



Ondokuz Mayıs Üniversitesi

Bilgisayar Mühendisliği

BİL490 - Yazılım Mühendisliği Laboratuvarı

Ders Görevlisi: Dr. Öğr. Üyesi Durmuş Özkan ŞAHİN

E-Sınav Platformu

Yazılım Tanıtım Dokümantasyonu

İçindekiler

1	Proje İçin Kullanılan Yazılımsal Teknolojiler	1
1.1	React.js	1
1.2	Node.js	2
1.3	MongoDB	2
1.4	Visual Studio Code (VS Code)	2
1.5	MongoDB Compass	3
1.6	Visual Paradigm	3
2	Projenin Kod İçeriği	4
2.1	MongoDB	4
2.2	Node.js	6
2.3	React.js	8
3	MyQuiz Sisteminin Ekran Görüntüleri	10
3.1	Ortak Kullanıcı Kısmı	10
3.1.1	Kayıt Ol Sayfası	10
3.1.2	Giriş Yap Sayfası	11
3.2	Öğretmen Seviyesinde Kullanıcı Girişi	11
3.2.1	Ana Sayfa	11
3.2.2	Sınav Hazırlama Sayfası	13
3.2.3	Sınav Sonuçlarını Görüntüleme Sayfası	16
3.2.4	Profil Sayfası	16
3.2.5	Çıkış	16
3.3	Öğrenci Seviyesindeki Kullanıcı Girişi	17
3.3.1	Ana Sayfa	17
3.3.2	Sınav Sonuçlarını Görüntüleme Sayfası	19
3.3.3	Profil Sayfası	19
3.3.4	Çıkış	19
4	Tasarım Deseni	20
5	MyQuiz Sisteminin Kodlarının GitHub Linki	22
6	MyQuiz Sisteminin Yazılım Tanıtım Videosu Linki	22

Şekil Listesi

1	MyQuiz sisteminde kullanılan yazılımsal teknolojiler	1
---	--	---

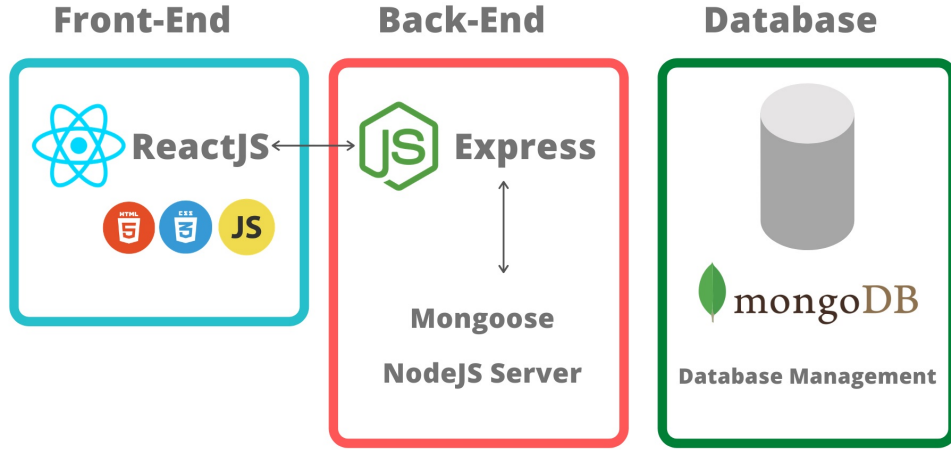
2	VS Code için anket[2]	3
3	MongoDB'yi kullanım için rehber fotoğraf-1	4
4	MongoDB'yi kullanım için rehber fotoğraf-2	5
5	MongoDB'de bulunan veri örneği	5
6	MongoDB bağlantısının .env dosyasına eklenmesi	6
7	MongoDB bağlantısının server.js dosyasına eklenmesi	6
8	Node.js'de sınavların seçilmesi	7
9	İstekler için axios kullanımı	9
10	Server'a kullanıcı kayıt isteği	9
11	Kullanıcı kayıt sayfası	10
12	Kullanıcı giriş sayfası	11
13	Öğretmen seviyesindeki kullanıcının Ana Sayfası	11
14	Öğretmen seviyesindeki kullanıcının sınava giriş ekranı	12
15	Öğretmen seviyesindeki kullanıcının sınav ekranı	12
16	Öğretmen seviyesindeki kullanıcının sınav sonuç ekranı	13
17	Öğretmen seviyesindeki kullanıcının Sınav Sayfası	13
18	Öğretmen seviyesindeki kullanıcının sınav ekleme ekranı	14
19	Öğretmen seviyesindeki kullanıcının oluşturduğu sınavların listesi	14
20	Öğretmen seviyesindeki kullanıcının soru ekleme ekranı	15
21	Öğretmen seviyesindeki kullanıcının soru oluşturma ekranı	15
22	Öğretmen seviyesindeki kullanıcının Sınav Sonuç Görüntüleme Sayfası	16
23	Öğrenci seviyesindeki kullanıcının Ana Sayfası	17
24	Öğrenci seviyesindeki kullanıcının sınava giriş ekranı	17
25	Öğrenci seviyesindeki kullanıcının sınav ekranı	18
26	Öğrenci seviyesindeki kullanıcının sınav sonuç ekranı	18
27	Öğrenci seviyesindeki kullanıcının Sınav Sonuç Görüntüleme Sayfası	19
28	MyQuiz sistemi Gözlemci Tasarım Deseni UML Diyagramı	22

Tablo Listesi

1	MyQuiz sistemi Gözlemci Tasarım Deseni	21
---	--	----

1 Proje İçin Kullanılan Yazılımsal Teknolojiler

Projede kullanılan yazılım geliştirme yöntemi çevik yöntem olduğundan dolayı, seçilen yazılım geliştirme kütüphaneleri proje geliştiricisinin öğrenmekte olduğu kütüphanelerdir. Projede arayüz kısmında (front-end side) React.js, fonksiyonel kısımda (back-end side) Node.js ve veritabanı kısmında (database side) MongoDB kullanılmıştır. IDE olarak da VS Code (Visual Studio Code) kullanılmaktadır.



Şekil 1: MyQuiz sisteminde kullanılan yazılımsal teknolojiler

1.1 React.js

React.js, arayüz geliştirmede kullanılan bir Javascript kütüphanesidir. Başlangıçta Facebook tarafından geliştirilip, 2013 yılında açık kaynak olmuştur. Meta kendi geliştirdiği uygulamalarda React.js'in kullanılmasının yanı sıra başka bir çok geliştirici ve şirket tarafından tercih edilmektedir.

React içerisinde kodun daha kısa ve fonksiyonel olmasını sağlayan Javascript'in bir sürümü olan ES6, sık sık kullanılmaktadır.

React, yapısal olarak DOM ve Virtual DOM (Sanal DOM) kullanmaktadır. Bu sayede yapılan değişikliklere daha hızlı tepki verilebilmektedir. Çünkü React Virtual DOM sayesinde sayfada yapılan değişiklikler kaydedilir ve bu değişiklikler program koşulduğunda DOM'a aktararak sayfada gösterilebilir.

React, tek sayfa üzerinde farklı bölümlerin ayrı birer sayfa gibi davranmasına izin vermektedir. Mesela projede, Ana Sayfa'da bulunan butonlar ile aynı sayfada farklı sekmeler görüntülenebilmektedir. Bu, yazılımın çalışmasını hızlandırmaktadır.

1.2 Node.js

Node.js, Google'ın Chrome V8 JavaScript motoru ile oluşturulan, sunucu tarafında JavaScript çalıştırmayı sağlayan bir çalışma ortamıdır.

Asenkron çalışma mantığına sahiptir. Bir istek geldiğinde, gelen isteği queue'ye (kuyruk) koyar ve arkasından gelen istekleri de aynı şekilde queue'ye yerleştirir. Bu istekleri işleneceği zaman bu kuyruktan alınarak işlenir. Bu sayede, bir istek geldiğinde öncesinde gelen isteği beklemek durumunda kalmadan programa dahil edilmiş olmaktadır. Bu da avantaj olarak hız kazandırmaktadır.

