

Meeting Organizer Çalışması

Hazırlayan : Uğur Berkay Güneydaş

Tarih: 20.02.2024

Proje Çıktısı : Bir veya birden fazla toplantının kayıt edilebileceği, düzenlenebileceği ve silenebileceği Toplantı Uygulaması Projesinin son görünümüdür.

The screenshot shows the 'MechSoft Meeting Organizer' web application. At the top, there's a header with the title. Below it, a form for adding a meeting includes fields for 'Toplantı Konusu:', 'Tarih:' (with a date picker showing 'gg.aa.yyyy'), 'Başlangıç Saati:' (with a time picker showing '--:--'), 'Bitiş Saati:' (with a time picker showing '--:--'), and 'Katılımcılar (virgülle ayırın):'. A 'Toplantı Ekle' button is on the right. Below the form, there's a 'Toplantı Listesi' section. It contains a list item: '• Proje Planlama - 2024-02-19, 10:00-12:00, Katılımcılar: ['Ahmet', 'Mehmet']'. To the right of this item are 'Düzenle' and 'Sil' buttons. At the bottom left of the list, there's a 'Mevcut Toplantıyı Güncelle' button.

Figure 1 Proje Görünüm

Tarih Seçme:

This screenshot shows the date selection interface. At the top, there's a 'Tarih:' label followed by a date picker showing '19.02.2024' and a 'Başlangıç Saati:' label. Below this is a calendar for 'Şubat 2024'. The calendar has a grid with days of the week (Pt, Sa, Ça, Pe, Cu, Ct, Pa) and dates. The date '19' is highlighted in blue. At the bottom of the calendar, there are two buttons: 'Temizle' and 'Bugün'.

Figure 2 Kullanıcının Toplantı tarihini seçmek için kullanacağı bölüm

Saat Seçimi:

Başlangıç Saati: 17 : 30 Bitiş Saati: 20 : 30

| | |
|----|----|
| 17 | 30 |
| 18 | 31 |
| 19 | 32 |
| 20 | 33 |
| 21 | 34 |
| 22 | 35 |
| 23 | 36 |

Figure 3 Kullanıcının Toplantı saatini seçmek için kullanacağı bölüm

Genel Görünüm:

MechSoft Meeting Organizer

Toplantı Konusu: deneme3 Tarih: 21.02.2024 Başlangıç Saati: 15 : 30 Bitiş Saati: 19 : 35 Katılımcılar (virgülle ayırın): Ahmet, Mehmet, Hüseyin, İ Toplantı Ekle

Toplantı Listesi

- deneme1 - 2024-02-19, 17:30-20:30, Katılımcılar: Uğur,Berkay,Güneydaş Düzenle Sil
- deneme2 - 2024-02-20, 20:30-00:30, Katılımcılar: Uğur,Berkay Düzenle Sil
- deneme3 - 2024-02-21, 15:30-19:35, Katılımcılar: Ahmet, Mehmet, Hüseyin, Mert Düzenle Sil

Mevcut Toplantıyı Güncelle

Figure 4 Örnek olarak eklenmiş birkaç toplantı

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

- deneme3 - 2024-02-21, 15:30-19:35, Katılımcılar: Ahmet, Mehmet, Hüseyin, Mert Düzenle Sil

Toplantı Güncelleme Formu

Toplantı Konusu: deneme3_Güncelleme_den

Tarih: 21.02.2024

Başlangıç Saati: 15 : 30

Bitiş Saati: 19 : 35

Katılımcılar: Uğur,Berkay

Toplantıyı Güncelle

Mevcut Toplantıyı Güncelle

Toplantı Listesi

- deneme1 - 2024-02-19, 17:30-20:30, Katılımcılar: Uğur,Berkay,Güneydaş Düzenle Sil
- deneme2 - 2024-02-20, 20:30-00:30, Katılımcılar: Uğur,Berkay Düzenle Sil
- deneme3_Güncelleme_deneme - 2024-02-21, 15:30-19:35, Katılımcılar: Uğur,Berkay Düzenle Sil

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> Meeting Organizer </title>
</head>
<body>
  <body style="background-color:#7796c7"></body>
  <h1 style="text-align:center;"> 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>
```