# Sprint 1 - 2. Hafta Raporu

**(Yapılanlar ve Kullanılan Kodlar)**

**1. Sayfalar Arası Geçiş ve Menü (Emir Avcı)**

* **Ne yapıldı:**

React ile ana sayfa, rezervasyon, hakkımızda ve iletişim sayfaları arasında geçiş sağlandı. Üstte bir menü oluşturuldu.

* **Kullanılan Kod:**

Kullanıcılar menüden tıklayarak sayfalar arasında geçiş yapabilir.

// client/src/App.js

import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';

function App() {

return (

<Router>

<nav>

<ul>

<li><Link to="/">Ana Sayfa</Link></li>

<li><Link to="/rezervasyon">Rezervasyon</Link></li>

<li><Link to="/hakkimizda">Hakkımızda</Link></li>

<li><Link to="/iletisim">İletişim</Link></li>

</ul>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/rezervasyon" element={<Rezervasyon />} />

<Route path="/hakkimizda" element={<Hakkimizda />} />

<Route path="/iletisim" element={<Iletisim />} />

</Routes>

</Router>

);

}

**2. Hata Yakalama Sistemi (Kayra Doğan)**

* **Ne yapıldı:**

React’ta uygulama çökerse kullanıcıya hata mesajı gösteriliyor.

* **Kullanılan Kod:**

Uygulamada beklenmeyen bir hata olursa kullanıcıya kırmızı bir uyarı gösterir.

1. // client/src/ErrorBoundary.js
2. class ErrorBoundary extends React.Component {
3. // ...
4. render() {
5. if (this.state.hasError) {
6. return <div>Bir hata oluştu!</div>;
7. }
8. return this.props.children;
9. }
10. }

**3. Veritabanına Örnek Veri Ekleme (Emir Demirtaş)**

* **Ne yapıldı:**

MongoDB’ye örnek kullanıcı ve rezervasyon verileri eklendi.

* **Kullanılan Kod :**

Sistemi test etmek için veritabanına örnek kullanıcı ve rezervasyon ekler.

1. // server/testData.js
2. const User = require('./models/User');
3. const Reservation = require('./models/Reservation');
4. const user = await User.create({
5. isim: 'Test Kullanıcı',
6. email: 'test@test.com',
7. sifre: 'test123',
8. telefon: '5551234567'
9. });
10. const reservation = await Reservation.create({
11. halisaha: 'halisaha-id',
12. kullanici: user.\_id,
13. tarih: new Date(),
14. saat: '18:00'
15. });

**4. Basit Testler Yazma (Çağrı Doğan)**

* **Ne yapıldı:**

API’lerin çalışıp çalışmadığını kontrol eden otomatik testler yazıldı.

* **Kullanılan Kod:**

Sunucunun doğru cevap verip vermediğini otomatik olarak kontrol eder.

1. // server/tests/api.test.js
2. const request = require('supertest');
3. const app = require('../app');
4. test('GET / - Ana sayfa mesajını kontrol et', async () => {
5. const response = await request(app).get('/');
6. expect(response.statusCode).toBe(200);
7. expect(response.body).toHaveProperty('message', 'Merhaba Dünya!');
8. });