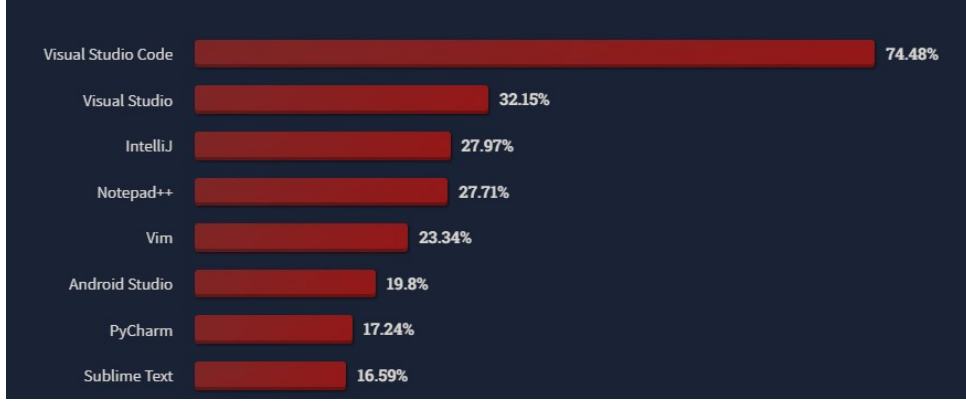
1.3 MongoDB

MongoDB bir veritabanıdır. Veritabanları kaydedip daha sonra tekrar kullanmak istediğimiz verilerimiz sakladığımız sistemlerdir. MongoDB NoSQL tabanlı ve verileri JSON biçiminde saklayan bir veritabanı sistemi olarak 2009 yılında açık kaynak olarak kullanıma sunulmuştur. Verileri ilişkisel şekilde tutulmaktadır. Bu sebeple sorguların çok olduğu yazılım sistemlerinde kullanıma uygun değildir. Bu gibi sistemler için PostgreSQL gibi sistemleri tercih etmek mantıklıdır. Hazırlanan proje, kullanıcısı fazla olmayacağı için ve sistemsel olarak da sorgu sayısı çok fazla olmayan bir sistem olduğu için MongoDB kullanmak mantıklıdır. Ayrıca MongoDB gizlilik ve güvenlik konusunda, mevcut akran sistemlere göre daha zayıftır.

Projede veritabanı olarak MongoDB'yi kullanmak için MongoDB Compass masaüstü uygulaması kullanılmıştır.

1.4 Visual Studio Code (VS Code)

Yazılım geliştirme süreçlerini desteklemek için Microsoft tarafından tasarlanmış olan ve ücretsiz olarak kullanıma sunulan VS Code, hafif, hızlı ve kullanıcı dostu bir arayüze sahiptir. Bir çok programlama diline destek sunar ve bir çok işletim sistemi platformlarında kullanılır. Bir çok kullanıcı gibi bu projede de VS Code tercih edilmiştir. Aşağıda bu durumu gösteren Stack Overflow sitesinde yapılan anket gösterilmektedir.



Şekil 2: VS Code için anket[2]

Ayrıca VS Code geniş bir eklenti desteğine sahiptir. Tüm bunların yanı sıra da geniş bir kullanıcı topluluğu da olduğundan dolayı, topluluğu oldukça geniştir.

1.5 MongoDB Compass

MongoDB Compass popüler olan bir MongoDB grafik arayüzüdür. MongoDB Compass'ı, MongoDB veritabanını kullanmak için kullanılması zorunlu değildir. MongoDB komut istemi üzerinden de kullanılır. Ancak MongoDB Compass'ı kullanarak, komut istemi üzerinden yapılacak işlemleri görselleştirmiş oluruz ve işlemleri daha somut hale getirmiş oluruz. Ve genellikle MongoDB, MongoDB Compass aracılığıyla kullanılır.

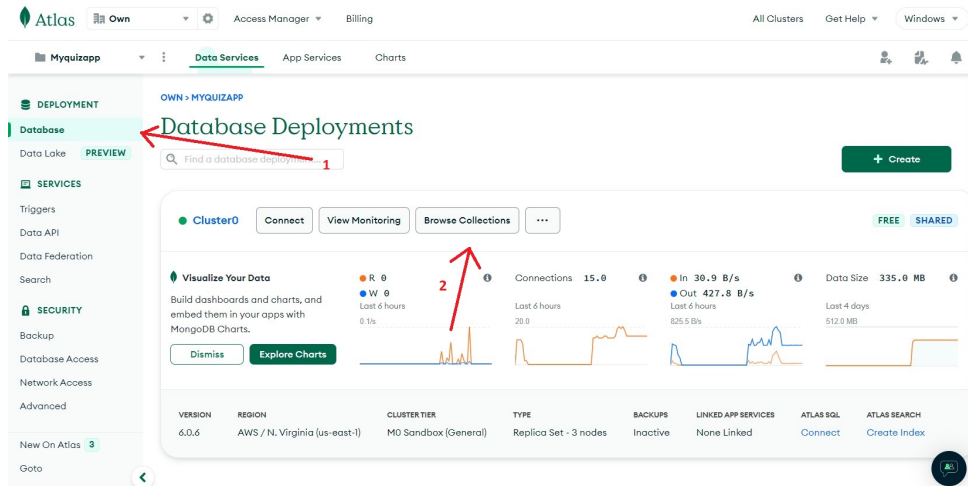
1.6 Visual Paradigm

Visual Paradigm, yazılımsal bir modelleme aracıdır. Bu modelleme aracı sayesinde sistem tasarımı, iş analizi ve sistem geliştirme süreç akışı modellenir. Visual Paradigm bir çok modelleme yöntemlerini kullanmaya imkan tanır. UML (Unified Modeling Language) bu modelleme yöntemlerinden birisidir. MyQuiz sisteminin modellenmesi için UML diyagramları, kullanılan modelleme yöntemlerinden birisidir. Bu sebeple bu modellemeler oluşturulmak için, Visual Paradigm uygulaması kullanılmıştır. Yazılım geliştiren ekip, ekonomik olarak en uygun ve işlevli programı kullanabilmek için Visual Paradigm'in Community sürümünü kullanmıştır.

2 Projenin Kod İçeriği

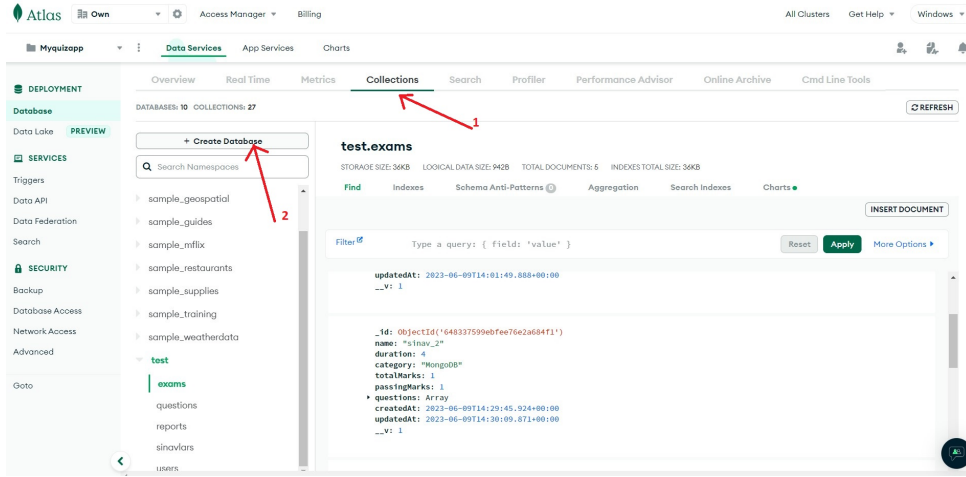
Proje, arayüz React, arka kısım Node.js ve veritabanı MongoDB ile hazırlanmıştır. React ve Node.js bir javascript kütüphanesi olduğundan dolayı, projenin bir kısmında varsayılan klasör ve dosya adları kullanılmıştır. Bu kısımlar kod açıklanırken belirtilecektir. Ayrıca projede Profil Sayfası kısmı hazırlanamamıştır. Bunun sebebi teknik sebepler kadar, geliştiricinin projeyi teslim etmesi gereken zamanda tam olarak hazırlayamamasından kaynaklanmaktadır. Ayrıca projenin geliştiricisi javascript'ın Node.js kütüphanesine tam hakim olamadığından dolayı, sitenin gösterildiği tarayıcının konsolunda bazı hatalar düzeltilememiştir. Bir miktar uğraştan sonra, bu hatalar kod çalışmasında engel teşkil etmediği için gözardı edilmiştir. Projenin kod içeriği anlatılırken MongoDB, Node.js ve React olarak 3 kısma ayrılacaktır. Ancak bu 3 başlıkta kod kısmı anlatılırken, kodun anlaşılabilirliğini mümkün kılabilmek için diğer başlıklarda anlatılan kısımlara başvurulur, bu kısımlar da anlatımın anlaşılabilirliğini mümkün kılacak kadar ifade edilecektir.

2.1 MongoDB



Şekil 3: MongoDB'yi kullanım için rehber fotoğraf-1

Veritabanı olarak MongoDB'nin kullanılabilmesi için öncelikle, MongoDB'nin sitesine gidilerek sisteme kayıt ve giriş yapılması gerekmektedir. Daha sonrasında yönlendirilen sayfadan, Resim1de 1 numaralı okta gösterildiği gibi "Database" sekmesine tıklanır. Sonra 2 numarada gösterildiği gibi "Browse Collection" sekmesine gidilir.



