

Методические указания для выполнения
лабораторной работы № 5 по курсу «Web-технологии»

Серверные сценарии. Форма и обработка get и post запросов.

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

Правила оценивания работ

Выполнение базового задания оценивается максимально в 6 баллов. Для получения за работу 10 баллов требуется выполнить **все** пункты из раздела «Расширенное задание».

Из максимально возможной оценки баллы будут вычитаться при следующих ситуациях:

- 1) **Срок.** Отчет выложен в ЛК после срока сдачи – минус 1 балл каждые 2 недели от даты, записанной в ЛК. Дата защиты на получение оценки не влияет;
- 2) **Защита.** «Пробелы» в знаниях во время устной защиты работы: оценка «хорошо» - минус 1 балл, оценка «удовлетворительно» - минус 2 балла;
- 3) **Отчет.** В отчете присутствуют незначительные нарушения требований - минус 1 балл, отсутствует несколько пунктов отчета - минус 2 балла или больше;
- 4) **Работа.** В выполненном задании отсутствуют второстепенные элементы - минус 1 балл, отсутствует несколько пунктов задания - минус 2 балла или больше.

Защита работы возможна только при наличии в ЛК отчета.

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 отдельные поля формы для анкеты: 1) фамилия 2) имя 3) отчество 4) список с выбором значений, 5) набор флажков (checkbox), 6) набор переключателей (radio), 7) электронная почта.

Применить оформление через css к форме и к странице с ответом сервера.

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.. В ответ сервера добавить строчку, сформированную по следующему шаблону: «А.И. Иванова является студент, носит средний размер обуви и любит язык питон.» (см. пример ниже, рис. 2 и 3). То есть на странице, подготовленной сервером (рис. 3), кроме таблицы должна еще быть написана фраза, в которой присутствуют инициалы, фамилия и другие значения, полученные из анкеты. Падеж и род в словах подбирать не надо.

- 2.. в форму добавить сценарий на javascript, проверяющий введенные значения.
Минимальные проверки: факт заполнения полей; текст проверяется на отсутствие любых символов, кроме букв; электронная почта – на наличие символа @. Можно добавить свои проверки.
- 3.. данные анкеты записывать в текстовый файл в серверном сценарии с накоплением результата. Названия полей не хранить.

Порядок выполнения

1. Установить оболочку языка python
2. Из листинга 1 сформировать файл python (расширение .py) и записать его в основной каталог оболочки python. Из оболочки python запустить сервер. Сервер работает постоянно.
3. Подготовить веб-страницу с формой (расширение .htm) по образцу Листинга 2. Веб-страница должна находиться в том же каталоге, что и файл с сервером. Вызывается в адресной строке браузера следующим образом: <http://localhost/forma.htm> Редактировать страницу можно в блокноте, а для проверки работы обновлять окно браузера.
4. Подготовить файл с серверным сценарием (расширение .py) по образцу Листингов 3, 4 или 5 в зависимости от варианта. Файл должен быть записан в каталог cgi-bin, кодировка utf-8. Вызывается сервером после нажатия кнопки на форме. Редактировать этот файл тоже можно в блокноте.
5. После завершения работы остановить сервер.

Содержание отчета

- 1) титульный лист;
- 2) цель работы;
- 3) вариант задания: тема сайта и вариант по работе № 5
- 4) при использовании метода get: адресная строка после получения ответа от сервера.
- 5) скриншоты: исходная страница и страница, сформированная веб-сервером.
- 6) листинги двух файлов (веб-страница с формой и серверный сценарий)
- 7) выводы по работе.

Работа с языком питон

Простейшую оболочку редактора языке питон можно загрузить с сайта <https://www.python.org/downloads/>, выбирайте версию 3.x . При работе в этой оболочке используется два окна: окно интерпретатора (рис. 1) и редактор текста (рис. 2). В окне интерпретатора в строке с приглашением (>>>) можно писать операторы питона. При выполнении программы в этом окне происходит вывод на экран.

