

**Кейс**. **Реализация веб интерфейса формы обратной связи**

**Цель:** Данный кейс направлен на оценку уровня владения Вами навыками разработки веб приложений, знакомство с базами данных.

**Задачи, которые необходимо решить:**

1. Спроектировать и реализовать структуру БД.
2. Создание backend приложения.
3. Разработка frontend приложения.

**Вводная информация:**

Задание разделено на 2 части. Базовую и дополнительную. Базовая часть обязательна для выполнения. Выполнение дополнительной рассматривается в качестве плюса.

**Описание программы:**

Создать базу данных в выбранной СУБД (sqlite, mysql, postgreesql или другая **SQL-**совместимая СУБД). Создать необходимые для выполнения задания таблицы и справочные данные (сделать это в виде sql скрипта).

Написать python веб приложение заполнения формы обратной связи с сохранением результата в базу данных. Приложение должно реализовывать возможность просмотра и удаления добавленных записей.

*Добавление комментариев*. После запуска приложения при обращении по относительному пути /comment/ должна отображаться форма для заполнения. Форма состоит из следующих полей:

* фамилия
* имя
* отчество
* регион
* город
* контактный телефон
* e-mail
* комментарий.

Поля фамилия, имя и комментарий являются обязательными. Поле комментарий текстовое. Для полей телефон и email следует производить проверку ввода. Номер телефона в формате «(код города) номер». Поля с некорректным вводом и не заполненные обязательные поля должны визуально выделяться красным цветом. Поля регион и город являются выпадающими списками, при этом список выбора поля город зависит от выбранного поля регион. Данные для этих списков должны храниться в СУБД. Значение в поля город должно динамически подгружаться по технологии ajax в соответствии с выбранным полем регион. Таблица соответствия для примера:

|  |  |
| --- | --- |
| **Регион** | **Город** |
| Краснодарский край | Краснодар |
|  | Кропоткин |
|  | Славянск |
| Ростовская область | Ростов |
|  | Шахты |
|  | Батайск |
| Ставропольский край | Ставрополь |
|  | Пятигорск |
|  | Кисловодск |

***Дополнительная часть.***

*Просмотр комментариев*. При обращении по относительному пути /view/ должна выводиться таблица со списком добавленных комментариев. В этом же представлении должна быть возможность удалить определенную запись.

*Удаление комментариев.* В представлении просмотра комментариев реализовать возможность удаления отдельно выбранного комментария.

*Просмотр статистики.* При обращении по относительному пути /stat/ должна выводиться таблица со списком тех регионов у которых количество комментариев больше 5, выводить так же и количество комментариев по каждому региону. Каждая строчка должны быть ссылкой на список города этого региона в котором отображается количество комментариев по этому городу.

**Удачи!**

**Ответ:**

**Файл *app.py***

from \_\_future\_\_ import print\_function  
from flask import Flask, render\_template, request, jsonify  
from flask\_mysqldb import MySQL  
import re  
  
app = Flask(\_\_name\_\_)  
  
  
app.config['MYSQL\_HOST'] = 'localhost'  
app.config['MYSQL\_USER'] = 'root'  
app.config['MYSQL\_PASSWORD'] = 'pass'  
app.config['MYSQL\_DB'] = 'MyBD'  
  
mysql = MySQL(app)  
  
@app.route('/index')  
@app.route('/')  
def index():  
 return render\_template('index.html')  
  