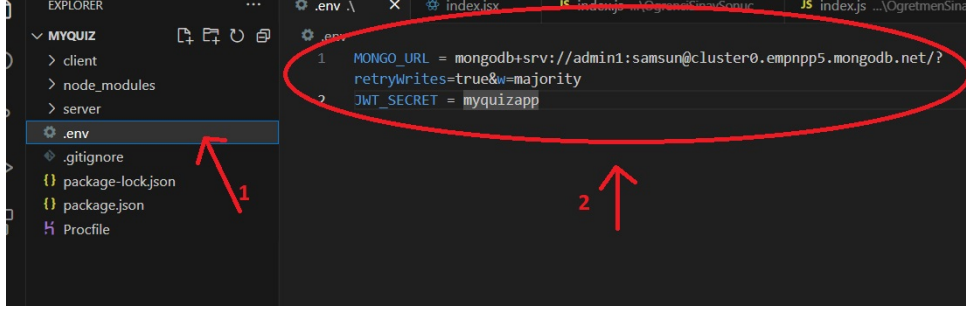
Şekil 4: MongoDB’yi kullanım için rehber fotoğraf-2

Şekil 4’teki gibi 1 numarada gösterildiği şekilde “Browse Collection” sekmesine tıkladıktan sonra, Şekil 4’te 2 numarada gösterildiği gibi “Create Database” bölümünden veritabanı oluşturulur. Daha sonrasında veritabanı ismi ve bu veritabanına ait collection isim/isimleri girilir. MongoDB ilişkisel veritabanı olduğundan SQL gibi tablolar oluşturulmaz. Bilgiler JSON biçiminde kaydedilir. Bu yüzden tutulan veriler JSON formatındadır. Verilerin tutulduğu JSON formatı örneği Şekil 5’te gösterilmektedir.

```
_id: ObjectId('648330c0671937f894836d09')
name: "sinav-1"
duration: 5
category: "Javascript"
totalMarks: 1
passingMarks: 1
▼ questions: Array
  0: ObjectId('648330cd671937f894836d0d')
createdAt: 2023-06-09T14:01:36.649+00:00
updatedAt: 2023-06-09T14:01:49.888+00:00
__v: 1
```

Şekil 5: MongoDB’de bulunan veri örneği

Collention’lar oluşturulduktan sonra tekrar “Database” sekmesine tıklanarak dönülür. Bu sekmede “Cluster”da “Connect” butonuna tıklanarak “Driver” uygulaması seçilir. Buradan bağlantı link kodu kopyalanır.



Şekil 6: MongoDB bağlantısının .env dosyasına eklenmesi

Şekil 6’da 1 numaralı okta gösterildiği gibi .env belgesinde, 2 numaralı okta gösterildiği gibi MONGO_URL değişkenine link yapıştırılır ve kaydedilir.



Şekil 7: MongoDB bağlantısının server.js dosyasına eklenmesi

Buradaki .env paketi Node.js tarafına (yani server ya da back-end tarafına) “dotenv” paketi ile birlikte alınır. Bu projede server.js dosyasında Şekil 7’de kırmızı ok ile gösterilen satırlardaki gibi yapılmaktadır. Bu sayede .env dosyasında bulunan mongodb’ye erişim linki, dotenv paketi aracılığı ile Node.js tarafındaki server.js dosyasındaki kodlar ile veritabanı ve Node.js arasında iletişim kurulmuş olur.

2.2 Node.js

Node.js ile temelde veritabanı arasındaki iletişim, veritabanındaki bilgilerin işlenmesi ve işlenen bu bilgilerin ilgili yerlere yönlendirilmesi sonrasında yetkili kullanıcı tarafından (arayüze bilgilerin verilmesi kısmı) bu bilgilere erişimin sağlanmasında rol oynar. Ayrıca burada kullanıcılara verilen yetkilerin düzenlenmesini de sağlar.

Server tarafında Node.js kütüphanesi koduyla, veritabanındaki verilerin temsil edilmesi modeller, temsil edilen bu verilerin belirtilen kullanıcı işlevleri sonucu oluşan isteklerine uygun olarak işlenmesi router (yönlendirici)’ler, veritabanından alınan kullanıcı yetkisi ile izin işlemleri middleware (izin için ara yazılım dosyası/kodu) ve veritabanı ile iletişim için ilk işlem olan veritabanı linkinin alınması configuration (konfigürasyon) ile sağlanır.

Router'lar ile gelen istekler üzerinde ilgili işlemi yaparız. Mesela projede sınav ile ilgili işlemlerden olan sınav oluşturma, sınav seçme, sınav düzenleme ve sınav silme işlemleri Şekil 8'de gösterilen examRoute.js dosyasında bulunmaktadır.

```
// var olan sınavları id'ye göre al
router.post("/get-exam-by-id", authMiddleware, async (req, res) => {
  try {
    const exam = await Exam.findById(req.body.examId).populate("questions");
    res.send({
      message: "Sınav Başarılı Şekilde Alındı",
      data: exam,
      success: true,
    });
  } catch (error) {
    res.status(500).send({
      message: error.message,
      data: error,
      success: false,
    });
  }
});
```

Şekil 8: Node.js'de sınavların seçilmesi

Şekil 8'de sınavlar, sınavın id'sine göre gelen istek üzerinden seçilir ve sunucuda ilgili rotaya yönlendirilir. Benzer kod yapısı ile kullanıcı ve sınav sonuçlarını görüntüleme ile ilgili işlemlerin ilgili rotaya yönlendirilmesi yapılmıştır.

Router'lara gelecek olan istekler (veriler), Model'ler ile önce sunucu tarafından kullanılabilirlik için veritabanındaki bilgileri sunucu tarafında kullanılacak biçime geçirir. Model'ler kod kısmında "models" klasöründe toplanmıştır. Mesela projede sınav ile ilgili verileri için oluşturulmuş Model'de; sınav ismi, süre, kategori, toplam not, geçme notu ve sorular için oluşturulmuş anahtarlar vardır. Ve bu anahtarlara karşılık gelen kilitlerde ise, anahtarın veri türü ve gereklilik durumu bulunmaktadır.

Kullanıcıların kimliğinin doğrulanması ve izinlerinin, isteğe (request nesnesine) ekler. Böylece Router'lar ile yönlendirme yapılmadan önce istek nesnesi üzerinden yetki kontrolü yapılarak güvenli bir yapı sağlanır. Bu izinler(yetkiler) authMiddleware.js dosyasında tanımlanan JWT(JSON Web Token) ile sağlanır. JWT bir veri parçasıdır. Kullanıcı oturum açtığında sunucu tarafından kendisine bir JWT (bazı kaynaklarda kısaca token denilir) oluşturur. Ve kullanıcı sisteme girip işlemler yaparken oluşan istekler içerisinde JWT'ler de gönderilir. Bu JWT'ler eğer yapılmak istenilen işlemin JWT'si ile doğrulanır ise işleme izin verilir. Dolayısıyla JWT, yetki(izin) işlemlerinin gerçekleşmesini sağlayan yetki verisini temsil eden yazılım parçasıdır. Mesela, MyQuiz sisteminde öğrenci sınav oluşturamaz. Kendisine sınav sayfası gösterilmez. Sınav sayfasını sadece öğretmen görebilir ve kullanabilir. Dolayısıyla, öğrenci seviyesindeki kullanıcı sisteme giriş yaptığında kendisine atanan JWT, Sınav sayfasını görüntüleyebilmek için gerekli olan JWT ile doğrulanmadığı için öğrenci kullanıcıya Sınav sayfası gösterilmez.

2.3 React.js

React ile temelde, html kodlarının yazılacağı alanın tanımlanması (JSX), değişebilecek ve değişmeyecek olan verilerin tutulması ve taşınmasında rol oynayan yapıların tanımlanması (state), sayfada gerçekleştirilen eylemlerin (olayların) yakalanması (event handling) ve yakalanan eylemlerin sonucu olarak sayfaya yönlendirme yapılacaksa yönlendirme işlemlerinin yapılması (routing) işlemlerini sağlar.

Ancak React ile beraber çoğu projede olduğu gibi Redux'da bu projede kullanılmıştır. Redux aslında proje içerisinde global bir depo oluşturur ve bu depoda, React'da değişiklikleri ve verileri tutan state'leri (durum) tutarak global bir değişken özelliği kazandırılır. Böylece uygulama arayüzü Redux sayesinde tek merkezden dağılan bir yapıya dönüştürülür. Ayrıca bu Redux deposu slice'ler ile sınıflandırılarak erişimi ve yönetimin kolaylaştırılmasını sağlar. Projelerde Redux, store.js dosyasıyla beraber projeye uygun olarak belirlenen redux işlemlerine özel oluşturulan redux dosyaları bulunur. Ve bu dosyalar projelerde genellikle redux klasörü altında bulunur. Bu projede de aynı şekilde yapılmıştır. Projede Redux deposu kullanıcı işlemleri için userSlice.js ve state değişikliklerinin ve değerlerinin yüklenmesi aşaması için loaderSlice.js dosyası tanımlanmıştır. Ayrıca Redux dosyalarının tanımlanması projelerde benzer olduğu için Redux yapısının React'da tanımlanması, kullanılmasına nazaran kolaydır.