Окно редактора текста запускается командой меню File/New file (рис. 2). В это окно следует скопировать текст веб-сервера из листинга 1. В питоне функцию открывающих и закрывающих скобок в составных операторах выполняют отступы от начала строки. Поэтому все операторы кода из листинга 1 обязательно должны находиться строго с начала строки. Сохраните файл с расширением .py и запустите его из редактора текста (кнопка F5). В окне интерпретатора (рис. 3) будут появляться сообщения о работе сервера. Сервер работает постоянно.

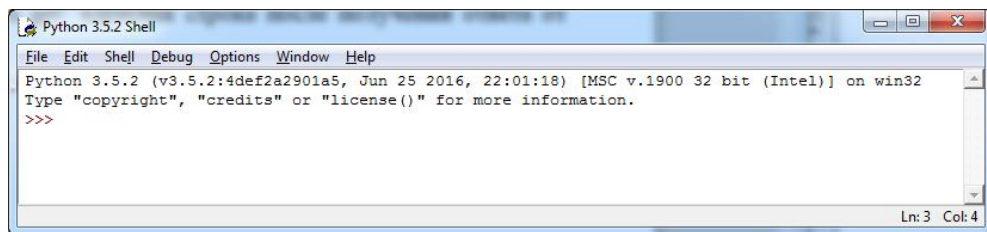


Рис. 1 окно интерпретатора для языка питон.

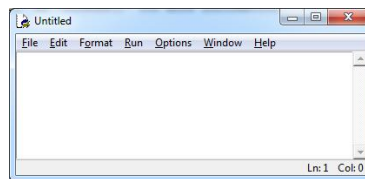


Рис. 2. Окно редактора текста

Листинг 1. Веб-сервер на языке python. Файл serwerpy.py

```
import os, sys
from http.server import HTTPServer, CGIHTTPRequestHandler
webdir = '.' # каталог с файлами HTML и подкаталогом cgi-bin для сценариев
port = 80 # http://servername/ если 80, иначе http://servername:xxxx/
if len(sys.argv) > 1: webdir = sys.argv[1] # аргументы командной строки
if len(sys.argv) > 2: port = int(sys.argv[2]) # иначе по умолчанию ., 80
print('webdir "%s", port %s' % (webdir, port))
os.chdir(webdir) # перейти в корневой веб-каталог
srvraddr = ('', port) # имя хоста, номер порта
srvrobj = HTTPServer(svraddr, CGIHTTPRequestHandler)
srvrobj.serve_forever() # обслуживать клиентов до завершения
```

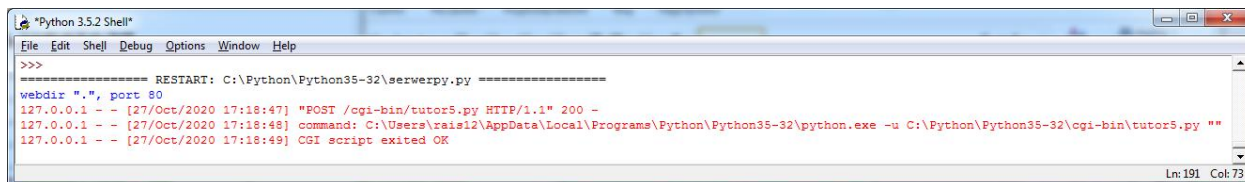


Рис. 3. Сообщения веб-сервера в окне интерпретатора.

Комментарии к листингу 2

Форма

Формы – это элемент веб-страницы. Она служат для оправки данных на веб-сервер. Форма заключается в парный тег `<form>`. Внутри этого тега помещаются элементы интерфейса пользователя – кнопки, поля ввода и т.п. Элементы интерфейса пользователя могут находиться и вне формы – в этом случае для того, чтобы интерфейс был полезен, к элементам прикрепляется сценарий на языке javascript.

Пример первого тега формы:

```
<FORM method="POST" action="/cgi-bin/tutor5.py">
```

Основные атрибуты формы: action, name, method.

action – содержит адрес (url) файла, который получает данные из формы. В примере это файл на языке питон tutor5.py, расположенный в каталоге cgi-bin.

name – имя формы.

method – содержит название метода (get или post) передачи данных на веб-сервер. Данные передаются в виде пары: имя_элемента_name = значение_элемента_value через разделитель &.

