TECHCAREER NODE.JS BOOTCAMP SUNUMU

Merhaba, Ben Resul Gencer

Merhaba, ben Resul Gencer. Zonguldak Bülent Ecevit Üniversitesi Bilgisayar Mühendisliği Bölümü'nden mezun oldum. Yazılım geliştirme alanına olan ilgim, üniversite eğitimimle birlikte şekillendi ve bugüne kadar farklı platformlarda projeler geliştirme fırsatım oldu.

Web, mobil ve backend teknolojilerinde projeler geliştirdim. HTML, CSS ve JavaScript gibi temel web teknolojilerinin yanı sıra, React gibi modern web kütüphaneleriyle çalıştım. Mobil uygulama geliştirme tarafında ise, Flutter ile kullanıcı dostu ve performanslı mobil uygulamalar geliştirdim. Ayrıca, backend tarafında Node.js ve Firebase gibi güçlü araçlarla dinamik ve ölçeklenebilir uygulamalar oluşturma deneyimim oldu.

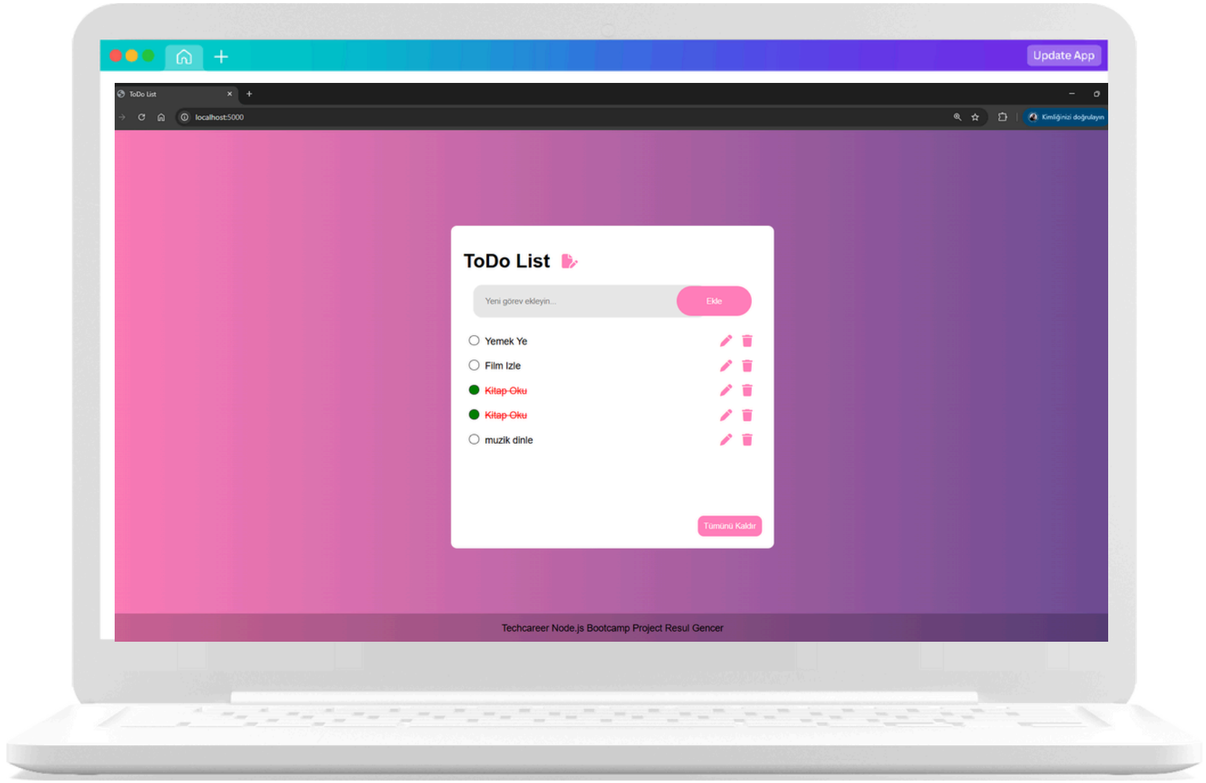
Son 6-7 aydır özellikle mobil uygulama geliştirme alanında kendimi geliştirmeye odaklandım. Flutter ile mobil projeler üzerinde çalışarak, bu alandaki becerilerimi geliştiriyorum.

Bu sunumda sizlere geliştirdiğim To-Do List (Yapılacaklar Listesi) Uygulaması hakkında detaylı bilgiler sunacağım.

ToDo List

Günümüzde kişisel ve profesyonel yaşamda görev yönetimi büyük bir önem taşımaktadır. Bu nedenle, kullanıcıların günlük işlerini organize edebileceği, tamamlanan görevleri işaretleyebileceği ve görevlerini belirli kategorilere ayırabileceği bir To-Do List (Yapılacaklar Listesi) Uygulaması geliştirdim.

Bu proje ile kullanıcılar görevlerini planlayabilir, önceliklendirebilir ve zaman yönetimlerini daha verimli bir hale getirebilirler.



Görev Yönetimi: Görev ekleme, güncelleme ve silme işlemleri.

Bulut Tabanlı Veri Saklama: Firebase Firestore ile veri kaybı önlenir.

Tamamlanan Görevleri İşaretleme: Görevlerin durumlarının takip edilmesi.