Bu projede ve bir çok projede API calls (API çağrıları) React içerisinde kullanılır. API çağrıları ile, React ile state'ler aracılığı ile yapılan veri ve eylem değişiklikleri Node.js tarafına bildirilerek server işlemlerine input (girdi) olmasına imkan verir. Ve server'da yapılan değişikliklerin uygun olanları en son olarak gereklilik duyulursa veritabanı tarafında kaydedilir. Projedeki her bir işlem sınıfı için dosya oluşturulur ve bu dosyalarda her bir işlem sınıfının içerdiği işlemler için istekte bulunan fonksiyonlar içerir. Bu istekler server tarafında Route'larda da aynı işlemler için fonksiyonlar bulunur ve bu fonksiyonlar server tarafında işlenir. Bu projede kullanıcıların işlemleri için kullanıcılar.js, sınavlar ile ilgili işlemler için sınavlar.js, sonuç görüntüleme ile ilgili işlemler için sonuclar.js ve server tarafına isteklerin (yetki istekleri de dahil) yapılması ortak olduğu için bu bir dosya içerisinde ayrı yazılarak diğer dosyalardan referans yöntemi ile alınabilmesi için index.js dosyası bulunur.

```

client > src > api_cagirlari > JS index.js > ...
1  import axios from 'axios';
2
3  const axiosInstance = axios.create({
4    headers: {
5      Authorization : `Bearer ${localStorage.getItem('token')}`
6    }
7  });
8
9  export default axiosInstance;

```

Şekil 9: İstekler için axios kullanımı

React'da istekler axios yapıları aracılığıyla yapılır. Şekil 9'da da görüleceği üzere axios oluşturulmuş (create) ve içerisine başlık (header) olarak yetki token'ı yerleştirilmiştir. Bu istek de, Javascript'in ES6 sürümü sayesinde kullanılabilen arrow function yapısı aracılığı ile axiosInstance değişkenine atanarak dışardan erişilebilir hale getirilmiştir.

```

client > src > api_cagirlari > JS kullanicilar.js > loginUser
1  const { default: axiosInstance } = require(".");
2
3  //kullanıcı kayıt
4  export const registerUser = async (payload) => {
5    try {
6      const response = await axiosInstance.post('/api/users/kayit', payload);
7      return response.data;
8    } catch (error) {
9      return error.response.data;
10   }
11 }

```

Şekil 10: Server'a kullanıcı kayıt isteği

Şekil 10'da görüleceği gibi kullanicilar.js içinde, kullanıcı kayıt işlem için tanımlanan registerUser arrow function'ı daha önceden index.js'de istekleri yapmak için oluşturulan axiosInstance değişkenini kullanarak server'a kullanıcı kayıt isteğinde bulunmuştur.

Bu projede kullanıcı sayfaları, sayfalar klasörü içerisinde bulunmaktadır. Ve burada öğrenci, öğretmen ve her iki kullanıcının da ortak kullandığı sayfalar olarak ayrılmıştır. Bu sayede kullanıcılar için ilgili işlemlere ait sayfalar bölümlendirilerek yönetim kolaylaştırılmıştır.

3 MyQuiz Sisteminin Ekran Görüntüleri

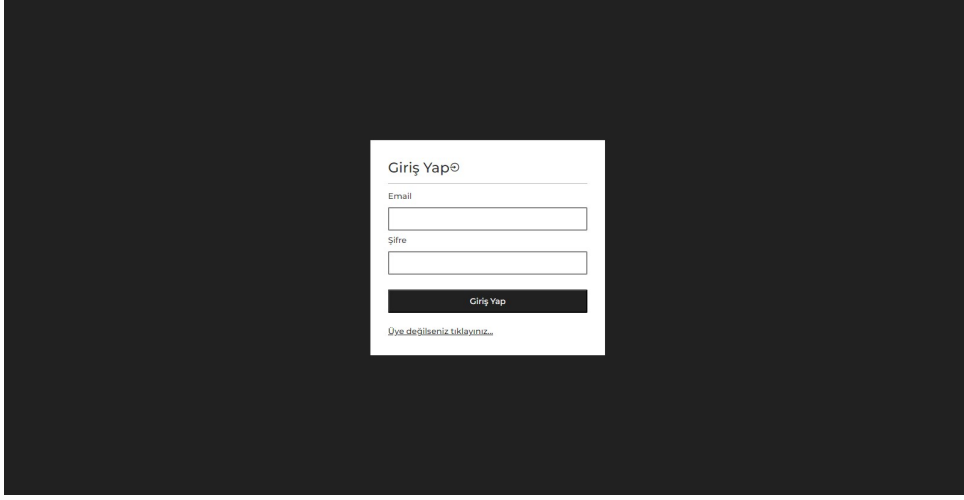
3.1 Ortak Kullanıcı Kısmı

3.1.1 Kayıt Ol Sayfası

Şekil 11: Kullanıcı kayıt sayfası

Şekil 11’de görüleceği üzere öncelikle kullanıcılar sistemi kullanabilmeleri için kayıt sayfasından sisteme kayıt olmaları gerekmektedir. Öğretmen olan kullanıcılar ise, yazılım ekibi ile iletişimleri sonucu kendilerine öğretmen (admin) yetkisi verilir. Bu sayede öğretmen olan kullanıcılar sistemde sınavlar oluşturabilir ve öğrencilerinin sınav sonuçlarını görüntüleyebilirler. Eğer kullanıcılar zaten sisteme kayıtlı ise, “Kayıt Ol” butonunun altında yazan “Üye iseniz giriş yapmak için tıklayınız...” yazısına tıklayarak giriş sayfasına yönlendirilirler.

3.1.2 Giriş Yap Sayfası

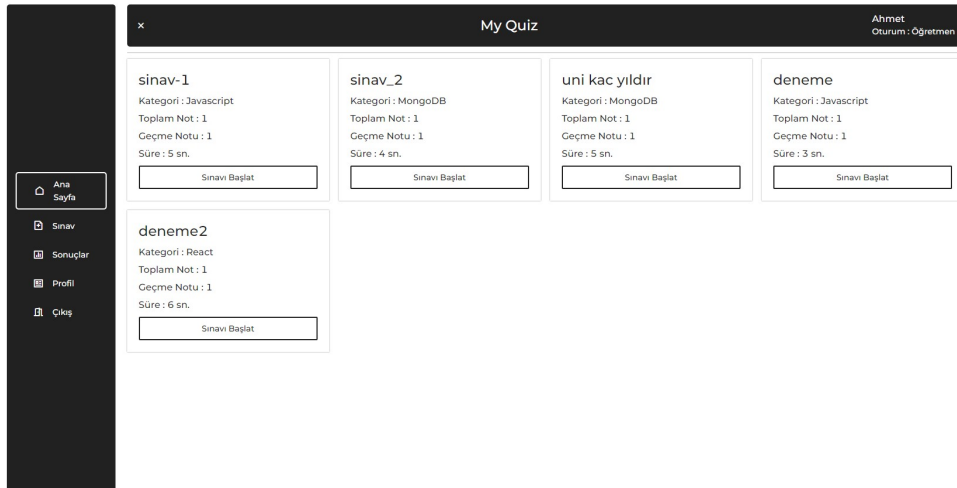


Şekil 12: Kullanıcı giriş sayfası

Şekil 12’de görüleceği üzere kullanıcılar sisteme kayıtlı iseler, eposta ve şifre bilgilerini girerek sisteme giriş yapabilirler. Eğer kullanıcılar sisteme kayıtlı değilseler, “Giriş Yap” butonunun altında yer alan “Üye değilseniz tıklayınız...” yazısına tıklayarak kayıt sayfasına gidebilmektedirler.