В описании элемента «переключатель» :

```
<input type=radio name=shoesize value=маленький>
```

имя элемента задано атрибутом name, а значение атрибутом value.

Метод get передает данные из формы внутри url, этот вариант означает, что передаваемые данные открыты и доступны всем, поскольку отображаются в адресной строке браузера. Пример передачи данных из формы при выполнении запроса get:

<http://localhost/cgi-bin/horiztable.py?name=Аделина&shoesize=маленький&job=студент&language=Python&comment=люблю+программировать>

Чтобы получить читаемый текст при использовании кириллицы, надо зайти на один из сайтов, которые выполняют кодировку/расшифровку url с кириллицей (например, <https://www.design-sites.ru/utility/url-encoding.php>) и преобразовать текст.

Метод post передает данные в виде отдельного файла. Рекомендуется использовать именно его.

Форма будет передавать данные после нажатия кнопки типа submit. Атрибут value здесь содержит надпись на кнопке. Пример кнопки:

```
<input type=submit value="ОТПРАВИТЬ">
```

Создать элемент интерфейса можно разными способами: с помощью непарного тега `<input>`, в котором через атрибут `type` задается тип элемента, и с помощью специальных тегов. Все элементы интерфейса имеют атрибуты `name` и `value`.

Атрибут `name` – имя элемента интерфейса, необходимо для обращения к элементу при отправке данных на веб-сервер. Атрибут `value` – значение элемента. Для элемента `text` – значение по умолчанию, для кнопки – надпись на ней.

Создание интерфейса через тег `<input>`

Основные атрибуты: `type`, `src`.

Атрибут **`type`** – задает вид элемента интерфейса. Варианты значений:

`button` - кнопка,
`checkbox` - флажок,
`file` – текстовое поле для ввода имени файла, есть кнопка для открытия диалогового окна выбора файла.
`hidden` – невидимый элемент, но его данные можно передать на сервер,
`image` – графическая кнопка типа «submit»; атрибут `src` содержит имя графического файла,
`password` – текстовое поле для ввода пароля, аналогично элементу `text`; все символы при печати скрыты.
`radio` - переключатель,
`reset` – кнопка для установки значений по умолчанию,
`submit` – кнопка, которая отправляет данные формы на сервер,
`text` – текстовое поле ввода.

Примеры элементов управления из Листинга 3:

```
<input type=checkbox name=language value=Python>  
<input type=radio name=shoesize value=маленький>  
<input type=text name=name>  
<input type=submit value="ОТПРАВИТЬ">
```

Другие теги для элементов интерфейса

В парном теге кнопки `<button>` могут размещаться другие теги (текст, изображение, таблица). Для указания типа кнопки используется атрибут `type` со значениями: `button` – обычная кнопка, `reset` – отмена всех изменений в элементах формы, `submit` – опрака данных формы, внутри которой находится кнопка, на веб-сервер.

Раскрывающийся список создается через парный тег `<select>`, внутри которого размещаются теги `<option>`, содержащие пункты списка.

```
<select name=job>  
  <option>студент <option>преподаватель <option>библиотекарь  
  <option>другое </select>
```

Область ввода текста `<textarea>` - поле для ввода многострочного текста, может содержать полосы прокрутки. Атрибуты `cols` и `rows` определяют размер поля в количестве символов по горизонтали и вертикали.

```
<textarea name=comment cols=30 rows=2>
```

Для аккуратного размещения элементов интерфейса используется невидимая таблица.

Листинг 2. Файл `forma.htm` с формой.

