oluşturuldu. Sayfanın altına karşılama sayfasına dönmek için bir buton eklendi. Kullanıcılar bilgilerini yazdıktan sonra gerekli kontroller sağlanıyor ve kullanıcı bilgileri veri tabanına kayıtlı ise giriş yapılıyor.

**metin, ekran, ekran görüntüsü, gümüş içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.16.** Giriş sayfası HTML kodları.

****

**Şekil 4.17.** Giriş sayfası.

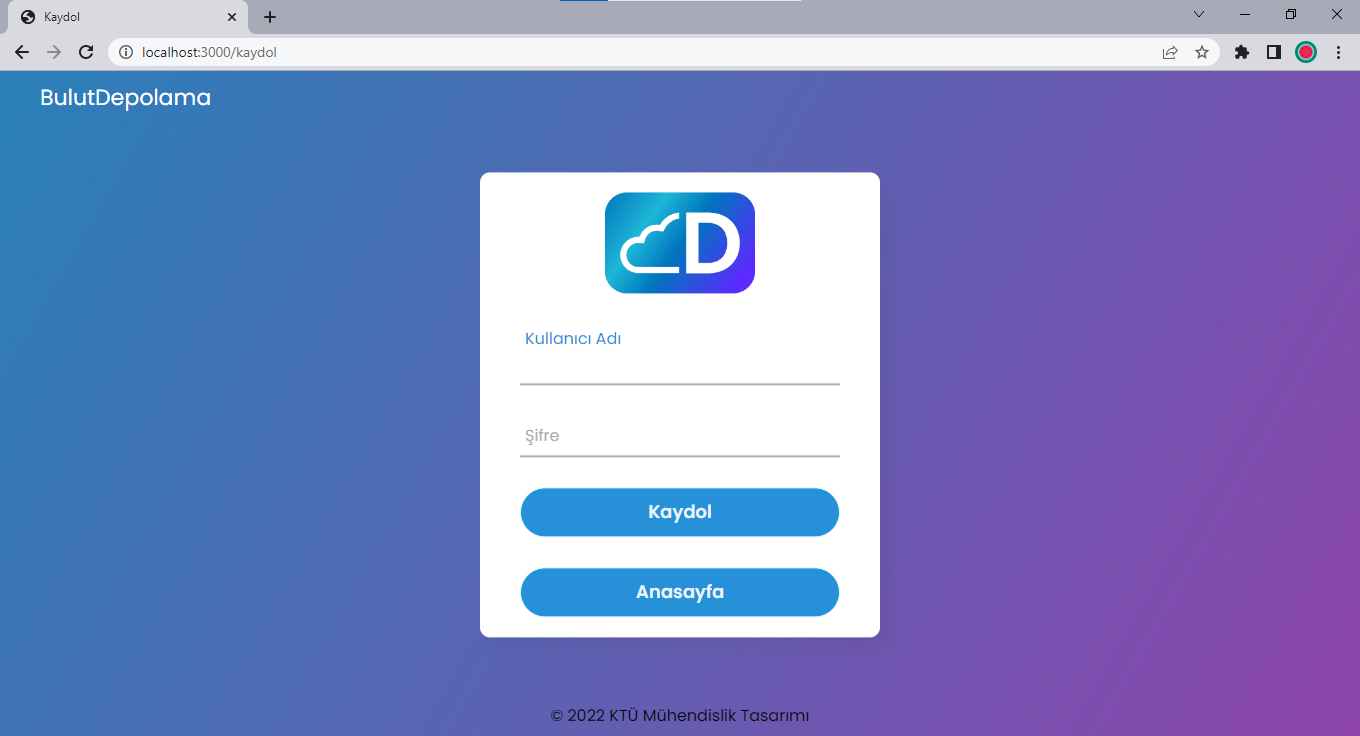
**4.8.3. Kayıt Ol Sayfası**

Kullanıcıların, kullanıcı adı ve şifreleri seçerek sisteme kaydolacakları kayıt olma sayfası tasarlandı. Giriş sayfasında olduğu gibi panelin altına ana sayfaya dönmek için buton eklendi.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 4.18.** Kayıt olma sayfası HTML kodları.