3.2 Öğretmen Seviyesinde Kullanıcı Girişi

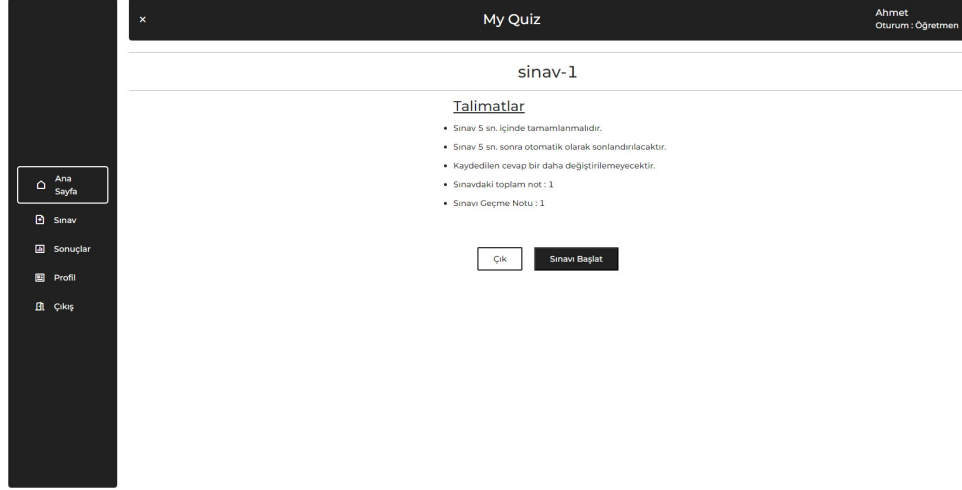
3.2.1 Ana Sayfa



Şekil 13: Öğretmen seviyesindeki kullanıcının Ana Sayfası

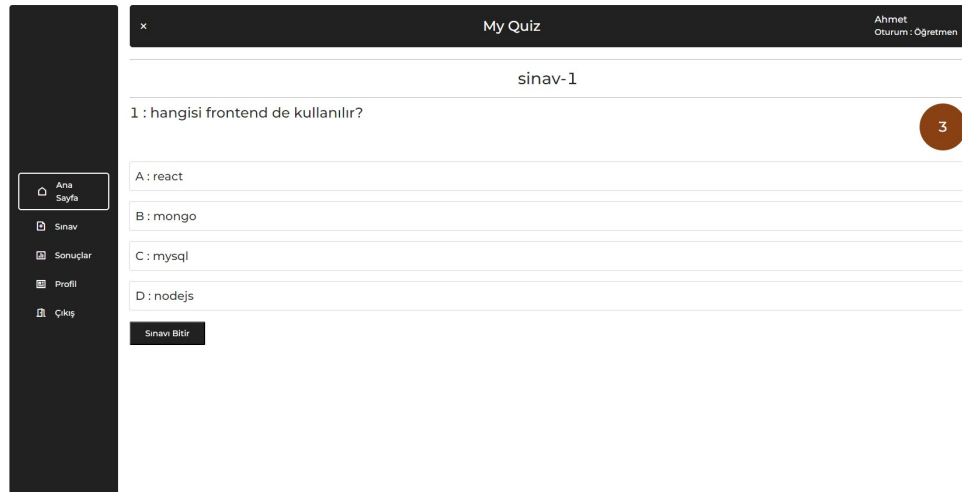
Öğretmen Ana Sayfası’nda, öğretmen seviyesindeki kullanıcının hazırlamış olduğu sınavlar yer almaktadır. Kullanıcı isterse bu sınavlara test amaçlı katılabilir ve sonuçlarını Sonuçlar

sayfasında görebilmektedir. Sayfanın sol tarafında menü bulunmaktadır. Bu menü aracılığı ile kullanıcı farklı sayfalara geçiş yapabilmektedir. Sağ üst tarafta kullanıcının ismi ve onun altında da kullanıcı öğretmen ise öğretmen ve öğrenci ise öğrenci yazarak kullanıcı yetkisi kullanıcının kendisine gösterilmektedir. Kullanıcı eğer sınava giriş yapmak isterse “Sınavı Başlat” butonuna tıklamalıdır.



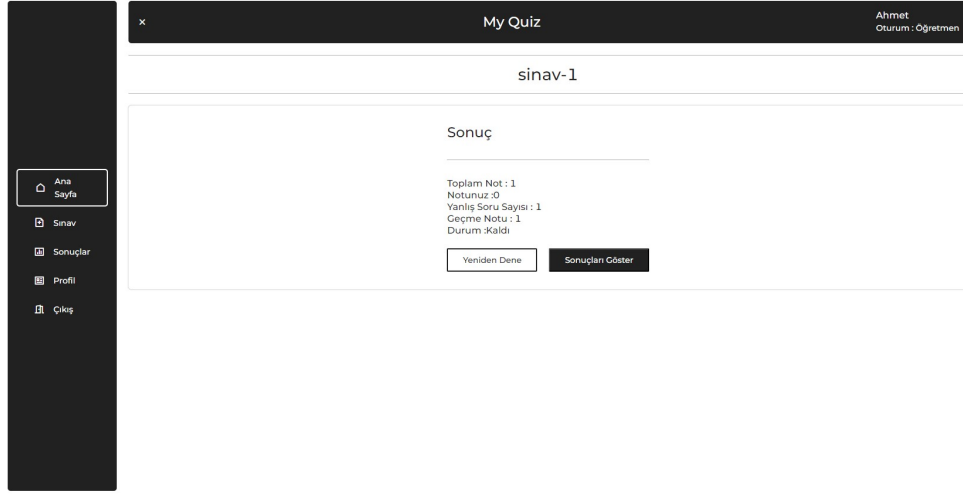
Şekil 14: Öğretmen seviyesindeki kullanıcının sınava giriş ekranı

“Sınavı Başlat” butonuna tıkladıktan sonra kullanıcı Ana Sayfa ekranı üzerinden sınav ekranına yönlendirilir. Kullanıcıya öncelikle bazı sınav talimatları gösterilir. Kullanıcı isterse çıkış yapabilir ya da isterse sınavı başlatabilir.



Şekil 15: Öğretmen seviyesindeki kullanıcının sınav ekranı

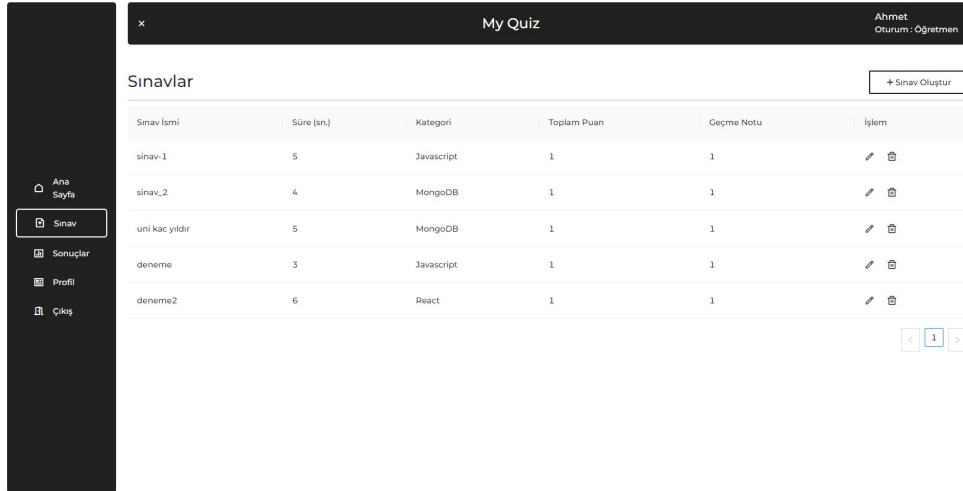
Sınav başlatıldıktan sonra kullanıcının karşısına sınav sorusu seçenekler ile beraber gösterilir. Ve sayfanın sağ üst tarafında sınav için ayrılan toplam süre saniye cinsinden olarak yazar ve geri sayım olarak çalışır. Süre tamamlandığında ise sınav otomatik olarak sonlandırılır.



Şekil 16: Öğretmen seviyesindeki kullanıcının sınav sonuç ekranı

Sınav tamamlandıktan sonra sınav sonucu kullanıcıya ekranda gösterilir. Eğer kullanıcı isterse sınav tekrar girebilir ya da isterse çıkış yapabilir.

3.2.2 Sınav Hazırlama Sayfası



Şekil 17: Öğretmen seviyesindeki kullanıcının Sınav Sayfası

Öğretmen seviyesindeki kullanıcı sınav hazırlamak için Sınav sekmesine tıklar ve Şekil 17’de gösterilen Sınav Sayfası’na yönlendirilir. Bu sayfada öncelikle öğretmenin daha önceden hazırlamış olduğu sınavlar bulunmaktadır. Eğer kullanıcı sınav eklemek isterse ekranda bulunan “Sınav Oluştur” butonuna tıklayarak Sınav Sayfası’nda bulunan sınav oluşturabileceği ekrana yönlendirilir.

My Quiz

Ahmet
Oturum : Öğretmen

Sınav Ekle

[Sınav Detayları](#)

Sınav Adı: Sınav Süresi (sn.): Kategori:

Toplam Not: Geçme Notu:

Şekil 18: Öğretmen seviyesindeki kullanıcının sınav ekleme ekranı

Kullanıcı “Sınav Oluştur” butonuna tıkladıktan sonra Şekil 18’de gösterilen sınav bilgilerini girebileceği ekran gelir. Bilgiler girildikten sonra kullanıcı soruları eklemekten önce “Sınavı Kaydet” butonuna tıklar.