Запуск файла: <http://localhost/forma.htm> при работающем веб-сервере

```
<HTML><TITLE>==Python==</TITLE>
<BODY>
<H1>Анкета пользователя</H1>
<HR>
<FORM method="POST" action="/cgi-bin/tutor5.py">
  <H3>Заполните форму и нажмите кнопку</H3>
  <P><TABLE>
    <TR>
      <TH align=right>Имя:
      <TD> <input type=text name=name>
    <TR>
      <TH align=right>Размер обуви:
      <TD><table>
        <td><input type=radio name=shoesize value=маленький>Маленький
        <td><input type=radio name=shoesize value=средний>Средний
        <td><input type=radio name=shoesize value=большой>Большой
      </table>
    <TR>
      <TH align=right>Должность:
      <TD><select name=job>
        <option>студент
        <option>преподаватель
        <option>библиотекарь
        <option>другое
      </select>
    <TR>
      <TH align=right>Любимый язык:
      <TD><table>
        <td><input type=checkbox name=language value=Python>Python
        <td><input type=checkbox name=language value=C>C
        <td><input type=checkbox name=language value=php>php
      </table>
    <TR>
      <TH align=right>Комментарий:
      <TD><textarea name=comment cols=30 rows=2>
        введите текст</textarea>
    <TR>
      <TD colspan=2 align=center>
        <input type=submit value="ОТПРАВИТЬ">
      </TD>
    </TABLE>
  </FORM>
<HR>
</BODY></HTML>
```

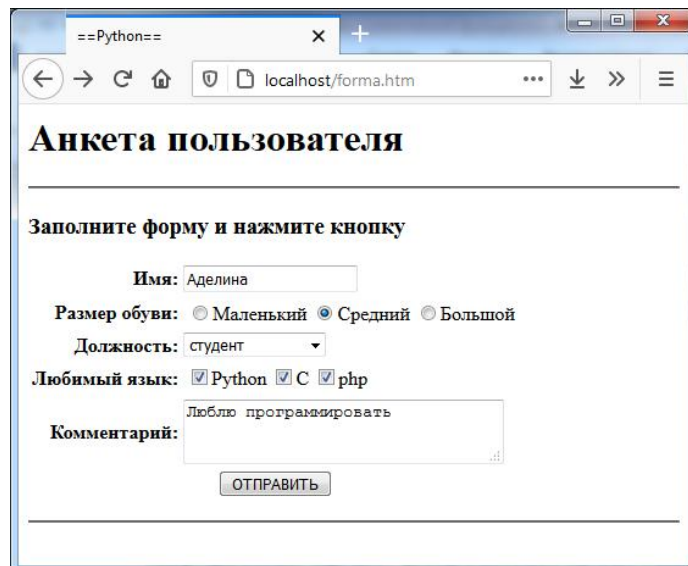


Рис. 2 Отображение веб-страницы из листинга 2.

Комментарии к листингу 3

Язык программирования Python

Python во многом похож на язык C, но имеет важные отличия:

- 1) объявление переменных не требуется, тип определяется по значению, которое присваивается переменной
- 2) знак «;» завершает оператор, но его роль может выполнять и конец строки
- 3) отсутствуют символы операторных скобок, их заменяет отступ от начала строки; выравнивания начала каждой строки имеет принципиальное значение.

Комментарий на одну строку обозначается символом #

`import cgi, sys` Подключение библиотек `cgi` и `sys`

`form = cgi.FieldStorage()` Переменной `form` присваиваются данные, которые переданы из формы. Для этого используется функция библиотеки. Переменная `form` является словарем, т.е. каждый элемент хранимого множества значений состоит из двух частей – ключ и значение. Ключом служит название элемента интерфейса (значение атрибута `name`).

Данный код формирует веб страницу. Это происходит постепенно, с помощью функции питона для вывода на экран `print()`. Выводимые на экран данные организованы в таблицу. Переменная `html1` является строкой и содержит начало веб-страницы. Чтобы записать длинный текст, располагающийся на нескольких строках, используются тройные кавычки. После первого применения функции `print` (строка `print("Content-type: text/html")`) здесь необходима обязательная пустая строка.

