# FLASK WITH SQLITE3 DATABASE

Flask can make use of the SQLite3 module of the python to create the database web applications.

**Flask SQLAlchemy**

Flask SQLAlchemy is an ORM tool which establishes the relationship between the objects and the tables of the relational databases.

The mapping between the both is important because the python is capable of storing the data in the form of objects whereas the database stores the data in the form of relational tables, i.e. the collection of rows and columns.

The object-relational mapping is the technique of storing python objects into the database tables without writing the raw SQL queries.

Creating a small web application using flask-sqlalchemy

pip install flask-sqlalchemy

## # app/\_\_init\_\_.py

from flask import Flask

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

app.config.from\_object("config.DevelopmentConfig")

from app import views, database

## # app/views.py

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

from app import app

from .database import Employees, db

@app.route('/')

def list\_employees():

    return render\_template('read.html', Employees = Employees.query.all())

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

def addEmployee():

    if request.method == 'POST':

        if not request.form['name'] or not request.form['salary'] or not request.form['age']:

            flash('Please enter all the fields', 'error')

        else:

            employee = Employees(request.form['name'], request.form['salary'], request.form['age'], request.form['pin'])

            db.session.add(employee)

            db.session.commit()

            flash('Record was successfully added')

            return redirect(url\_for('list\_employees'))

    return render\_template('create.html')

@app.route('/edit/<int:id>', methods=['GET', 'POST'])

def editEmployee(id):

    employee = Employees.query.get\_or\_404(id)

    if request.method == 'POST':

        if not request.form['name'] or not request.form['salary'] or not request.form['age']:

            flash('Please enter all the fields', 'error')

        else:

            employee.name = request.form['name']

            employee.salary = request.form['salary']

            employee.age = request.form['age']

            employee.pin = request.form['pin']

            db.session.commit()

            flash('Record was successfully updated')

            return redirect(url\_for('list\_employees'))

    return render\_template('update.html', employee=employee)

@app.route('/delete/<int:id>', methods=['POST'])

def deleteEmployee(id):

    employee = Employees.query.get\_or\_404(id)

    db.session.delete(employee)

    db.session.commit()

    flash('Record was successfully deleted')

    return redirect(url\_for('list\_employees'))

## # app/database.py

from flask\_sqlalchemy import SQLAlchemy

from app import app

db = SQLAlchemy(app)

class Employees(db.Model):

    id = db.Column('employee\_id', db.Integer, primary\_key=True)

    name = db.Column(db.String(100))

    salary = db.Column(db.Float)

    age = db.Column(db.String(200))

    pin = db.Column(db.String(100))

    def \_\_init\_\_(self, name, salary, age, pin):

        self.name = name

        self.salary = salary

        self.age = age

        self.pin = pin

# Create the database and tables

with app.app\_context():

    db.create\_all()

## <!-- app/templates/create.html -->

<!DOCTYPE html>

<html>

<body>

    <h3>Add new Employee</h3>

    <hr/>

    {% for message in get\_flashed\_messages() %}

        <div class="alert alert-danger">

            {{ message }}

        </div>

    {% endfor %}

    <form action="{{ url\_for('addEmployee') }}" method="post">

        <label for="name">Name</label><br>

        <input type="text" name="name" placeholder="Name" /><br>

        <label for="salary">Salary</label><br>

        <input type="text" name="salary" placeholder="Salary" /><br>

        <label for="age">Age</label><br>

        <textarea name="age" placeholder="Age"></textarea><br>

        <label for="PIN">Pin</label><br>

        <input type="text" name="pin" placeholder="Pin" /><br>

        <input type="submit" value="Submit" />

    </form>

</body>

</html>

## <!-- templates/read.html -->

<!DOCTYPE html>

<html lang="en">

<head><title>Home</title></head>

<body>

    <h3>

        <a href="{{ url\_for('list\_employees') }}">Employee Management System</a>

    </h3>

    <hr/>

    {% for message in get\_flashed\_messages() %}

        {{ message }}

    {% endfor %}

    <h3>Employees List</h3>

    <table border="2" padding="5">

        <thead>

            <tr>

                <th>Name</th>

                <th>Salary</th>

                <th>Age</th>

                <th>Pin</th>

                <th>Action</th>

            </tr>

        </thead>

        <tbody>

            {% for employee in Employees %}

                <tr>

                    <td>{{ employee.name }}</td>

                    <td>{{ employee.salary }}</td>

                    <td>{{ employee.age }}</td>

                    <td>{{ employee.pin }}</td>

                    <td>

                        <form action="{{ url\_for('deleteEmployee', id=employee.id) }}" method="post" style="display:inline;">

                            <button type="submit">Delete</button>

                        </form>

                        <a href="{{ url\_for('editEmployee', id=employee.id) }}">Edit</a>

                    </td>

                </tr>

            {% endfor %}

        </tbody>

    </table>

    <br><br>

    <a href="{{ url\_for('addEmployee') }}">Add New Employee</a>

</body>

</html>

## <!-- templates/update.html -->

<!DOCTYPE html>

<html>

<body>

    <h3>Edit Employee</h3>

    <hr/>

    {%- for message in get\_flashed\_messages() %}

        <div class="alert alert-danger">

            {{ message }}

        </div>

    {%- endfor %}

    <form action="{{ url\_for('editEmployee', id=employee.id) }}" method="post">

        <label for="name">Name</label><br>

        <input type="text" name="name" value="{{ employee.name }}" /><br>

        <label for="salary">Salary</label><br>

        <input type="text" name="salary" value="{{ employee.salary }}" /><br>

        <label for="age">Age</label><br>

        <textarea name="age">{{ employee.age }}</textarea><br>

        <label for="PIN">Pin</label><br>

        <input type="text" name="pin" value="{{ employee.pin }}" /><br>

        <input type="submit" value="Update" />

    </form>

</body>

</html>

## # config.py

class Config:

    SECRET\_KEY = "ka76mnivninr093039lxndko392"

    SQLALCHEMY\_TRACK\_MODIFICATIONS = False

class DevelopmentConfig(Config):

    SQLALCHEMY\_DATABASE\_URI = 'sqlite:///employees.sqlite3'

    DEBUG = True

## # run.py

from app import app

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)