My Quiz

Ahmet
Oturum : Öğretmen

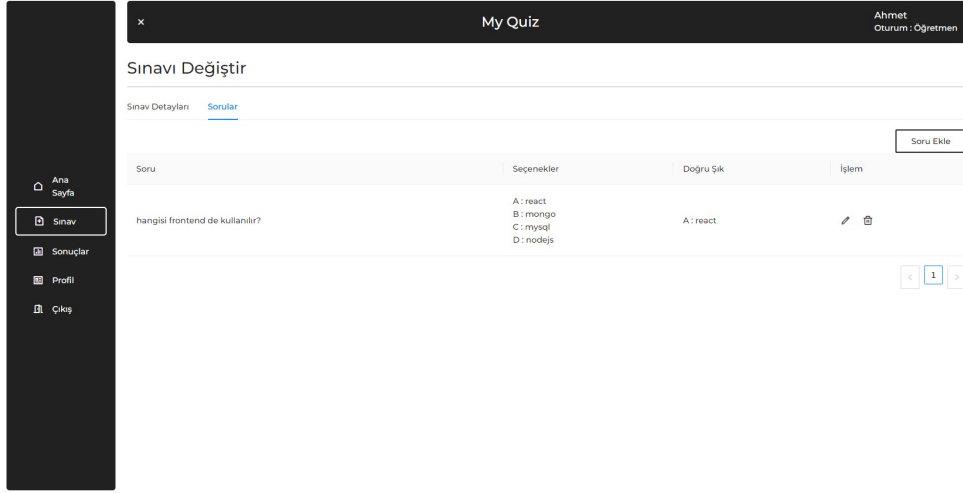
Sınavlar

Sınav İsmi	Süre (sn.)	Kategori	Toplam Puan	Geçme Notu	İşlem
sınav-1	5	Javascript	1	1	<input type="button" value="Düzenle"/> <input type="button" value="Sil"/>
sınav_2	4	MongoDB	1	1	<input type="button" value="Düzenle"/> <input type="button" value="Sil"/>
uni kaç yıldır	5	MongoDB	1	1	<input type="button" value="Düzenle"/> <input type="button" value="Sil"/>
deneme	3	Javascript	1	1	<input type="button" value="Düzenle"/> <input type="button" value="Sil"/>
deneme2	6	React	1	1	<input type="button" value="Düzenle"/> <input type="button" value="Sil"/>

< 1 >

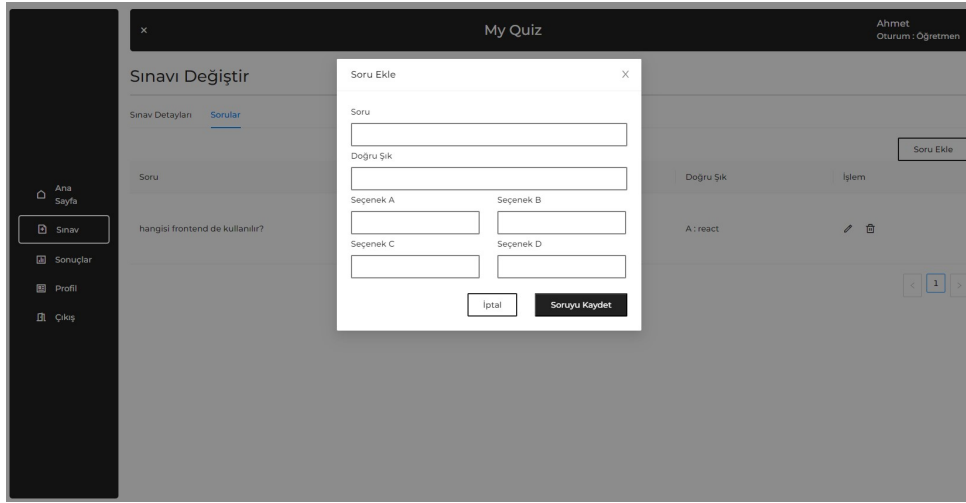
Şekil 19: Öğretmen seviyesindeki kullanıcının oluşturduğu sınavların listesi

Kullanıcı butona tıkladıktan sonra kullanıcı Şekil 19’da gösterilen önceki sayfaya yönlendirilir. Bu sayfada kullanıcı soruları eklemek için ilgili sınavın satırında en sağda bulunan kalem ikonuna tıklayarak sınavı düzenleyebileceği ekrana yönlendirilir.



Şekil 20: Öğretmen seviyesindeki kullanıcının soru ekleme ekranı

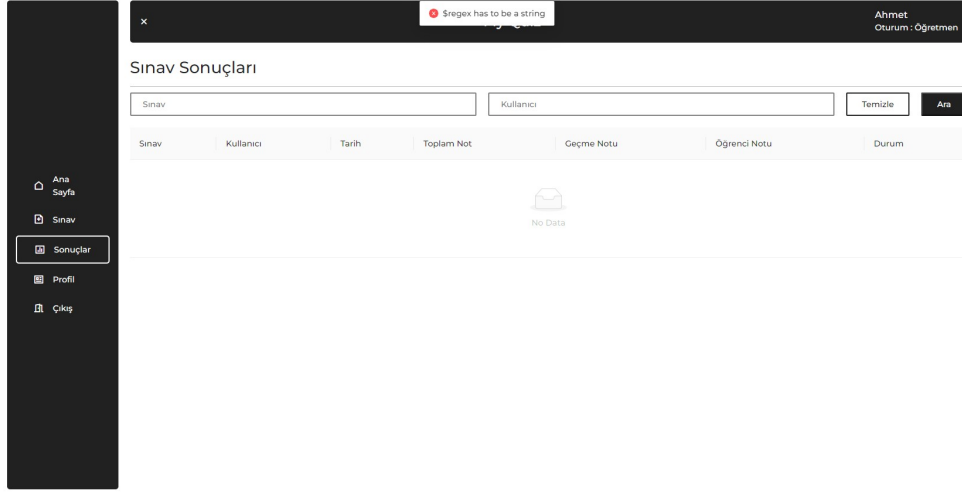
Kullanıcı kalem ikonuna tıkladıktan sonra Şekil 20’de gösterilen ekrana yönlendirilince, sağ üstte bulunan “Soru Ekle” butonuna tıklar. Butona tıklanması sonucu soru yazabileceği ve kaydedebileceği ekran gösterilir.



Şekil 21: Öğretmen seviyesindeki kullanıcının soru oluşturma ekranı

Şekil 21’de gösterilen ekrana bilgiler girildikten sonra soruyu kaydetmek için “Soruyu Kaydet” butonuna tıklanır. Her soru eklemek için aynı işlemler tekrarlanır.

3.2.3 Sınav Sonuçlarını Görüntüleme Sayfası



Şekil 22: Öğretmen seviyesindeki kullanıcının Sınav Sonuç Görüntüleme Sayfası

Bu sayfada kullanıcı öğrencilerin sınav sonuçlarını görüntüler. Ancak projede Sonuç Sayfası hatalı çalışmaktadır. Bu sebeple ekranda sınav sonuçlarını listelenmesi Şekil 22’de gösterildiği gibi başarısızdır.

3.2.4 Profil Sayfası

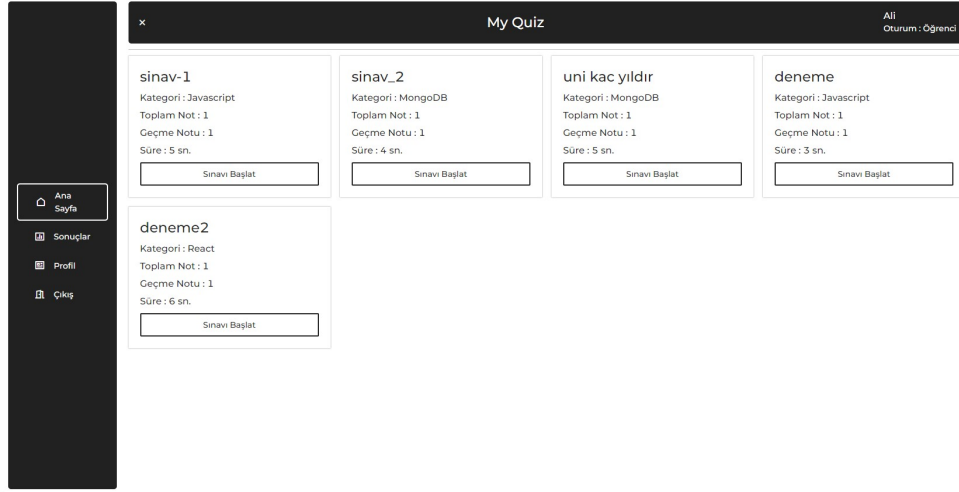
Bu sayfada kullanıcı profil bilgileri olan isim, eposta ve kullanıcı seviyesi gösterilmesi hedeflenmiştir. Ancak projenin bu kısmının hatalı çalışması ve yetiştirilememesi sonucu, profil sayfası gösterilememektedir.

