**Meeting Organizer Çalışması**

**Hazırlayan :** Uğur Berkay Güneydaş

**Tarih:** 20.02.2024

**Proje Çıktısı :** Bir veya birden fazla toplantının kayıt edilebileceği, düzenlenebileceği ve silebeleceği MechSoft Toplantı Uygulaması Projesinin son görünümüdür.A screenshot of a meeting organizer

Description automatically generated

Figure 1 Proje Görünüm

**Tarih Seçme:**

A screenshot of a computer

Description automatically generated

Figure 2 Kullanıcının Toplantı tarihini seçmek için kullanacağı bölüm

**Saat Seçimi:**

A screenshot of a computer

Description automatically generated

Figure 3 Kullanıcının Toplantı saatini seçmek için kullanacağı bölüm

**Genel Görünüm:** A screenshot of a computer

Description automatically generated

Figure 4 Örnek olarak eklenmiş birkaç toplantı

Toplantı Lisetindeki Bir Toplantının Düzenlenmesi: Örnek olarak deneme 3 numune olarak seçilmiştir.

A blue screen with black text

Description automatically generated

A blue screen with black text

Description automatically generated

**Proje Teknolojisi:**

Bu proje python(flask) ve html ile geliştirilmiştir.

**Proje Kodları:**

Python:

from flask import Flask, render\_template, request, jsonify

app = Flask(\_\_name\_\_)

meetings = [

]

@app.route('/')

def index():

    return render\_template('index.html', meetings=meetings)

@app.route('/add\_meeting', methods=['POST'])

def add\_meeting():

    new\_meeting = {

        "id": len(meetings) + 1,

        "topic": request.form.get('topic'),

        "date": request.form.get('date'),

        "start\_time": request.form.get('start\_time'),

        "end\_time": request.form.get('end\_time'),

        "participants": request.form.get('participants').split(',')

    }

    meetings.append(new\_meeting)

    return jsonify({"status": "success"})

@app.route('/update\_meeting/<int:meeting\_id>', methods=['PUT'])

def update\_meeting(meeting\_id):

    meeting = next((m for m in meetings if m['id'] == meeting\_id), None)

    if meeting:

        meeting.update(request.json)

        return jsonify({"status": "success"})

    else:

        return jsonify({"status": "error", "message": "Meeting not found"})

@app.route('/delete\_meeting/<int:meeting\_id>', methods=['DELETE'])

def delete\_meeting(meeting\_id):

    global meetings

    meetings = [m for m in meetings if m['id'] != meeting\_id]

    return jsonify({"status": "success"})

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

Html kodları:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>MechSoft Meeting Organizer </title>

</head>

<body>

    <body style="background-color:#7796c7"></body>

    <h1 style ="text-align:center;">MechSoft Meeting Organizer</h1>

    <form id="meetingForm">

        <label for="topic">Toplantı Konusu:</label>

        <input type="text" id="topic" name="topic" required>

        <label for="date">Tarih:</label>

        <input type="date" id="date" name="date" required>

        <label for="start\_time">Başlangıç Saati:</label>

        <input type="time" id="start\_time" name="start\_time" required>

        <label for="end\_time">Bitiş Saati:</label>

        <input type="time" id="end\_time" name="end\_time" required>

        <label for="participants">Katılımcılar (virgülle ayırın):</label>

        <input type="text" id="participants" name="participants" required>

        <button type="button" onclick="addMeeting()">Toplantı Ekle</button>

    </form>

    <h2>Toplantı Listesi</h2>

    <ul id="meetingsList">

        {% for meeting in meetings %}

            <li>

                <strong>{{ meeting['topic'] }}</strong> - {{ meeting['date'] }}, {{ meeting['start\_time'] }}-{{ meeting['end\_time'] }}, Katılımcılar: {{ meeting['participants'] }}

                <button type="button" onclick="editMeeting('{{ meeting.id }}')">Düzenle</button>

                <button type="button" onclick="deleteMeeting('{{ meeting.id }}')">Sil</button>

            </li>

        {% endfor %}

    </ul>

    <div id="editForm" style="display: none;">

        <h2>Toplantı Güncelleme Formu</h2>

        <form id="updateForm">

            <label for="editTopic">Toplantı Konusu:</label>

            <input type="text" id="editTopic" name="editTopic" required><br>

            <label for="editDate">Tarih:</label>

            <input type="date" id="editDate" name="editDate" required><br>

            <label for="editStartTime">Başlangıç Saati:</label>

            <input type="time" id="editStartTime" name="editStartTime" required><br>

            <label for="editEndTime">Bitiş Saati:</label>

            <input type="time" id="editEndTime" name="editEndTime" required><br>

            <label for="editParticipants">Katılımcılar:</label>

            <input type="text" id="editParticipants" name="editParticipants" required><br>

            <button type="button" onclick="updateMeeting()">Toplantıyı Güncelle</button>

        </form>

    </div>

    <button type="button" onclick="editSelectedMeeting()">Mevcut Toplantıyı Güncelle</button>