Gerçek Zamanlı Güncellenme: Verilerin anlık olarak eşitlenmesi.

Kullanılan Teknolojiler

Backend Teknolojileri

- Node.js & Express.js ile REST API geliştirme
- Firebase Firestore ile Gerçek zamanlı veri saklama
- Body-Parser & Cors ile API isteklerini yönetme

Frontend Teknolojileri

- HTML, CSS, JavaScript ile dinamik bir kullanıcı arayüzü
- EJS (Embedded JavaScript) ile şablon yönetimi

Proje Mimarisi ve Çalışma Mantığı

Kullanıcı yeni bir görev ekler.

Veri Firebase Firestore’a kaydedilir.

Tüm görevler anlık olarak güncellenir.

Tamamlanan görevler işaretlenebilir ve gerektiğinde silinebilir.

Backend API Endpoint’leri

HTTP Metodu	Endpoint	Açıklama
GET	/	Tüm görevleri getirir.
POST	/add	Yeni görev ekler.
PUT	/edit/:id	Belirtilen görevi günceller.
DELETE	/delete/:id	Belirtilen görevi siler.
DELETE	/delete-all	Tüm görevleri temizler.

Örnek Kodlar

```
todoForm.addEventListener("submit", async (e) => {
       e.preventDefault();
       const name = todoInput.value.trim();

       if (!name) {
           errorMessage.style.display = "block";
           setTimeout(() => {
               errorMessage.style.display = "none";
           }, 5000);
           return;
       }

       const response = await fetch("/add", {
           method: "POST",
           headers: { "Content-Type": "application/json" },
           body: JSON.stringify({ name, isChecked: false, date: new
Date().toISOString() }),
       });

       if (response.ok) {
           location.reload();
       }
   });
```

Kullanıcı, yeni görev eklemek için bir form kullanıyor. Form verisi, backend'e gönderiliyor ve Firebase'e ekleniyor.
Bu kısımda, kullanıcıdan alınan görev bilgisi JSON formatında backend'e gönderiliyor ve Firebase Firestore'a kaydediliyor.

```
<div class="todo-list">
        <ul id="todoList">
            <% todos.forEach(todo => { %>
            <li data-id="<%= todo.id %>">
                <div class="todo-item">
                    <label class="round-checkbox">
                        <input type="checkbox" class="todo-checkbox"
 <%= todo.IsChecked ? 'checked' : '' %>>
                        <span></span>
                    </label>
                    <label class="todo-name <
%= todo.IsChecked ? 'completedTask' : '' %>"><%= todo.Name %></label
>
                </div>
                <div class="todo-item-actions">
                    <button class="edit-btn">
                        <i class="fa-solid fa-pen fa-xl" style="
color: #ff7eb9;"></i>
                    </button>
                    <button class="delete-btn">
                        <i class="fa-solid fa-trash fa-xl" style="
color: #ff7eb9;"></i>
                    </button>
                </div>
            </li>
            <% }) %>
        </ul>
    </div>
```

HTML ve EJS şablonları kullanılarak, Firebase'den çekilen görevler kullanıcıya dinamik olarak sunuluyor.
Bu kısımda, todos dizisi döngüye sokularak her bir görev listeleniyor. Eğer görev tamamlandıysa (IsChecked true ise), görev ismi üzerine "completedTask" sınıfı ekleniyor ve bu da CSS ile görsel olarak işaretleniyor.

```
import express from "express";
import {db} from "../firebase";
import { collection, getDocs, addDoc, deleteDoc, doc,
updateDoc, writeBatch } from "firebase/firestore";
import { Request, Response } from "express";

const router = express.Router();

const todoCollection = collection(db,"Todo");

router.get("/",async (req: Request, res: Response) => {
  try{
    const snapshot = await getDocs(todoCollection);
    const todos : any[] = [];
    snapshot.forEach((doc) => {
      todos.push({id: doc.id, ...doc.data()});
    });
    res.render("todo",{todos});
  }
  catch(err){
    res.status(500).send(err);
  }
  });
```

Aşağıdaki kodda, Firebase Firestore'dan görevler çekilmekte ve kullanıcıya sunulmaktadır. getDocs() fonksiyonu ile veriler alınıyor ve bunlar todos dizisine aktarılıyor.
Bu kod, Firestore'dan tüm görevleri çekiyor ve endpoint oluşturuyor. Frontend tarafında bu veriler, kullanıcıya gösterilmek üzere listeleniyor.

GELECEK GELIŞTIRME PLANLARI

Bu proje şu an temel bir To-Do List uygulaması olarak çalışmaktadır. Gelecekte şu ek özellikler planlanmaktadır:

Kullanıcı Kimlik Doğrulama:
Kullanıcıların kişisel verilerini güvenle saklayabilmesi için kimlik doğrulama sistemi

Kategori Bazlı Görev Yönetimi:
Kullanıcıların görevlerini farklı kategoriler altında düzenleyebilmesi

Hatırlatıcı & Bildirimler:
Görevlerin belirli zamanlarda hatırlatılması için bildirim desteği

Mobil Uygulama Geliştirme:
Uygulamanın Flutter kullanılarak mobil versiyonunun oluşturulması

Beni dinlediğiniz için teşekkür ederim.

RESUL GENCER