3.2.5 Çıkış

Kullanıcı çıkış yapmak istediğinde sol menüden “Çıkış” butonuna tıklar ve kullanıcı oturumundan çıkış yapılarak kullanıcı Giriş Yap Sayfası’na yönlendirilir.

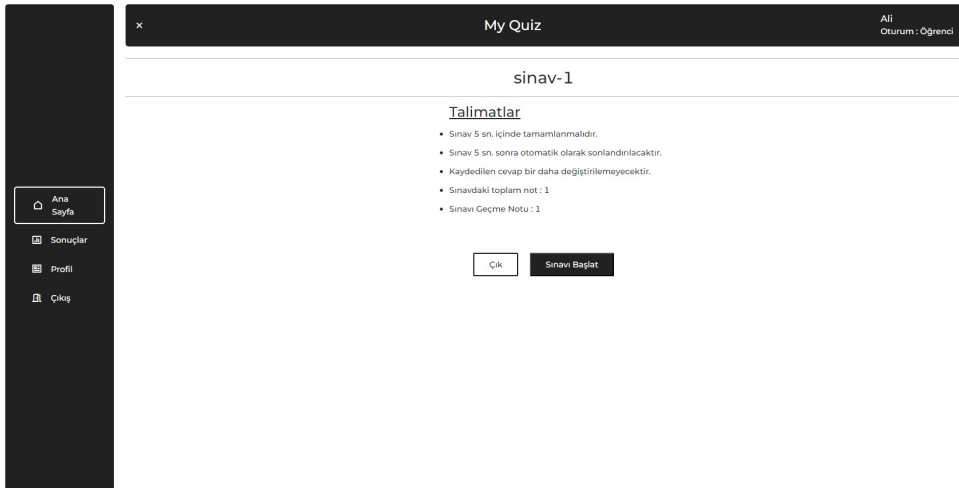
3.3 Öğrenci Seviyesindeki Kullanıcı Girişi

3.3.1 Ana Sayfa



Şekil 23: Öğrenci seviyesindeki kullanıcının Ana Sayfası

Öğrenci Ana Sayfası’nda, öğretmen seviyesindeki kullanıcının hazırlamış olduğu sınavlar yer almaktadır. Kullanıcı buradan sınava katılabilir ve sınavdan sonra sınav başarıları kullanıcıya gösterilir. Sayfanın sol tarafında menü bulunmaktadır. Bu menü aracılığı ile kullanıcı farklı sayfalara geçiş yapabilmektedir. Sağ üst tarafta kullanıcının ismi ve onun altında da kullanıcı öğretmen ise öğretmen ve öğrenci ise öğrenci yazarak kullanıcı yetkisi kullanıcının kendisine gösterilmektedir. Kullanıcı eğer sınava giriş yapmak isterse “Sınavı Başlat” butonuna tıklamalıdır.



Şekil 24: Öğrenci seviyesindeki kullanıcının sınava giriş ekranı

“Sınavı Başlat” butonuna tıkladıktan sonra kullanıcı Ana Sayfa ekranı üzerinden sınav ekranına yönlendirilir. Kullanıcıya öncelikle bazı sınav talimatları gösterilir. Kullanıcı isterse çıkış

yapabilir ya da isterse sınavı başlatabilir.

The screenshot shows a web application titled 'My Quiz' with a user profile 'Ali' and role 'Öğrenci'. The quiz is titled 'sinav-1'. The first question is '1 : hangisi frontend de kullanılır?'. There are four options: A : react, B : mongo, C : mysql, and D : nodejs. A 'Sınavı Bitir' button is at the bottom. A sidebar on the left contains links for 'Ana Sayfa', 'Sonuçlar', 'Profil', and 'Çıkış'. A red circle with the number '3' is in the top right corner.

Şekil 25: Öğrenci seviyesindeki kullanıcının sınav ekranı

Sınav başlatıldıktan sonra kullanıcının karşısına sınav sorusu seçenekler ile beraber gösterilir. Ve sayfanın sağ üst tarafında sınav için ayrılan toplam süre saniye cinsinden olarak yazar ve geri sayım olarak çalışır. Süre tamamlandığında ise sınav otomatik olarak sonlandırılır.

The screenshot shows the 'My Quiz' interface with the title 'sinav-1'. The results section is titled 'Sonuç'. It displays the following information: 'Toplam Not : 1', 'Notunuz : 0', 'Yanlış Soru Sayısı : 1', 'Geçme Notu : 1', and 'Durum :Kaldı'. There are two buttons: 'Yeniden Dene' and 'Sonuçları Göster'. The sidebar on the left is the same as in the previous screenshot.

Şekil 26: Öğrenci seviyesindeki kullanıcının sınav sonuç ekranı

Sınav tamamlandıktan sonra sınav sonucu kullanıcıya ekranda gösterilir. Eğer kullanıcı isterse sınav tekrar girebilir ya da isterse çıkış yapabilir.

3.3.2 Sınav Sonuçlarını Görüntüleme Sayfası

Sınav İsmi	Tarih	Toplam Puan	Geçme Notu	Öğrenci Notu	Durum
sınav-1	11-06-2023 02:48:45	1	1	0	Kaldı
uni kac yıldır	10-06-2023 06:28:26	1	1	1	Geçti
deneme	10-06-2023 06:28:22	1	1	1	Geçti
deneme2	10-06-2023 06:28:06	1	1	0	Kaldı
deneme	10-06-2023 06:26:56	1	1	0	Kaldı
sınav-1	10-06-2023 06:26:23	1	1	0	Kaldı
sınav-1	10-06-2023 05:37:22	1	1	0	Kaldı
sınav-1	10-06-2023 05:37:12	1	1	0	Kaldı
sınav-1	10-06-2023 05:36:36	1	1	1	Geçti
sınav-1	10-06-2023 05:32:48	1	1	1	Geçti

Şekil 27: Öğrenci seviyesindeki kullanıcının Sınav Sonuç Görüntüleme Sayfası

Şekil 27’de gösterilen sayfada öğrenciler girmiş oldukları sınavlara ait bilgilerle birlikte sınav sonuçlarını da listelenmiş olarak görüntüleyebilmektedirler.

3.3.3 Profil Sayfası

Bu sayfada kullanıcı profil bilgileri olan isim, e-posta ve kullanıcı seviyesi gösterilmesi hedeflenmiştir. Ancak projenin bu kısmının hatalı çalışması ve yetiştirilememesi sonucu, profil sayfası gösterilememektedir.

3.3.4 Çıkış

Kullanıcı çıkış yapmak istediğinde sol menüden “Çıkış” butonuna tıklar ve kullanıcı oturumundan çıkış yapılarak kullanıcı Giriş Yap Sayfası’na yönlendirilir.

4 Tasarım Deseni

Tasarım desenleri (design pattern) uygulamalar üzerinde oluşan genel soyutlamalar, soyut ve somut nesneler ve etkileşimleri göstererek temsil edilirler. Tasarım desenleri, çözümün farklı ortamlarda yeniden kullanılabilmesi için yapılan bir tasarımın ya da problemin gerçekleştirilmesi için gereken bir çözümdür. Tasarım deseni açıklama, UML diyagramları ya da her ikisi ile birlikte tanımlanabilir. Bu proje raporunda her ikisi ile birlikte açıklanacaktır. Tasarım deseni açıklanırken 5 temel başlık ile ifade edilebilir:

1. **Desen adı:** Desenin anlaşılabilirliğini arttıracak bir isim
2. **Tanım:** Desenin ne veya neler yapabileceğinin kısa ve öz açıklamasıdır.
3. **Tasarım Amacı:** Bazı kaynaklarda problem tanımı olarak da geçer. Temelde desenin uygulanmasının açıklanmasını ifade eder.
4. **Tasarımın Gerçekleştirilmesi:** Bazı kaynaklarda çözüm tanımı olarak geçer. Uygulama olarak class'ları ve bu class'lar arasındaki ilişkiler gösterilir. UML diyagramları genelde (seçilen desene bağlı olarak) bu aşamayı ifade etmek için oluşturulur.
5. **Sonuç:** Tasarım deseni ifade edilirken ve uygulanırken gerçekleştirilenlerin bir sonucunu ya da özetini ifade eder.