****

**Şekil 4.19.** Kayıt olma sayfası.

**4.8.4. Ana Sayfa Tasarımı**

Bu sayfa kullanıcıların dosyalarını ve klasörlerini görüntüleyebileceği ana sayfadır. Bu sayfada klasör oluşturma, dosya yükleme, indirme ve silme işlemleri yapılabilir. Kullanıcılar dosya yükleme seçeneği ile bilgisayar veya telefonlarında bulunan dosyayı kişisel hesaplarına yükleyebilirler. Her kullanıcının verileri localde ayrı adreslerde tutulduğu için karışıklık olmaz. Sayfa tasarımında kullanılan HTML ve CSS kodları aşağıdaki gibidir.

metin içeren bir resim

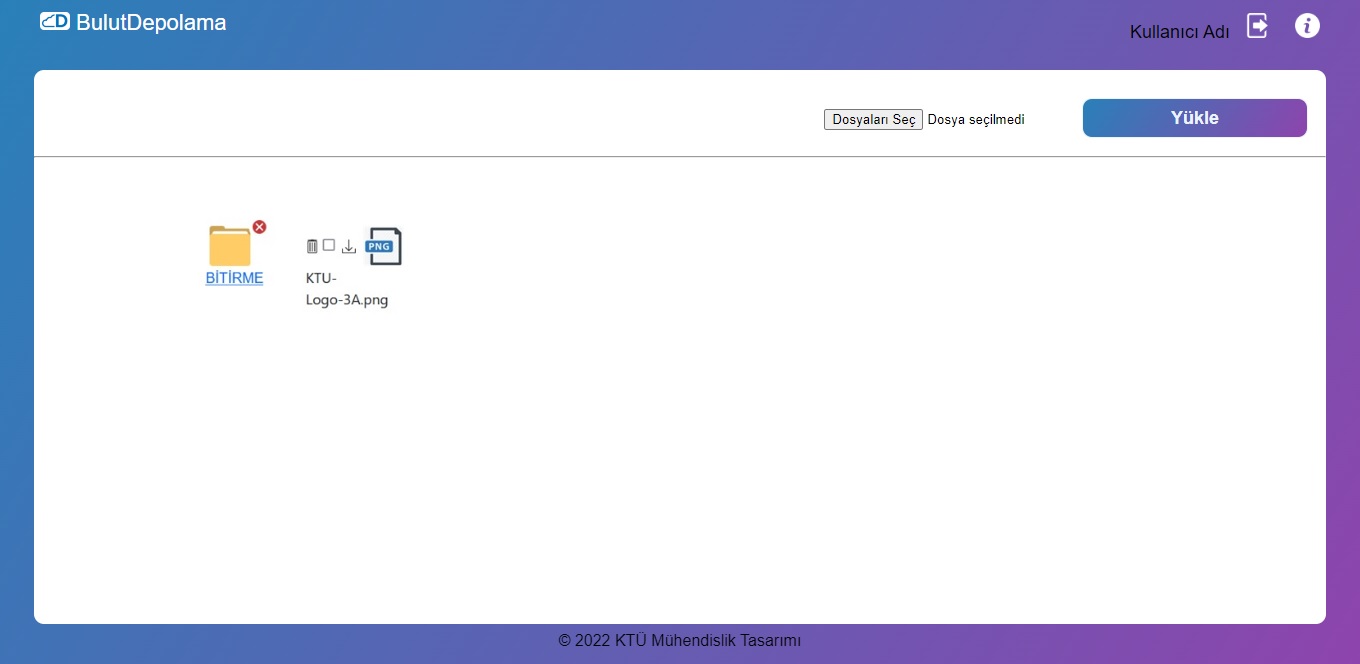
Açıklama otomatik olarak oluşturuldu

**Şekil 4.20.** Ana sayfa HTML kod örneği.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.21.** Ana sayfa CSS kod örneği.

****

**Şekil 4.22.** Ana sayfa tasarımı.

**4.9. Back-End Kodlama**

**4.9.1. Mongoose İle Şema Oluşturma**

Gerekli veriler Mongoose şemalerı ile veri tabanına yüklenmekte. Öncelikle kullanıcı bilgilerini saklamak için aşağıda kodu verilen User şeması oluşturuldu.

const userSchema = new mongoose.Schema({

id: String,

password: String

});

Dosyalar için bir şema oluşturuldu.

const userSchema = new mongoose.Schema({

id: String,

password: String

});

Klasörler için bir şema oluşturuldu.

var klasorlerSchema = new mongoose.Schema({

klasorHangiKullanicida: String,

klasorunAdi: String,

items: [dosyalarSchema]

});

Bu şemalar modellenip bir liste oluşturuldu.

**var Item = mongoose.model("Item", dosyalarSchema);**

**var File = mongoose.model("File", klasorlerSchema);**

**const listSchema = new mongoose.Schema({**

**name: String,**

**items: [dosyalarSchema]**

**});**

**var List = mongoose.model("List", listSchema);**

**4.9.2. Şifreleme**

Web platformunda kullanıcıyla etkileşim sonucu güvenliğine riayet edilmesi gereken bilgiler mevcuttur. Bizim projemizde hassasiyet barındıran ve gizili tutulması gereken bilgi kullanıcı şifreleridir. Şifreleme işlemini Node.js paketi olan Crypto ile yapmaktayız. Bu modül sayesinde kullanıcı şifreleri bir anahtar koda göre şifrelenerek veri tabanına saklanmakta. Bu 44 karakterlik anahtar ile hashlenerek veri tabanına saklanan şifreler güvende tutulur.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.23.** Encrypt fonksiyonu.