@app.route('/сomment', methods=['GET', 'POST'])  
def comment():  
 # словарь для сообщений об ошибках  
 messages = {}  
 # словарь для данных из формы  
 details = {}  
 if request.method == "POST":  
 details = request.form  
 last\_name = details['last\_name']  
 first\_name = details['first\_name']  
 ot4estvo = details['ot4estvo']  
 region = details['region']  
 city = details['city']  
 email = details['email']  
 phone = details['phone']  
 message = details['message']  
 # если пустое поле имя  
 if not last\_name:  
 messages.update({"last\_name": "Введите фамилию!!!"})  
 # если пустое поле фамилия  
 if not first\_name:  
 messages.update({"first\_name": "Введите имя!!!"})  
 # если пустое поле комментарий  
 if not message:  
 messages.update({"message": "Введите комментарий!!"})  
 # сравниваем введенный email с шаблоном  
 mo = re.search(r'^.+@(\[?)[a-zA-Z0-9-.]+.([a-zA-Z]{2,3}|[0-9]{1,3})(]?)$', email)  
 # если email не совпадает с шаблоном и НЕ пустой  
 if not mo and email:  
 messages.update({"email": "Введите Email в правильной форме example@mail.com!!!"})  
 # сравниваем введенный телефон с шаблоном  
 mo = re.search(r'\(\d{3}\)\d{7}', phone)  
 # если телефон не совпадает с шаблоном и НЕ пустой  
 if not mo and phone:  
 messages.update({"phone": "Введите телефон в правильной формате: (918)1234567!!!"})  
 # если есть ошибки(т.е. словарь message НЕ пустой), перенапрявляем на исходную страницу с сообщениями об ошибках  
 if len(messages) != 0:  
 messages.update({"ok": "Ошибка ввода"})  
 return render\_template('comment.html', data=messages, post=details)  
 # если ввод верный то остовляем сообщение об удачной отправке  
 else:  
 # обнуляем массив данных  
 details = []  
 messages.update({"ok": "Спасибо за Ваш комментарий"})  
 # отправляем данные в базу  
 cur = mysql.connection.cursor()  
 cur.execute('INSERT INTO anketa(last\_name, first\_name, ot4estvo, region, city, email, phone\_number, message) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)', (last\_name, first\_name, ot4estvo, region, city, email, phone, message))  
 mysql.connection.commit()  
 cur.close()  
 return render\_template('comment.html', data=messages, post=details)

# просмотр комментариев  
@app.route('/view', methods=['GET', 'POST'])  
def view():  
 # если метод post, то требуется удаление записи  
 if request.method == "POST":  
 details = request.form  
 # id записи, которую надо удалить  
 id = (details['id'], )  
 cur = mysql.connection.cursor()  
 # удаляем запись из базы  
 cur.execute("Delete from anketa where anketa\_id=%s", (id, ))  
 mysql.connection.commit()  
 cur.close()  
 # выводим комментарии(без удаленого, если ранее был запрос на удаление строки)  
 cur = mysql.connection.cursor()  
 cur.execute("SELECT \* FROM anketa")  
 data = cur.fetchall()  
 mysql.connection.commit()  
 cur.close()  
 return render\_template('view.html', data=data)  
  
# просмотр статистики  
@app.route('/stat', methods=['GET', 'POST'])  
def stat():  
 # если GET(просмотр статистики по регионам)  
 if request.method == "GET":  
 cur = mysql.connection.cursor()  
 # список городов и кол-во комментариев из конкретного региона  
 query = "SELECT region, *count*(region) FROM anketa GROUP BY region;"  
 cur.execute(query)  
 data = cur.fetchall()  
 mysql.connection.commit()  
 cur.close()  
 return render\_template('stat.html', data=data)  
 # если POST(просмотр статистики по городам выбраного региона)  
 elif request.method == "POST":  
 details = request.form  
 id = details['id']  
 cur = mysql.connection.cursor()  
 # список городов и кол-во комментариев из конкретного региона  
 query = "SELECT city, *count*(city) FROM anketa where region = %s GROUP BY city;"  
 cur.execute(query, (id,))  
 data = cur.fetchall()  
 mysql.connection.commit()  
 cur.close()  
 # возвращаем список городов на отдельную страницу для просмотра городов  
 return render\_template('stat\_city.html', data=data, region=id)  
  
@app.route('/\_get\_region/', methods=['POST'])  
def \_get\_region():  
 cur = mysql.connection.cursor()  
 # список регионов  
 # если надо увеличить список регионов, достаточно добавить их в таблицу, код менять не надо  
 query ="Select region from regions"  
 cur.execute(query)  
 data = cur.fetchall()  
 mysql.connection.commit()  
 cur.close()  
 # возвращаем список регионов  
 return jsonify({'data': data})  
  