şeklindeki başlıklar ile ifade edilebilir.[1]

MyQuiz sisteminde tasarım deseni olarak davranışsal tasarım desenlerinden olan Gözlemci Tasarım Deseni (Observer Design Pattern) kullanılmıştır. Bu tasarım deseni bir nesnede yapılan değişiklik ya da etkileşim sonucu bir çok nesnede meydana gelecek değişiklikler ya da etkilenmeler olduğunda kullanılır. Gözlemci Tasarım Deseni özne ve gözlemci olarak iki kısımdan oluşur. Ve hem öznenin hem de gözlemciden, kalıtım ile somut nesne elde edilebilir. Bu, UML diyagramlarında ayrıntıyı göstermeye imkan tanır.

Gözlemci Tasarım Deseni birden çoğa bir ilişkiyi ifade eder. Bu durumu yaygın olarak açıklayan örneklerden biri otobüs şirketi ve müşteri örneğidir. Bu örnekte otobüs şirketinin sistemde yaptığı değişiklik (yani güncelleme) ve etki tüm müşteriler tarafından anlık olarak görüntülenebilmektedir. Aynı şekilde bir veya birden çok kullanıcı tek bir sistemde (otobüs şirketi sistemi) talep oluşturabilir ya da sistemden yanıt alabilir. Ve bir yazılım sistemi kendi içerisinde dahili olarak bir çok yazılım sistemini barındırabilir. Aynı şekilde bir çok kullanıcı farklı sınıflandırmalara dahil olabilir. Bunlar somut nesneler ile ifade edilir. Somut nesneler, nesnelerden kalıtım yolu ile oluşturulur. Burada özne otobüs sistemi ve gözlemci kullanıcılar olduğundan dolayı, otobüs sisteminin barındırdığı yazılımsal sistemler de somut özne ve farklı sınıflara ve yetkilere ayrılmış kullanıcılar da somut gözlemcileri ifade etmektedir.

Somut özne ve somut nesnenin kalıtım yolu ile özne ve nesneden türetildiği belirtilmişti. Ancak bazı diller ya da bazı dillere ait kütüphaneler kalıtımı doğrudan bir yöntem olarak kullanmak

yerine, farklı yollardan kalıtımı taklit eden yöntemler geliştirmişlerdir. Bu şekilde yapmaktaki amaç kodun sadeliğini ve hızını arttırmaktır. Javascript bu dillerden biridir. Javascript'in kendi yapısında kalıtımı sağlayan basit yapılar bulunmaktadır. Ancak MyQuiz sisteminde kullanılan yöntem bu basit yapılar olmadığından dolayı, bu raporda o yöntemlerden bahsedilmemesine karar kılınmıştır.

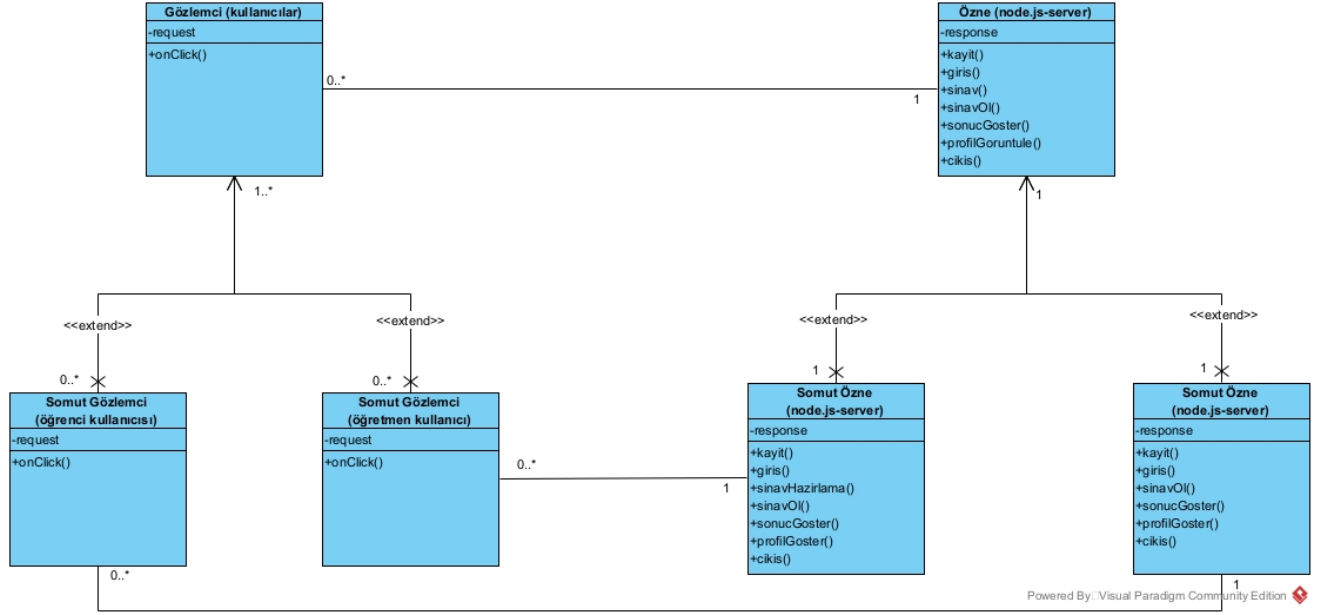
MyQuiz sistemi Javascript kütüphaneleri olan React ve Node.js'i kullanmaktadır. React kütüphanesinde kalıtımı taklit etmek için props ve hook yapıları, Node.js kütüphanesinde ise require() metodu ile birlikte kullanılan modül yapıları bulunmaktadır. Dolayısıyla MyQuiz ve bahsedilen kütüphaneler kullanılan sistemler kalıtımı uygulamak için ifade edilen yöntemleri kullanacaktır.

MyQuiz sisteminin Gözlemci Tasarım Deseni Tablo 1'de belirtilmektedir:

Başlık	Açıklama
Desen Adı	Gözlemci
Tanım	Etki eden ya da işlemi başlatan taraf (gerçek kullanıcılar da olabilir) ile etkilenen ya da işlemi gerçekleştiren tarafın (server gibi) birbirinden ayrıştırılması sağlanmalıdır. Bu tasarım deseninde işlemi başlatan taraf gözlemci ve işlemi gerçekleştiren taraf öznedir. Bu tablo ve gerçekleştirilen UML diyagramında gözlemci olarak projenin arayüz tasarımı ve bu tasarıma etki eden kullanıcılar seçilmiştir. Seçilen bu gözlemciler, özne şeklinde tanımlanarak işlemleri gerçekleştiren taraf olan server'ı (yani node.js kısmını) ifade etmektedir.
Tasarım Amacı	Gözlemci, özne ile iletişime geçerek sistemin güncelliğini interaktif biçimde gözlemci ve özne sayısından bağımsız olarak gerçekleştirmesini sağlar. Problem gözlemci ve öznenin sayısında değildir. Problem, gözlemci ve öznenin çeşitliliğe (sınıflandırmaya) sahip olmasındadır. Mesela MyQuiz sisteminde, öğrenci ve öğretmen olarak iki farklı arayüz ve server yönünden iki farklı oturum bulunmaktadır.
Tasarımın Gerçekleştirilmesi	MyQuiz sisteminin iki farklı oturumları, observer (gözlemci) tasarım deseninde somut gözlemci ve somut özne ile ifade edilebilmektedir. Bu yöntemin gerçekleştirilmesi aşağıdaki UML diyagramında gösterilmektedir.
Sonuç	Tasarım deseni gerçekleştirilirken, gözlemci olarak arayüz ve özne olarak server ifade edilmiştir. Sistemde özne ve gözlemci çeşitliliği, somut özne ve somut gözlemci ile ifade edilmiştir. Sisteme katılan kullanıcı sayısında limit olmadığından gözlemciler 0..* ya da 1..* ile ifade edilmiştir. Sistemde kullanılan özne (server) öğrenci ve öğretmen için iki sınıfa ayrılrsa da server temelde tek olduğundan 1..1 ile ifade edilmiştir.

Tablo 1: MyQuiz sistemi Gözlemci Tasarım Deseni

Tablo 1’de bulunan ”Tasarımın Gerçekleştirilmesi” aşamasında ifade edilen UML diyagramı, MyQuiz sistemi için Şekil’de görülmektedir:



Şekil 28: MyQuiz sistemi Gözlemci Tasarım Deseni UML Diyagramı

5 MyQuiz Sisteminin Kodlarının GitHub Linki

<https://github.com/ayg55/MyQuiz-App.git>

6 MyQuiz Sisteminin Yazılım Tanıtım Videosu Linki

<https://youtu.be/AY9iGiPg1V0>

Kaynaklar

- [1] I. Sommerville. *Yazılım Mühendisliği*. Nobel, Inc., 10rd edition, 2018.
- [2] Stackoverflow en çok kullanılan geliştirme araçları anketi. <https://survey.stackoverflow.co/2022/#section-most-popular-technologies-integrated-development-environment>.