ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Соловьева Н.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

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

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Столяров Н.С. |
|  |  |  | подпись, дата |  | 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.

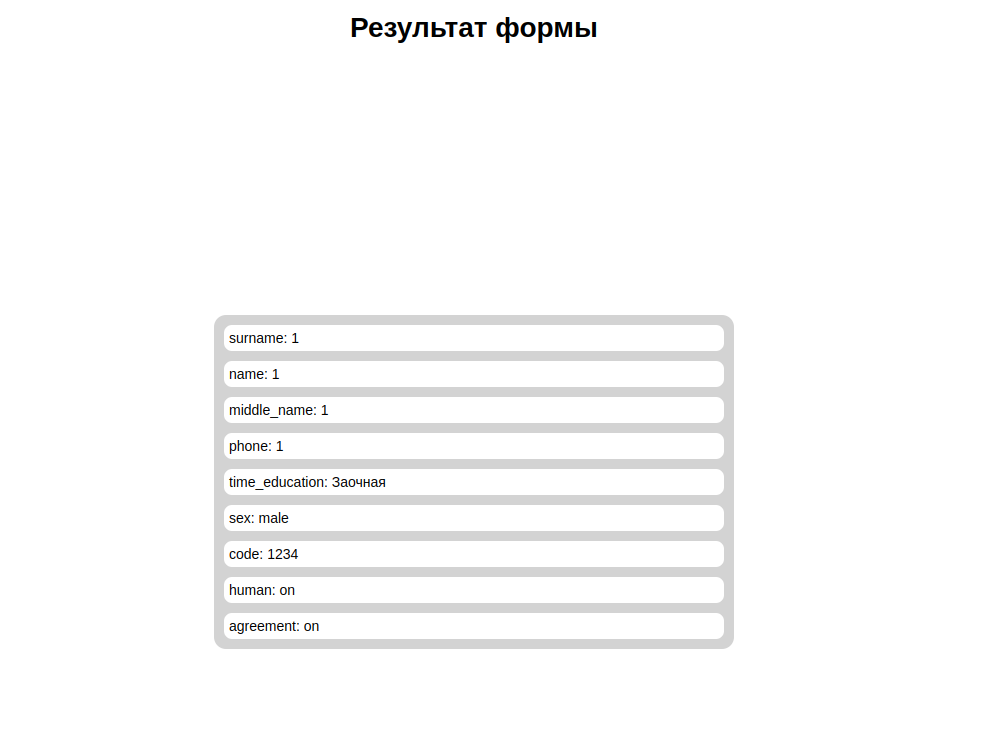
4.. применить оформление через css к странице с ответом сервера

5.. ответ сервера оформить как список определений

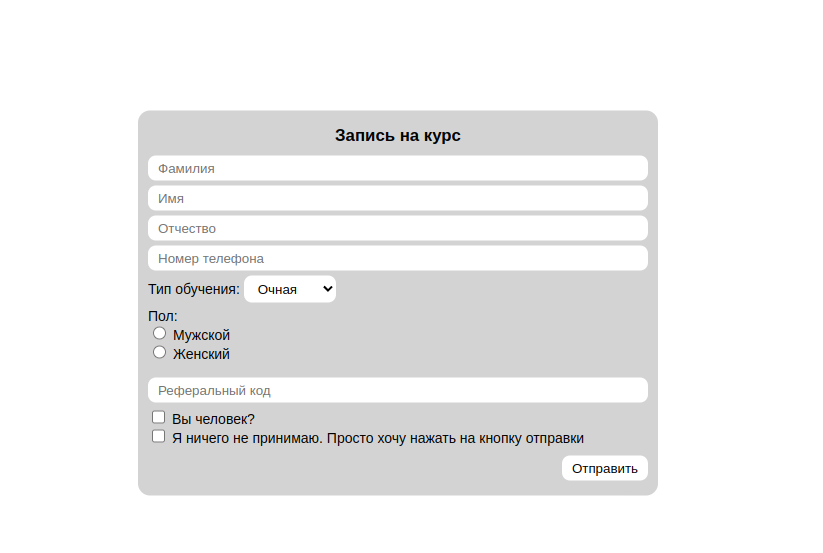
Вариант



# Ответ от сервера



# Вид страницы



# 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