Для вывода названий по-русски, формируем список названий `ll`:

`ll = ['имя','размер обуви','должность', 'Любимый язык программирования','комментарий']`

Список питона во многом похож на массив в C, но имеются отличия: список меняет размер во время выполнения программы, элементы списка могут иметь разные типы данных, возможна нумерация с конца списка (последний элемент имеет индекс -1), список

является стандартным классом и обладает набором методов. Признак списка – квадратные скобки [].

Второй список, data, будет хранить значения, извлеченные из переменной-словаря form. Предварительно мы формируем список из нескольких пустых строк. Переменная i служит для индексации наших списков.

Строка for field in ('name', 'shoesize', 'job', 'language', 'comment'):
начинает цикл по 5 названиям элементов интерфейса формы. Выражение for x in A означает, что переменная x последовательно будет принимать все значения из последовательности A. Этот цикл завершается перед вторым словом for. Внутри цикла выполняется проверка заполненности полей формы.

Строка if not field in form:
проверяет отсутствие текущего значения переменной field среди ключей словаря form. Если название элемента интерфейса присутствует, то проверяется, не является ли значение этого элемента интерфейса списком (не состоит ли оно из нескольких элементов):
if not isinstance(form[field], list):
Несколько элементов может содержать поле language.

Если значение одно, то оно добавляется в список data:
data[i] = form[field].value

При наличии нескольких значений формируется список values. При этом применяется конструкция, которая называется «генератор списка» ([x.value for x in form[field]]). Генератор списка является сокращенной формой записи цикла по списку. Далее элементы списка объединяются в одну строку через запятые. Здесь используется метод строк .join(). Полученная строка присваивается списку data.

```
values = [x.value for x in form[field]]  
data[i] = ','.join(values)
```

После формирования списка data, выводим его в веб-страницу через цикл:
for i in range(5):
 print ('<tr><td> %s </td> <td> %s </td></tr>'% (ll[i], data[i]))

Функция range(5) генерирует индексы по спискам, от 0 до 4. Знаки %s в строке вывода обозначают позиции, в которые будет выведено значение, указанное после разделителя % (% (ll[i], data[i])). Буква s говорит, что значение будет выведено как строка.

Листинг 3. Вывод в вертикальную таблицу. Файл tutor5.py

#вывод в таблицу

```
import cgi, sys  
form = cgi.FieldStorage()      # извлечь данные из формы
```



```

print("Content-type: text/html") # плюс пустая строка

html1 = """
<TITLE>таблица с анкетой</TITLE>
<H1>Анкета пользователя</H1>
<table border =2> <tr> <td>Имя поля</td><td>Значение</td> </tr>
"""

# печать заголовка таблицы
print (html1)

ll = ['имя','размер обуви','должность', 'Любимый язык программирования','комментарий']

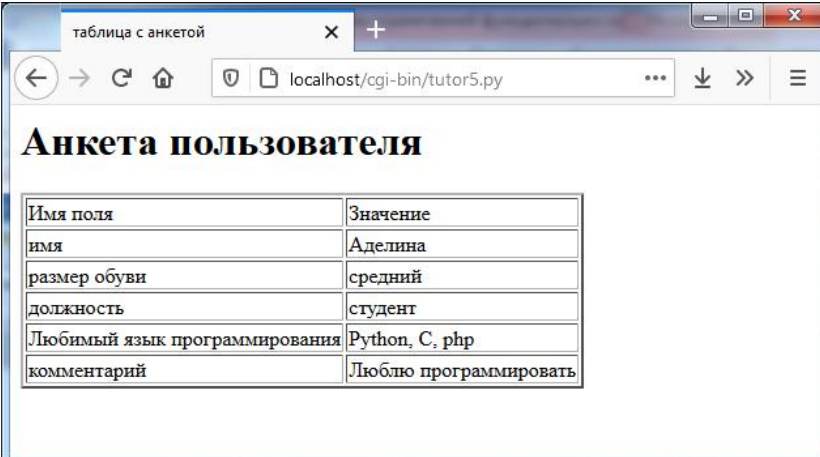
data = [",",",",","]; i=0
for field in ('name', 'shoesize', 'job', 'language', 'comment'):
    if not field in form:
        data[i] = '(unknown)'
    else:

        if not isinstance(form[field], list):
            data[i] = form[field].value
        else:
            values = [x.value for x in form[field]]
            data[i] = ','.join(values)
        i+=1

for i in range(5):
    print ('<tr><td> %s </td> <td> %s </td></tr>'%(ll[i], data[i]))

print (' </table>')

