NAMA: Hasim Hasan  
NIM: 23EO10043

**1. Create: Membuat Pemesanan Lapangan Olahraga**

**User Story:**

* **Sebagai** seorang pengguna (atlet atau tim),
* **Saya ingin** dapat membuat pemesanan untuk lapangan olahraga,
* **Sehingga** saya dapat memastikan bahwa lapangan tersedia pada waktu yang saya inginkan.

**Acceptance Criteria:**

* Pengguna dapat memilih jenis lapangan yang ingin dipesan.
* Pengguna dapat memilih tanggal dan waktu pemesanan.
* Pengguna dapat mengisi informasi kontak dan nama pemesan.
* Setelah pemesanan berhasil, pengguna menerima konfirmasi pemesanan.

**2. Read: Melihat Jadwal Pemakaian Lapangan**

**User Story:**

* **Sebagai** seorang pengguna,
* **Saya ingin** dapat melihat jadwal pemakaian lapangan olahraga,
* **Sehingga** saya dapat mengetahui kapan lapangan tersedia untuk pemesanan.

**Acceptance Criteria:**

* Pengguna dapat melihat daftar lapangan yang tersedia.
* Pengguna dapat melihat jadwal pemakaian untuk setiap lapangan berdasarkan tanggal.
* Pengguna dapat memfilter jadwal berdasarkan jenis lapangan atau tanggal tertentu.

**3. Update: Mengubah Waktu Pemesanan**

**User Story:**

* **Sebagai** seorang pengguna,
* **Saya ingin** dapat mengubah waktu pemesanan yang telah saya buat,
* **Sehingga** saya dapat menyesuaikan jadwal saya jika ada perubahan.

**Acceptance Criteria:**

* Pengguna dapat mencari pemesanan yang sudah ada.
* Pengguna dapat memilih pemesanan yang ingin diubah.
* Pengguna dapat memilih waktu baru untuk pemesanan.
* Setelah perubahan berhasil, pengguna menerima konfirmasi perubahan.

**4. Delete: Menghapus Pemesanan yang Dibatalkan**

**User Story:**

* **Sebagai** seorang pengguna,
* **Saya ingin** dapat menghapus pemesanan yang telah saya batalkan,
* **Sehingga** saya tidak lagi terikat pada pemesanan yang tidak akan saya gunakan.

**Acceptance Criteria:**

* Pengguna dapat mencari pemesanan yang ingin dihapus.
* Pengguna dapat memilih pemesanan yang ingin dihapus.
* Setelah pemesanan dihapus, pengguna menerima konfirmasi bahwa pemesanan telah berhasil dihapus.

Kode Program  
  
from flask import Flask, render\_template, request, redirect, url\_for

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

# Simulasi database menggunakan list

bookings = []

@app.route('/')

def index():

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

@app.route('/create', methods=['GET', 'POST'])

def create():

if request.method == 'POST':

name = request.form['name']

date = request.form['date']

time = request.form['time']

bookings.append({'name': name, 'date': date, 'time': time})

return redirect(url\_for('index'))

return render\_template('create.html')

@app.route('/update/<int:booking\_id>', methods=['GET', 'POST'])

def update(booking\_id):

if request.method == 'POST':

bookings[booking\_id]['date'] = request.form['date']

bookings[booking\_id]['time'] = request.form['time']

return redirect(url\_for('index'))

return render\_template('update.html', booking=bookings[booking\_id], booking\_id=booking\_id)

@app.route('/delete/<int:booking\_id>')

def delete(booking\_id):

bookings.pop(booking\_id)

return redirect(url\_for('index'))

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

app.run(debug=True)