Daha sonra kullanıcı giriş yaparken veri tabanında tutulan şifrelenmiş veri Decrypt fonksiyonu ile çözülerek getirilir. Bu sayede veri tabanında tutulan verilerin güvenliği sağlanmış olur.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 4.24.** Decrypt fonksiyonu.

**4.9.3. Multer İle Dosya Yükleme**

Multer ile yüklenen dosyaların hangi lokasyona kaydedileceği aşağıdaki kod ile belirleniyor. Yükleme yaparken storage motoru kullanılıyor. Dosya yüklenirken destination kodu ile dosyanın yükleneceği konum, filename ile de dosyanın hangi isimle kaydedileceği belirlenir.

var storage = multer.diskStorage({

destination: function(req, file, cb) {

cb(null, 'uploads/');

},

filename: function(req, file, cb) {

cb(null, file.originalname);

}

});

const upload = multer({

storage: storage

});

**4.9.4. Yönlendirmeler**

Buradaki kodlar ile sayfalar arası yönlendirmeler yapılıyor. İlk sayfa olarak adres çubuğuna “/” isteği geldiği anda karşılama sayfası gösteriliyor. Gelen isteği “*get”* fonksiyonu ile alıp, gösterilmesi gereken sayfayı “*render”* ile gösteriyor.

app.get("/", function(req, res) {

var dir = './uploads';

if (!fs.existsSync(dir)){

fs.mkdirSync(dir);

}

res.render("hosgeldiniz.ejs");

});

app.get("/kaydol", function(req, res) {

res.render("signin.ejs");

});

app.get("/login", function(req, res) {

res.render("login.ejs");

});

**4. KAYNAKLAR**

**1.** Bulut bilişim nedir? <https://azure.microsoft.com> Erişim: 19 Kasım 2021

**2.** Bulut depolama. [https://tr.wikipedia.org](https://tr.wikipedia.org/wiki/) Erişim: 27 Kasım 2021

**3.** Cloud Storage System. [https://www.sciencedirect.com](https://www.sciencedirect.com/) Erişim: 9 Aralık 2021

**4.** Okutucu, Baki Onur, *“Bulut Bilişim ve Teknolojileri”*, Yüksek Lisans Tezi, Okan Üniversitesi, 2012 Erişim: 10 Aralık 2021

**5.** Overview of Cloud Storage System. [https://www.researchgate.net](https://www.researchgate.net/) Erişim: 13 Aralık 2021

**6.** Visual Studio. [https://visualstudio.microsoft.com](https://visualstudio.microsoft.com/tr/) Erişim: 19 Aralık 2021

**7.** About Node.js. [https://nodejs.org](https://nodejs.org/en/) Erişim: 19 Aralık 2021

**8.** HTML. <https://www.w3schools.com> Erişim: 20 Aralık 2021

**9.** Unified Modeling Language. <https://commons.wikimedia.org> Erişim: 27 Aralık 2021

**10.** Veritabanı yönetim sistemi (DBMS). <https://www.oracle.com> Erişim: 2 Ocak 2022

**STANDARTLAR ve KISITLAR FORMU**

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

|  |
| --- |
| Tamamı bize ait olan yeni bir projedir. |

1. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

|  |
| --- |
| Hayır. |

1. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

|  |
| --- |
| Proje tasarlama ve kodlama becerilerimizi kullandık. |

1. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

|  |
| --- |
| Evrişimsel/Blok kodlama, hata bulma ve düzeltme gibi standartlardan uygun olanlar seçilerek kullanılmıştır. |

1. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi

|  |
| --- |
| Açık kaynak kodlu ve ücretsiz olan platform ve editörleri kullandığımız için mali desteğe ihtiyaç duyulmamıştır. |

b) Çevre sorunları:

|  |
| --- |
| Çevre sorunlarının projemiz üzerinde bir etkisi olmamıştır. |

c) Sürdürülebilirlik:

|  |
| --- |
| İhtiyaçtan yola çıkılarak yapılan bir proje olduğu için sürdürülebilir olduğuna inanıyoruz. |

d) Üretilebilirlik:

|  |
| --- |
| Sistemin kullanıcılara sunduğu depolama alanı ile ilgili değişiklikler yapılabilir. |

e) Etik:

|  |
| --- |
| Proje etikler doğrultusunda yürütüldü. |

f) Sağlık:

|  |
| --- |
| Sağlık açısından bir sorun teşkil etmemektedir. |

g) Güvenlik:

|  |
| --- |
| Şifreleme yöntemini kullanarak güvenliği sağladığımızı düşünüyoruz. |

h) Sosyal ve politik sorunlar:

|  |
| --- |
| Sosyal sorunlara çözüm olarak üretilen bir projedir. |