@app.route('/\_get\_city/', methods=['POST'])  
def \_get\_city():  
 details = request.form  
 id = details['id']  
 cur = mysql.connection.cursor()  
 # список городов из данного региона  
 # если надо увеличить список городов, достаточно добавить их в таблицу, код менять не надо  
 query ="Select city from cities where region\_id = %s"  
 cur.execute(query, (id,))  
 data = cur.fetchall()  
 mysql.connection.commit()  
 cur.close()  
 # возвращаем список городов  
 return jsonify({'data': data})  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run()

**Файл templates/index.html**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
<a href = "{{url\_for('comment')}}">comment</a>  
<br>  
<a href = "{{url\_for('view')}}">view</a>  
<br>  
<a href = "{{url\_for('stat')}}">stat</a>  
</body>  
</html>

**Файл templates/comment.html**

<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta http-equiv="content-type" content="text/html; charset=utf-8">  
 <title>comment</title>  
 <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
 <!--скрипт для получения регионов из таблицы-->  
 <script type="text/javascript" src="{{ url\_for('static', filename='sripts/get\_region.js') }}"></script>  
 <!--скрипт для получения из таблицы городов из конкретного регона -->  
 <script type="text/javascript" src="{{ url\_for('static', filename='sripts/get\_city.js') }}"></script>  
 <script>  
 //скрипт для скрытия(показа) города если выбран пустой регион(не пустой регион)  
 $(***document***).ready(function() {  
 $('#region').change(function() {  
 if ($("#region :selected").val()!="") {  
 //если регион не пустой  
 //показываем "город"  
 $("#city").css("display", "inline-block");  
 $("#label\_city").css("display", "inline-block");  
 } else {  
 //если регион пустой  
 //скрываем "город"  
 $("#city").css("display", "none");  
 $("#label\_city").css("display", "none");  
 //очищаем "город"  
 $("#city").empty();  
 }  
 });  
 });  
 </script>  
</head>  
<BODY onload="get\_region()">  
 <a href = "{{url\_for('index')}}">Главное</a>  
 <br>  
 <a href = "{{url\_for('view')}}">view</a>  
 <br>  
 <a hre f = "{{url\_for('stat')}}">stat</a>  
 <br>  
 <div id="w">  
 <form method="POST" action="" id="contactform" name="MyForm">  
 <p class="note"><span class="req">\*</span> Поля со звездочкой обязательны для заполнения</p>  
 <!-- сообщение об удачной/неудачной отправке-->  
 <h3>{{ data['ok'] }}</h3>  
  
 <!--поле фамилия  
 крассный цвет если ошибка-->  
 {% if not data['last\_name'] %}  
 <div class="req">  
 {% else %}  
 <div class="err">  
 {% endif %}  
 <label for="last\_name">Фамилия<span class="req">\*</span></label>  
 <input type = "text" name = "last\_name" class="txt" tabindex="1" placeholder="Иванов", value="{{ post['last\_name'] }}"/><br>  
 <!--если не было введено-->  
 <h3>{{ data['last\_name'] }}</h3>  
 </div>  
  
 <!--поле имя  
 крассный цвет если ошибка-->  
 {% if not data['first\_name'] %}  
 <div class="req">  
 {% else %}  
 <div class="err">  
 {% endif %}  
 <label for="first\_name">Имя<span class="req">\*</span></label>  
 <input type = "text" name = "first\_name" class="txt" tabindex="2"placeholder="Иван" value="{{ post['first\_name'] }}" /><br>  
 <!--если не было введено-->  
 <h3>{{ data['first\_name'] }}</h3>  
 </div>  
  
 <!--поле отчество-->  
 <div class="row">  
 <label for="ot4estvo">Отчество</label>  
 <input type = "text" name = "ot4estvo" class="txt" tabindex="3" placeholder="Иванович" value="{{ post['ot4estvo'] }}"/><br>  
 </div>  
  
 <!--поле регион-->  
 <div class="row">  
 <label for='region'>Регион</label>  
 <SELECT name="region" id="region" onChange="get\_city()">  
 <option value=""></option>  
 </SELECT>  
 </div>  
  
