# Sprint 1 - 5. Hafta Raporu

1-Emir Avcı (Backend API Geliştirme)

Ne Yapıldı:

Rezervasyon API endpoint'leri oluşturuldu, CRUD işlemleri eklendi. Aşağıdaki kodda rezervasyonların listelenmesi ve yeni rezervasyon eklenmesi için Express route'ları yer alıyor.

Kod: javascript

1. const express = require('express');
2. const router = express.Router();
3. const Rezervasyon = require('../models/Rezervasyon');
4. // Tüm rezervasyonları getir
5. router.get('/', async (req, res) => {
6. try {
7. const rezervasyonlar = await Rezervasyon.find()
8. .sort({ tarih: 1, saat: 1 });
9. res.json(rezervasyonlar);
10. } catch (error) {
11. res.status(500).json({ message: error.message });
12. }
13. });
14. // Yeni rezervasyon oluştur
15. router.post('/', async (req, res) => {
16. const rezervasyon = new Rezervasyon({
17. tarih: req.body.tarih,
18. saat: req.body.saat,
19. isim: req.body.isim,
20. telefon: req.body.telefon,
21. email: req.body.email,
22. durum: req.body.durum || 'beklemede'
23. });
24. try {
25. const yeniRezervasyon = await rezervasyon.save();
26. res.status(201).json(yeniRezervasyon);
27. } catch (error) {
28. if (error.code === 11000) {
29. res.status(400).json({ message: 'Bu tarih ve saat için zaten bir rezervasyon var' });
30. } else {
31. res.status(400).json({ message: error.message });
32. }
33. }
34. });

2-Kayra Doğan (Frontend Form Geliştirme)

Ne Yapıldı:

Rezervasyon formu React ile oluşturuldu, form validasyonu ve hata yönetimi eklendi. Kodda formun temel yapısı ve gönderim işlemi yer alıyor.

Kod: javascript

1. import React, { useState } from 'react';
2. import axios from 'axios';
3. import { useNavigate } from 'react-router-dom';
4. const RezervasyonForm = () => {
5. const navigate = useNavigate();
6. const [formData, setFormData] = useState({
7. tarih: '',
8. saat: '',
9. isim: '',
10. telefon: '',
11. email: ''
12. });
13. const [error, setError] = useState('');
14. const handleSubmit = async (e) => {
15. e.preventDefault();
16. setError('');
17. try {
18. await axios.post('http://localhost:5000/api/rezervasyonlar', formData);
19. navigate('/rezervasyonlar');
20. } catch (err) {
21. setError(err.response?.data?.message || 'Bir hata oluştu');
22. }
23. };
24. return (
25. <div className="max-w-md mx-auto mt-8 p-6 bg-white rounded-lg shadow-lg">
26. <h2 className="text-2xl font-bold mb-6 text-center">Yeni Rezervasyon</h2>
27. {/\* Form içeriği \*/}
28. </div>
29. );
30. };

3-Emir Demirtaş (Veritabanı Modeli)

Ne Yapıldı:

MongoDB için rezervasyon şeması oluşturuldu, indeksler ve veri doğrulama eklendi. Kodda şema tanımı ve indeksler yer alıyor.

Kod: javascript

1. const mongoose = require('mongoose');
2. const rezervasyonSchema = new mongoose.Schema({
3. tarih: {
4. type: Date,
5. required: true
6. },
7. saat: {
8. type: String,
9. required: true
10. },
11. isim: {
12. type: String,
13. required: true
14. },
15. telefon: {
16. type: String,
17. required: true
18. },
19. email: {
20. type: String,
21. required: true
22. },
23. durum: {
24. type: String,
25. enum: ['beklemede', 'onaylandı', 'iptal'],
26. default: 'beklemede'
27. }
28. }, {
29. timestamps: true
30. });
31. // İndeksler
32. rezervasyonSchema.index({ tarih: 1, saat: 1 }, { unique: true });
33. rezervasyonSchema.index({ email: 1 });
34. rezervasyonSchema.index({ durum: 1 });
35. rezervasyonSchema.index({ createdAt: -1 }); // Tarihe göre sıralama için
36. rezervasyonSchema.index({ durum: 1, tarih: 1 }); // Durum ve tarihe göre sorgulama için
37. // Veri doğrulama
38. rezervasyonSchema.pre('save', function(next) {
39. if (this.isNew) {
40. this.createdAt = new Date();
41. }
42. next();
43. });
44. module.exports = mongoose.model('Rezervasyon', rezervasyonSchema);

Çağrı Doğan (Sunucu Yapılandırması)

Ne Yapıldı:

Express sunucusu kuruldu, gerekli middleware'ler ve hata yakalama eklendi. Kodda sunucu yapılandırması ve route tanımlamaları yer alıyor.

Kod: javascript

1. const express = require('express');
2. const mongoose = require('mongoose');
3. const cors = require('cors');
4. const rezervasyonRoutes = require('./routes/rezervasyon');
5. const app = express();
6. // Middleware
7. app.use(cors());
8. app.use(express.json());
9. // MongoDB bağlantısı
10. mongoose.connect('mongodb://localhost:27017/halisaha', {
11. useNewUrlParser: true,
12. useUnifiedTopology: true
13. });
14. // Routes
15. app.use('/api/rezervasyonlar', rezervasyonRoutes);
16. // Hata yakalama
17. app.use((err, req, res, next) => {
18. console.error(err.stack);
19. res.status(500).json({ message: 'Bir şeyler ters gitti!' });
20. });