

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень,  
звание

подпись, дата

Соловьева Н.А.

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5  
**Серверные сценарии. Работа с базой данных**  
по дисциплине: Web-Технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

4134к

подпись, дата

Столяров Н.С.

инициалы, фамилия

Санкт-Петербург  
2024

**Цель работы:** получение опыта написания и применения серверных сценариев для обработки данных, введенных пользователем.

### **Базовое задание**

На основе приведенных примеров подготовить веб-страницу, на которой пользователь вводит несколько значений и отправляет их на сервер. После отправки на сервер введенные данные печатаются на странице. Вариант определяет вид запроса, отправляемого на сервер (get или post) и внешний вид отображаемых данных. Кроме того, в варианте указан дополнительный элемент формы, который необходимо самостоятельно освоить и включить в форму.

Вопросы анкеты должны отражать тему сайта, выбранную в первой лабораторной работе. Запрещается сохранять поля образца кроме имени.

Обязательные поля формы для анкеты: 1) фамилия 2) имя 3) отчество 4) список с выбором значений, 5) набор флажков (checkbox), 6) набор переключателей (radio).

Серверный сценарий написан на языке python.

При использовании метода get в отчет следует включить содержимое адресной строки после получения ответа от сервера.

Страница должна быть встроена в сайт, подготовленный в предыдущих работах.

### **Расширенное задание**

- 1.. применить оформление через css к форме
- 2.. в форму добавить сценарий на javascript, проверяющий введенные значения
- 3.. данные анкеты записывать в текстовый файл в серверном сценарии с накоплением результата. Названия полей не хранить.
- 4.. применить оформление через css к странице с ответом сервера
- 5.. ответ сервера оформить как список определений

Вариант

19	Get	Построчно	password
----	-----	-----------	----------

Ответ от сервера

Результат формы

surname: 1

name: 1

middle\_name: 1

phone: 1

time\_education: Заочная

sex: male

code: 1234

human: on

agreement: on

Вид страницы

Запись на курс

Фамилия

Имя

Отчество

Номер телефона

Тип обучения: Очная

Пол:

Мужской

Женский

Реферальный код

☐ Вы человек?

☐ Я ничего не принимаю. Просто хочу нажать на кнопку отправки

Отправить

## Листинг html

```
<!DOCTYPE html>
<html lang="ru" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Волшебная форма</title>
    <link rel="stylesheet" href="main.css">
  </head>
  <body>
    <div class="form">
      <p class="form_name">Запись на курс</p>
      <form id="form" class="" action="/" method="post">
        <input type="text" name="surname" placeholder="Фамилия">
        <input type="text" name="name" placeholder="Имя">
        <input type="text" name="middle_name" placeholder="Отчество">
        <input type="text" name="phone" placeholder="Номер телефона">
        <label for="test">Тип обучения: </label>
        <select class="" name="time_education">
          <option value="Очная">Очная</option>
          <option value="Заочная">Заочная</option>
        </select>
        <br>
```

```

<label>Пол:</label>
<br>
<label for="">
  <input type="radio" name="sex" value="male">
    Мужской
</label>
<br>
<label for="">
  <input type="radio" name="sex" value="female">
    Женский
</label>
<br>
<br>
<input type="password" name="code" placeholder="Реферальный код">
<label for="">
  <input type="checkbox" name="human">
    Вы человек?
</label>
<br>
<label for="">
  <input type="checkbox" name="agreement">
    Я ничего не принимаю. Просто хочу нажать на кнопку отправки
</label>
<br>

  <input type="submit" value="Отправить">
</form>
</div>
<script src="main.js" charset="utf-8"></script>
</body>
</html>

```

## Листинг python

```

from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.parse import parse_qs

class handler(BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path.endswith('.html'):
            self.send_response(200)
            self.send_header('Content-type', 'text/html; charset=utf-8')
            self.end_headers()
            with open(self.path[1:], 'rb') as file:
                self.wfile.write(file.read())
        elif self.path.endswith('.css'):
            self.send_response(200)
            self.send_header('Content-type', 'text/css; charset=utf-8')
            self.end_headers()
            with open(self.path[1:], 'rb') as file:
                self.wfile.write(file.read())

```

```

elif self.path.endswith('.js'):
    self.send_response(200)
    self.send_header('Content-type', 'application/javascript; charset=utf-8')
    self.end_headers()
    with open(self.path[1:], 'rb') as file:
        self.wfile.write(file.read())
else:
    self.send_response(404)
    self.send_header('Content-type', 'text/html; charset=utf-8')
    self.end_headers()
    self.wfile.write(bytes("404 Not Found", "utf8"))

def do_POST(self):
    self.send_response(200)
    self.send_header('Content-type', 'text/html; charset=utf-8')
    self.end_headers()

    content_length = int(self.headers['Content-Length'])
    post_data_bytes = self.rfile.read(content_length)

    post_data_str = post_data_bytes.decode("UTF-8")
    post_data_dict = parse_qs(post_data_str)

    list = ""
    for key, values in post_data_dict.items():
        list += f"<li>{key}: {' '.join(values)}</li>"

    response_message = f"""
<!DOCTYPE html>
<html lang="ru" dir="ltr">
<head>
    <meta charset="utf-8">
    <title>Ответ формы</title>
    <link rel="stylesheet" href="main.css">
</head>
<body>
    <h1 class="name">Результат формы</h1>
    <ul class="result">
        {list}
    </ul>
</body>
</html>
"""

    self.wfile.write(bytes(response_message, "utf8"))

# Сохраняем данные в файл
with open("data.txt", "a", encoding="utf-8") as file:
    for value in post_data_dict.values():
        file.write(value[0] + '\n')
    file.write("-"*10)

```

```
with HTTPServer(('0.0.0.0', 80), handler) as server:  
    print("server started")  
    server.serve_forever()
```

## **Выводы**

Я освоил работу с формами на html и обработку информации на python