    <script>

        let selectedMeetingId = null;

        let meetings = []; // meetings listesini tanımlama

        function editMeeting(meetingId) {

            // Güncelleme formunu görünür yapma

            document.getElementById('editForm').style.display = 'block';

            // Seçilen toplantının bilgilerini alma

            selectedMeetingId = meetingId;

            const selectedMeeting = meetings.find(meeting => meeting.id === meetingId);

            // Bilgileri form alanlarına doldurma

            if (selectedMeeting) {

                document.getElementById('editTopic').value = selectedMeeting.topic || '';

                document.getElementById('editDate').value = selectedMeeting.date || '';

                document.getElementById('editStartTime').value = selectedMeeting.start\_time || '';

                document.getElementById('editEndTime').value = selectedMeeting.end\_time || '';

                document.getElementById('editParticipants').value = selectedMeeting.participants || '';

            } else {

                // Eğer seçilen toplantı tanımlı değilse, hata mesajını konsola yaz

                console.error(`Toplantı ID'si ${meetingId} ile eşleşen bir toplantı bulunamadı.`);

                alert('Düzenlenecek bir toplantı bulunamadı.');

                document.getElementById('editForm').style.display = 'none';

            }

        }

        function updateMeeting() {

            // Güncelleme formunu gizle

            document.getElementById('editForm').style.display = 'none';

            // Seçilen toplantının bilgilerini alma

            const selectedMeetingIndex = meetings.findIndex(meeting => meeting.id === selectedMeetingId);

            const selectedMeeting = meetings[selectedMeetingIndex];

            // Güncelleme formundan yeni bilgileri alma

            selectedMeeting.topic = document.getElementById('editTopic').value;

            selectedMeeting.date = document.getElementById('editDate').value;

            selectedMeeting.start\_time = document.getElementById('editStartTime').value;

            selectedMeeting.end\_time = document.getElementById('editEndTime').value;

            selectedMeeting.participants = document.getElementById('editParticipants').value;

            // Güncellenmiş bilgileri ekrana yansıt

            renderMeetings();

        }

        function editSelectedMeeting() {

            // Mevcut toplantıyı düzenleme formunu görünür yapma

            const selectedMeeting = meetings.find(meeting => meeting.id === selectedMeetingId);

            if (selectedMeeting) {

                editMeeting(selectedMeetingId);

            } else {

                alert('Düzenlenecek bir toplantı seçilmemiş.');

            }

        }

        function renderMeetings() {

            const meetingsList = document.getElementById('meetingsList');

            meetingsList.innerHTML = ''; // Önceki toplantıları temizle

            for (const meeting of meetings) {

                const listItem = document.createElement('li');

                listItem.innerHTML = `<strong>${meeting.topic}</strong> - ${meeting.date}, ${meeting.start\_time}-${meeting.end\_time}, Katılımcılar: ${meeting.participants}`;

                const editButton = document.createElement('button');

                editButton.textContent = 'Düzenle';

                editButton.addEventListener('click', () => editMeeting(meeting['id']));

                const deleteButton = document.createElement('button');

                deleteButton.textContent = 'Sil';

                deleteButton.addEventListener('click', () => deleteMeeting(meeting['id']));

                listItem.appendChild(editButton);

                listItem.appendChild(deleteButton);

                meetingsList.appendChild(listItem);

            }

        }

        function addMeeting() {

            const form = document.getElementById('meetingForm');

            const formData = new FormData(form);

            // ID'yi oluştur

            const meetingId = createMeetingId();

            const meeting = {

                id: meetingId,

                topic: formData.get('topic'),

                date: formData.get('date'),

                start\_time: formData.get('start\_time'),

                end\_time: formData.get('end\_time'),

                participants: formData.get('participants')

            };

            // meetings listesine toplantıyı ekleme

            meetings.push(meeting);

            // Toplantıları güncelle ve ekranda göster

            renderMeetings();

        }

        function createMeetingId() {

            // Rasgele bir sayı oluşturarak benzersiz bir ID elde et

            return Math.floor(Math.random() \* 1000000).toString();

        }

        function deleteMeeting(id) {

            meetings = meetings.filter(meeting => meeting.id !== id);

            renderMeetings();

        }

    </script>

</body>

</html>