 <!--поле город-->  
 <div class="row">  
 <label id="label\_city" for='city'>Город</label>  
 <SELECT name="city" id="city">  
 <option value=""></option>  
 </SELECT>  
  
 <!--поле email  
 крассный цвет если ошибка-->  
 {% if not data['email'] %}  
 <div class="req">  
 {% else %}  
 <div class="err">  
 {% endif %}  
 <label for="email">E-mail</label>  
 <input type="text" id="email" name="email" placeholder="example@mail.com" class="txt" tabindex="6" value="{{ post['email'] }}"><br>  
 <!--если было введено неверно-->  
 <h3>{{ data['email'] }}</h3>  
 </div>  
  
 <!--поле телефон  
 крассный цвет если ошибка-->  
 {% if not data['phone'] %}  
 <div class="req">  
 {% else %}  
 <div class="err">  
 {% endif %}  
 <label for="phone">Телефон</label>  
 <input type="text" id="phone" name="phone" placeholder="(900)1234567" class="txt" tabindex="7" value="{{ post['phone'] }}"><br>  
 <!--если было введено неверно-->  
 <h3>{{ data['phone'] }}</h3>  
 </div>  
  
 <!--поле комментария  
 крассный цвет если ошибка-->  
 {% if not data['message'] %}  
 <div class="req">  
 {% else %}  
 <div class="err">  
 {% endif %}  
 <label for="message">Комментарий<span class="req">\*</span></label>  
 <textarea type="text" rows="10" cols="45" name="message" placeholder="Комментарий" class="txt" tabindex="8" value="{{ post['message'] }}"></textarea><br>  
 <!--если не было введено-->  
 <h3>{{ data['message'] }}</h3>  
 </div>  
  
 <!--кнопка отправки данных-->  
 <div class="center">  
 <input type = "submit" id="validate" name="submitbtn" tabindex="8" value="Отправить">  
 </div>  
 </form>  
 </div>  
</BODY>  
</HTML>

**Файл templates/view.html**

<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  
 <title>view</title>  
</head>  
<body>  
<a href = "{{url\_for('index')}}">Главное</a>  
<br>  
<a href = "{{url\_for('comment')}}">comment</a>  
<br>  
<a href = "{{url\_for('stat')}}">stat</a>  
<br>

<table class="table\_blur">  
 <!--верхняя строка-->  
 <tr>  
 <th>id</th>  
 <th>Фамилия</th>  
 <th>Имя</th>  
 <th>Отчество</th>  
 <th>Регион</th>  
 <th>Город</th>  
 <th>Еmail</th>  
 <th>Телефон</th>  
 <th>Комментарий</th>  
 <th>Удалить</th>  
 </tr>  
 </tr>  
 <!--вывод коментариев-->  
 {% for item in data %}  
 <tr>  
 <td>{{item[0]}}</td>  
 <td>{{item[1]}}</td>  
 <td>{{item[2]}}</td>  
 <td>{{item[3]}}</td>  
 <td>{{item[4]}}</td>  
 <td>{{item[5]}}</td>  
 <td>{{item[6]}}</td>  
 <td>{{item[7]}}</td>  
 <td>{{item[8]}}</td>  
 <!--кнопка для удаления комментария-->  
 <td>  
 <form method="POST" action="">  
 <p><input type='hidden' name ='id' value={{item[0]}}></p>  
 <p><input type="submit" value="удалить запись"></p>  
 </form>  
 </td>  
 </tr>  
 {% endfor %}  
</table>  
</body>  
</html>

