Project Title:

Employee Assessment System

Technologies Used:

- **Python** (Flask framework)
- **CSV** for data storage
- **HTML** (Embedded using `render_template_string`)

Objective:

The Employee Assessment System is a simple Flask-based web application that enables users (like HR personnel or managers) to:

- View employee records.
- Assess and update each employee's performance score and attendance.
- Persist changes using a CSV file instead of a database.

Key Components:

1. Data Storage:

- Stored in a file named `employees.csv`
- Columns: `id`, `name`, `department`, `performance_score`, `attendance`
- Initial data is preloaded if the file doesn't exist

2. Functions:

- `load_employees()`: Loads employee data from CSV into a list of dictionaries.
- `save_employees()`: Saves updated employee data back to CSV.

📦 Routes and Functionality:

`/` → *Home Page*

- Displays all employees in a table
- Provides a link to assess each employee

`/assess/<employee_id>` → *Assessment Page*

- Displays a form to update `performance_score` and `attendance` for the selected employee

`/update/<employee_id>` → *Update Handler*

- Processes the submitted form and updates the CSV with new values

---

👦💼 Sample Employee Data:

ID	Name	Department	Performance Score	Attendance
1	John Doe	Engineering	85	95
2	Jane Smith	Marketing	90	90
3	Alice Johnson	Sales	78	85
4	Bob Brown	HR	88	92
5	Charlie Davis	Engineering	82	88

📌 Features Summary:

- Lightweight web-based interface.
- Employee information displayed in a clean HTML table.
- Inline editing via a dedicated assessment page.
- Real-time updates saved to CSV file.
- Fully functional CRUD-like behavior (limited to read/update).

🚀 Usage Instructions:

1. Run the script using Python:
2. `python employee_assessment.py`
3. Open the browser and navigate to `http://localhost:5000/`
4. Click **Assess** to view and edit individual records.

Source code

Employee Assessment System

```
from flask import Flask, render_template_string, request, redirect, url_for
import csv
import os

app = Flask(name)
```

Dataset (embedded in the script)

```
EMPLOYEES_CSV = "employees.csv"
if not os.path.exists(EMPLOYEES_CSV):
with open(EMPLOYEES_CSV, mode='w', newline='') as file:
writer = csv.writer(file)
writer.writerow(['id', 'name', 'department', 'performance_score', 'attendance'])
writer.writerows([
[1, 'John Doe', 'Engineering', 85, 95],
[2, 'Jane Smith', 'Marketing', 90, 90],
[3, 'Alice Johnson', 'Sales', 78, 85],
[4, 'Bob Brown', 'HR', 88, 92],
[5, 'Charlie Davis', 'Engineering', 82, 88]
])
```

Load employee data from CSV

```
def load_employees():
employees = []
with open(EMPLOYEES_CSV, mode='r') as file:
reader = csv.DictReader(file)
for row in reader:
employees.append(row)
return employees
```

Save employee data to CSV

def save_employees(employees):
with open(EMPLOYEES_CSV, mode='w', newline='') as file:
fieldnames = ['id', 'name', 'department', 'performance_score', 'attendance']
writer = csv.DictWriter(file, fieldnames=fieldnames)
writer.writeheader()
for employee in employees:
writer.writerow(employee)

HTML Templates (embedded in the script)

INDEX_HTML = """

Employee List

ID	Name	Department	Performance Score	Attendance	Action {% for employee in employees %}
{{ employee.id }}	{{ employee.name }}	{{ employee.department }}	{{ employee.performance_score }}	{{ employee.attendance }}	Assess {% endfor %}

"""

ASSESS_HTML = """

Assess Employee: {{ employee.name }}

Performance Score:
Attendance:
Update

Back to List

"""

Home page - Display all employees

@app.route('/')
def index():
employees = load_employees()
return render_template_string(INDEX_HTML, employees=employees)

Assess an employee

@app.route('/assess/int:employee_id')
def assess(employee_id):
employees = load_employees()
employee = next((emp for emp in employees if int(emp['id']) == employee_id), None)
if employee:
return render_template_string(ASSESS_HTML, employee=employee)
return redirect(url_for('index'))

Update an employee's assessment

@app.route('/update/int:employee_id', methods=['POST'])
def update(employee_id):
employees = load_employees()
employee = next((emp for emp in employees if int(emp['id']) == employee_id), None)
if employee:
employee['performance_score'] = request.form['performance_score']
employee['attendance'] = request.form['attendance']
save_employees(employees)
return redirect(url_for('index'))

Run the application

if **name** == '**main**':
app.run(debug=True)

Data set

id,name,department,performance_score,attendance
1,John Doe,Engineering,85,95
2,Jane Smith,Marketing,90,90

3,Alice Johnson,Sales,78,85
4,Bob Brown,HR,88,92
5,Charlie Davis,Engineering,82,88