

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»**

Отчет

по лабораторной работе №4

«РАЗРАБОТКА ИНТЕРФЕЙСОВ ДЛЯ ВЫПОЛНЕНИЯ CRUD-ОПЕРАЦИЙ НАД
БАЗОЙ ДАННЫХ СРЕДСТВАМИ PHP»
по дисциплине «Проектирование и реализация баз данных»

Авторы: Антонова М. А.

Факультет: ФИКТ

Группа: K32422

Преподаватель: Говорова М.М.

Санкт-Петербург, 2023

1 Описание работы

Цель работы: овладеть практическими навыками разработки форм для вставки, выборки и редактирования данных.

Оборудование: компьютерный класс.

Программное обеспечение: среда разработки XAMPP 8 (PHP 8) или аналогичная, СУБД PostgreSQL 1X, pgadmin 4.

Практическое задание:

1. Изучить функции открытия соединения к базе данных средствами PHP.
2. Изучить основные функции для создания php-скрипта (на базе видео-уроков 1-7).
3. Создать сайт с использованием базовых возможностей PHP (в соответствии с содержанием видео-уроков 1-7).

2 Описание выполнения работы

База данных

Была создана база данных студентов по курсам английского языка.

создание файла с данными

```
import sqlite3
```

```
conn = sqlite3.connect('students.db') # Warning: This file is created in the current directory
```

```
conn.execute("CREATE TABLE students (id INTEGER PRIMARY KEY, student char(100) NOT NULL, status bool NOT NULL)")
```

```
conn.execute("INSERT INTO students (student,status) VALUES ('Антонов Николай',0)")
```

```
conn.execute("INSERT INTO students (student,status) VALUES ('Петров Иван',1)")
```

```
conn.execute("INSERT INTO students (student,status) VALUES ('Иванов
Дмитрий',1)")
conn.execute("INSERT INTO students (student,status) VALUES ('Денисов
Артем',0)")
conn.commit()
```

Таблица со списком студентов

Была создана таблица со списком, которые учатся на данный момент.

```
#создание таблицы со списком
```

```
@route('/students')
```

```
@route('/student_list')
```

```
def student_list():
```

```
    conn = sqlite3.connect('students.db')
```

```
    c = conn.cursor()
```

```
    c.execute("SELECT id, student FROM students WHERE status LIKE '1'")
```

```
#вывод строк, где статус равен единице
```

```
    result = c.fetchall()
```







```
    c.close()
```

```
    output = template('make_table', rows=result)
```

```
    return output
```

Студенты, которые сейчас проходят курс:



№	Студенты	Редактировать/Удалить
2	Петров Иван	 
3	Иванов Дмитрий	 
7	Андреев Никита	 



Добавление новых студентов

```
@route('/new', method='GET')
```

```
def new_item():
```

```
    if request.GET.save:
```

```
        new = request.GET.student.strip()
```

```
        conn = sqlite3.connect('students.db')
```

```
        c = conn.cursor()
```

```
        c.execute("INSERT INTO students (student,status) VALUES (?,?)", (new,1))
```

```
        new_id = c.lastrowid
```

```
        conn.commit()
```

```
        c.close()
```

```
    return '<p>The new student was inserted into the database, the ID is %s</p>'
% new_id

else:

    return template('new_student.tpl')
```

Создаем форму для вставки

```
<p>Add a new student to the list:</p>
<form action="/new" method="GET">
    <input type="text" size="100" maxlength="100" name="student">
    <input type="submit" name="save" value="save">
</form>
```

Add a new student to the list:

The new student was inserted into the database, the ID is 13

Удаление студента

Удаляем студента, если он не заплатил за вторую половину курса.

```
def delete_student(no):
    if request.GET.save:
        delete = request.GET.delete.strip()
        print(delete)
        if delete == 'yes':
            conn = sqlite3.connect('students.db')
            c = conn.cursor()
```

```

        c.execute("DELETE FROM students WHERE id=(?)", (no,))

        conn.commit()

        return '<p>The student %s was successfully deleted</p>' % no

    else:

        return 'Nothing was deleted'

    else:

        return template('delete_student', no=no)

```

Создана форма для удаления студента

```

<head>

    <link      rel="stylesheet"      href="//cdnjs.cloudflare.com/ajax/libs/semantic-
ui/2.2.12/semantic.min.css"></link>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>

    <script      src="https://cdnjs.cloudflare.com/ajax/libs/semantic-
ui/2.2.12/semantic.min.js"></script>

</head>

<p>Are you sure that you want to delete student with ID = { {no} }?</p>

<form action="/delete/{ {no} }" method="get">

    <select name="delete">

        <option>yes</option>

        <option>no</option>

    </select>

    <br>

    <input type="submit" name="save" value="save">

</form>

<th colspan="7">

    <a href="/student_list" class="ui small button">Return to list</a>

</th>

```

При удалении необходимо ответить на вопрос

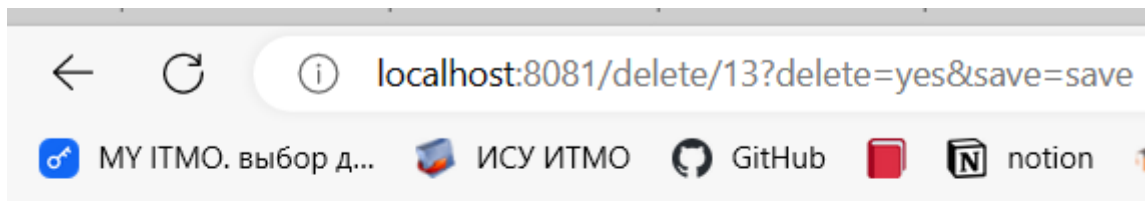
Are you sure that you want to delete student with ID = 13?

yes ▾

save

Return to list

А затем удалить



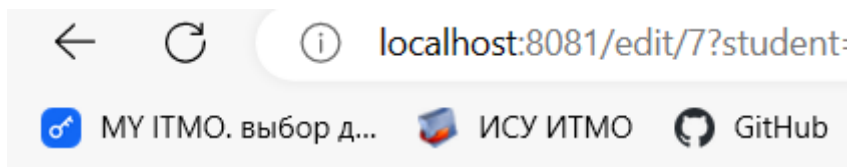
The student 13 was successfully deleted

Редактирование

Также можно отредактировать имя и статус студента. Например, если студент сменил фамилию или закончил курсы английского языка.

Edit the student with ID = 7

<input type="text" value="Николаев Никита"/>	<input type="text" value="currently studying"/>
<input type="button" value="save"/>	



The student number 7 was successfully updated

Была создана форма для редактирования

%#template for editing a student

%#the template expects to receive a value for "no" as well a "old", the text of the selected student

```
<p>Edit the student with ID = {{no}}</p>
<form action="/edit/{{no}}" method="get">
  <input type="text" name="student" value="{{old[0]}}" size="100"
maxlength="100">
  <select name="status">
    <option>currently studying</option>
    <option>finished studying</option>
  </select>
  <br>
  <input type="submit" name="save" value="save">
</form>
```

ВЫВОД

При выполнении данной лабораторной работы были приобретены практические навыки по работе с базой данных. Были реализованы операции добавления, редактирования и удаления данных.

Таким образом, был получен опыт по работе и использованию форм для вставки, выборки и редактирования данных