**Файл templates/stat.html**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  
 <title>stat</title>  
 <!--Статистика по регионам-->  
</head>  
<body>  
 <a href = "{{url\_for('index')}}">Главное</a>  
 <br>  
 <a href = "{{url\_for('comment')}}">comment</a>  
 <br>  
 <a href = "{{url\_for('view')}}">view</a>  
 <br>  
 <table class="table\_blur">  
 <!--вверхняя строка-->  
 <tr>  
 <th>Регион</th>  
 <th>Кол-во комментарев</th>  
 <th>Подробнее</th>  
 </tr>  
 <tr>  
 <!--Вывод строк по регионам-->  
 {% for item in data %}  
 <!--Выводим если у региона больше 5 комментаниев-->  
 {% if item[1] > 5%}  
 <tr>  
 {% if item[0] == '' %}  
 <td>Комменариев без региона</td>  
 <td>{{item[1]}}</td>  
 <td></td>  
 {% else %}  
 <td>{{item[0]}}</td>  
 <td>{{item[1]}}</td>  
 <!--кнопка для показа статистики по региону-->  
 <td>  
 <form method="POST" action="">  
 <p><input type='hidden' name ='id' value="{{item[0]}}"></p>  
 <p><input type="submit" value="Подробнее"></p>  
 </form>  
 </td>  
 {% endif %}  
 </tr>  
 {% endif %}  
 {% endfor %}  
 </table>  
</body>  
</html>

**Файл templates/stat\_city.html**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  
 <title>stat\_city</title>  
 <!--Статистика по городам конкретного региона-->  
</head>  
<body>  
 <a href = "{{url\_for('index')}}">Главное</a>  
 <br>  
 <a href = "{{url\_for('comment')}}">comment</a>  
 <br>  
 <a href = "{{url\_for('view')}}">view</a>  
 <br>  
 <a href = "{{url\_for('stat')}}">stat</a>  
 <br>  
 <table class="table\_blur">  
 <!--вверхняя строка-->  
 <tr>  
 <th>{{region}}</th>  
 <th>Кол-во комментарев</th>  
 </tr>  
 <!--Вывод строк по городам-->  
 {% for item in data %}  
 <tr>  
 <td>{{item[0]}}</td>  
 <td>{{item[1]}}</td>  
 </tr>  
 {% endfor %}  
 </table>  
</body>  
</html>

**Файл static\sripts\get\_region.js**

function get\_region(){  
 // ссылка на select region  
 var regionObj = ***document***.getElementById("region");  
 // массив для вывода  
 $.ajax({  
 url: "/\_get\_region/",  
 type: "POST",  
 success: function(resp) {  
 // массив регионов  
 var arr = resp.data  
 for (var n = 0; n < arr.length; n++){  
 // заполняем строку для select  
 // option ('название объекта', 'значение')  
 regionObj[n+1] = new ***Option***(arr[n], arr[n]);  
 }  
 }  
 });  
}

**Файл static\sripts\get\_city.js**

function get\_city(){  
 // ссылка на select region  
 var regionObj = ***document***.getElementById("region");  
 // ссылка на select city  
 var cityObj = ***document***.getElementById("city");  
 // индекс выбраного региона  
 var index = regionObj.options.selectedIndex;  
 $.ajax({  
 url: "/\_get\_city/",  
 type: "POST",  
 data: {'id': index},  
 success: function (resp) {  
 // массив городов из выбраного региона  
 var arr = resp.data  
 // заполняем строку для select  
 // option ('название объекта', 'значение')  
 for (var n = 0; n < arr.length; n++) {  
 cityObj[n] = new ***Option***(arr[n], arr[n]);  
 }  
 }  
 });  
}

**Файл static\css\styles.css**

/\* form styles \*/  
#city, #label\_city {  
 display: none;  
}  
form{  
 background-color: rgba(246, 172, 186, 0.72)  
}  
form .row {  
display: block;  
padding: 7px 8px;  
margin-bottom: 7px;  
}  
  
form.err {  
 display: block;  
 padding: 7px 8px;  
 margin-bottom: 7px;  
 background-color: red  
}  
  
form label {  
display: inline-block;  
font-size: 1.2em;  
font-weight: bold;  
width: 120px;  
padding: 6px 0;  
color: #000000;  
vertical-align: top;  
}  
  
