**Sprint 1 - 3. Hafta Raporu**

1. React ile Sayfa Yönlendirme ve Form Yapısı (Emir Avcı)

* Ne yapıldı:

React Router kurulumu yapıldı, sayfa yönlendirmeleri eklendi, rezervasyon formu oluşturuldu.

* Kullanılan Kodlar:

App.js:

1. import React from 'react';
2. import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
3. import Navbar from './components/Navbar';
4. import Home from './pages/Home';
5. import Rezervasyon from './pages/Rezervasyon';
6. import Hakkimizda from './pages/Hakkimizda';
7. import Iletisim from './pages/Iletisim';
8. function App() {
9. return (
10. <Router>
11. <div className="App">
12. <Navbar />
13. <Routes>
14. <Route path="/" element={<Home />} />
15. <Route path="/rezervasyon" element={<Rezervasyon />} />
16. <Route path="/hakkimizda" element={<Hakkimizda />} />
17. <Route path="/iletisim" element={<Iletisim />} />
18. </Routes>
19. </div>
20. </Router>
21. );
22. }
23. export default App;

2- Rezervasyon.js:

1. import React, { useState } from 'react';
2. import './Rezervasyon.css';
3. function Rezervasyon() {
4. const [formData, setFormData] = useState({
5. tarih: '',
6. saat: '',
7. isim: '',
8. telefon: '',
9. email: ''
10. });
11. const handleSubmit = (e) => {
12. e.preventDefault();
13. // API'ye gönderme işlemi burada yapılacak
14. console.log('Form verileri:', formData);
15. };
16. return (
17. <div className="rezervasyon-container">
18. <h2>Rezervasyon Yap</h2>
19. <form onSubmit={handleSubmit}>
20. <div className="form-group">
21. <label>Tarih:</label>
22. <input
23. type="date"
24. value={formData.tarih}
25. onChange={(e) => setFormData({...formData, tarih: e.target.value})}
26. required
27. />
28. </div>
29. <div className="form-group">
30. <label>Saat:</label>
31. <input
32. type="time"
33. value={formData.saat}
34. onChange={(e) => setFormData({...formData, saat: e.target.value})}
35. required
36. />
37. </div>
38. <div className="form-group">
39. <label>İsim:</label>
40. <input
41. type="text"
42. value={formData.isim}
43. onChange={(e) => setFormData({...formData, isim: e.target.value})}
44. required
45. />
46. </div>
47. <div className="form-group">
48. <label>Telefon:</label>
49. <input
50. type="tel"
51. value={formData.telefon}
52. onChange={(e) => setFormData({...formData, telefon: e.target.value})}
53. required
54. />
55. </div>
56. <div className="form-group">
57. <label>E-posta:</label>
58. <input
59. type="email"
60. value={formData.email}
61. onChange={(e) => setFormData({...formData, email: e.target.value})}
62. required
63. />
64. </div>
65. <button type="submit">Rezervasyon Yap</button>
66. </form>
67. </div>
68. );
69. }
70. export default Rezervasyon;
71. Node.js ile Rezervasyon API'si (Kayra Doğan)

* Ne yapıldı:

Rezervasyon modeli oluşturuldu, API endpoint'leri eklendi, veritabanı işlemleri yapılandırıldı.

* Kullanılan Kodlar:

1- models/Rezervasyon.js:

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. module.exports = mongoose.model('Rezervasyon', rezervasyonSchema);

2- routes/rezervasyon.js:

1. const express = require('express');
2. const router = express.Router();
3. const Rezervasyon = require('../models/Rezervasyon');
4. // Yeni rezervasyon oluştur
5. router.post('/', async (req, res) => {
6. try {
7. const rezervasyon = new Rezervasyon(req.body);
8. await rezervasyon.save();
9. res.status(201).json(rezervasyon);
10. } catch (error) {
11. res.status(400).json({ message: error.message });
12. }
13. });
14. // Tüm rezervasyonları getir
15. router.get('/', async (req, res) => {
16. try {
17. const rezervasyonlar = await Rezervasyon.find();
18. res.json(rezervasyonlar);
19. } catch (error) {
20. res.status(500).json({ message: error.message });
21. }
22. });
23. module.exports = router;
24. MongoDB Şema ve İndeksler (Emir Demirtaş)

* Ne yapıldı:

Rezervasyon şeması için indeksler oluşturuldu, veritabanı optimizasyonu yapıldı.

* Kullanılan Kodlar:

1. models/Rezervasyon.js (indeksler eklendi):

1// ... mevcut şema tanımı ...

1. // İndeksler
2. rezervasyonSchema.index({ tarih: 1, saat: 1 }, { unique: true });
3. rezervasyonSchema.index({ email: 1 });
4. rezervasyonSchema.index({ durum: 1 });
5. // ... model export ...
6. Dokümantasyon Güncellemesi (Çağrı Doğan)

* Ne yapıldı:

API dokümantasyonu eklendi, kurulum adımları güncellendi.

* Kullanılan Kodlar:

1. docs/API.md:

1# API Dokümantasyonu

1. ## Rezervasyon Endpoint'leri
2. ### POST /api/rezervasyon
3. Yeni rezervasyon oluşturur.
4. Request Body:
5. ```json
6. {
7. "tarih": "2024-03-20",
8. "saat": "19:00",
9. "isim": "Ahmet Yılmaz",
10. "telefon": "5551234567",
11. "email": "ahmet@example.com"
12. }
13. ```
14. ### GET /api/rezervasyon
15. Tüm rezervasyonları listeler.