.err { background-color: red }  
  
h3 {  
 color: black;  
}  
  
form .req { color: #ca5354; }  
  
form .note {  
font-size: 1.2em;  
line-height: 1.33em;  
font-weight: normal;  
padding: 2px 7px;  
margin-bottom: 10px;  
}  
  
form input:focus, form textarea:focus { outline: none; }  
  
form .txt {  
display: inline-block;  
padding: 8px 9px;  
padding-right: 30px;  
width: 240px;  
font-family: 'Oxygen', sans-serif;  
font-size: 1.35em;  
font-weight: normal;  
color: #000000;  
  
}  
form .center{  
 padding: 8px 9px;  
 padding-left: 130px;  
}  
form #submitbtn {  
 background: rgb(162, 0, 35);  
 color: white;  
 border-style: outset;  
 border-color: rgb(162, 0, 35);  
 width: 240px;  
 font: bold 15px arial, sans-serif;  
 text-shadow:none;  
}  
  
.table\_blur {  
 background: #f5ffff;  
 border-collapse: collapse;  
 text-align: left;  
}  
.table\_blur th {  
 border-top: 1px solid #777777;  
 border-bottom: 1px solid #777777;  
 box-shadow: inset 0 1px 0 #999999, inset 0 -1px 0 #999999;  
 background: linear-gradient(#c599a6, #7f5666);  
 color: white;  
 padding: 10px 15px;  
 position: relative;  
}  
.table\_blur th:after {  
 content: "";  
 display: block;  
 position: absolute;  
 left: 0;  
 top: 25%;  
 height: 25%;  
 width: 100%;  
 background: linear-gradient(rgba(255, 255, 255, 0), rgba(255,255,255,.08));  
}  
.table\_blur tr:nth-child(odd) {  
 background: #ebf3f9;  
}  
.table\_blur th:first-child {  
 border-left: 1px solid #777777;  
 border-bottom: 1px solid #777777;  
 box-shadow: inset 1px 1px 0 #999999, inset 0 -1px 0 #999999;  
}  
.table\_blur th:last-child {  
 border-right: 1px solid #777777;  
 border-bottom: 1px solid #777777;  
 box-shadow: inset -1px 1px 0 #999999, inset 0 -1px 0 #999999;  
}  
.table\_blur td {  
 border: 1px solid #e3eef7;  
 padding: 10px 15px;  
 position: relative;  
 transition: all 0.5s ease;  
}  
.table\_blur tbody:hover td {  
 color: transparent;  
 text-shadow: 0 0 3px #a09f9d;  
}  
.table\_blur tbody:hover tr:hover td {  
 color: #444444;  
 text-shadow: none;  
}  
  
#city {  
 display: none;  
}

**Sql-скрипт для создание и заполнения базы**

create table anketa(

    anketa\_id int(10) not null auto\_increment,

    last\_name varchar(128) not null,

    first\_name varchar(128) not null,

    ot4estvo varchar(128) null,

    region varchar(128) not null,

    city varchar(128) not null,

    phone\_number varchar(12) null,

    email varchar(128) null,

    message varchar(500) not null,

    PRIMARY KEY (anketa\_id)

);

create table regions(

    region\_id int(10) not null auto\_increment,

    region varchar(128) not null,

    PRIMARY KEY (region\_id)

);

create table cities(

    city\_id int(10) not null auto\_increment,

    region\_id int(10) not null,

    city varchar(128) not null,

    PRIMARY KEY (city\_id),

    FOREIGN KEY (region\_id) REFERENCES regions (region\_id)

);

insert into regions(region) values ('Краснодарский край');

insert into regions(region) values ('Ростовская область');

insert into regions(region) values ('Ставропольский край');

insert into cities(region\_id, city) values (1, 'Краснодар');

insert into cities(region\_id, city) values (1, 'Кропоткин');

insert into cities(region\_id, city) values (1, 'Славянск');

insert into cities(region\_id, city) values (2, 'Ростов');

insert into cities(region\_id, city) values (2, 'Шахты');

insert into cities(region\_id, city) values (2, 'Батайск');

insert into cities(region\_id, city) values (3, 'Ставрополь');

insert into cities(region\_id, city) values (3, 'Пятигорск');

insert into cities(region\_id, city) values (3, 'Кисловодск');

**ФИО: Хижний Евгений Георгиевич**