```



Имя поля	Значение
имя	Аделина
размер обуви	средний
должность	студент
Любимый язык программирования	Python, C, php
комментарий	Люблю программировать

Рис. 3 Ответ сервера. Вывод введенных данных анкеты в виде вертикальной таблицы.

Комментарии к Листингу 4

В отличие от Листинга 3, здесь переменная `data` является словарем, в котором ключ – название элемента интерфейса, а значение – данные, введенные пользователем. Общая логика алгоритма такая же.

В строке `html` подготовлен шаблон веб-страницы, в который в последнем операторе на места, отмеченные знаком `%`, будут добавлены данные пользователя.

При работе со словарем к элементу обращаются по значению ключа:

```
data[field] = '(unknown)'
```

Листинг 4. Вывод построчно. Файл `tutor5.py`

```
import cgi, sys
form = cgi.FieldStorage()      # извлечь данные из формы
print("Content-type: text/html") # плюс пустая строка

html = """
<TITLE>tutor5.py</TITLE>
<H1>Анкета пользователя</H1>  <HR>
<H4>Вас зовут: %(name)s</H4>
<H4>Размер обуви: %(shoesize)s</H4>
<H4>Ваша работа: %(job)s</H4>
<H4>Любимый язык: %(language)s</H4>
<H4>Комментарии:</H4>
<P>%(comment)s</P>
<HR>"""
data = {}
for field in ('name', 'shoesize', 'job', 'language', 'comment'):
    if not field in form:
        data[field] = '(unknown)'
    else:
        if not isinstance(form[field], list):
            data[field] = form[field].value
        else:
            values = [x.value for x in form[field]]
            data[field] = ' and '.join(values)
print(html % data)
```

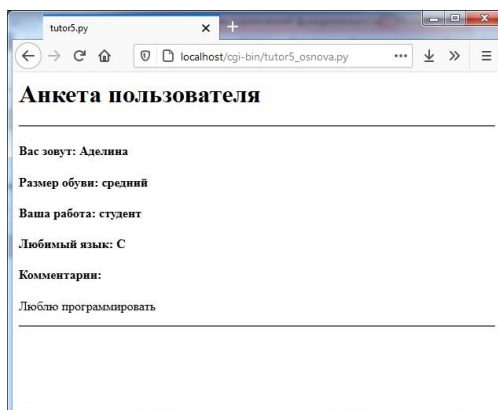


Рис. 4 Ответ сервера. Вывод введенных данных анкеты построчно

Комментарии к Листингу 5

Элементы программы, вошедшие в код в Листинге 5, уже объяснены в предыдущих комментариях.

Листинг 5. Вывод в горизонтальную таблицу. Файл tutor5.py

```
#вывод в таблицу
import cgi, sys
form = cgi.FieldStorage()      # извлечь данные из формы
print("Content-type: text/html") # плюс пустая строка

html1 = """
<TITLE>таблица с анкетой</TITLE>
<H1>Анкета пользователя</H1>
<table border =2> <tr>
"""
print (html1)
# печать заголовка таблицы
ll = ['имя','размер обуви','работа', 'язык','комментарий']
for head in ll:
    ss = '<td>'+head+'</td>'
    print ( ss)
print ('</tr> <tr>')

data = [",",",",","]; i=0
for field in ('name', 'shoesize', 'job', 'language', 'comment'):
    if not field in form:
        data[i] = '(unknown)'
    else:
        if not isinstance(form[field], list):
            data[i] = form[field].value
        else:
            values = [x.value for x in form[field]]
            data[i] = ' and '.join(values)
    i+=1
for el in data:
    print ('<td> %s </td>%s el)

print ('</tr> </table>')
```

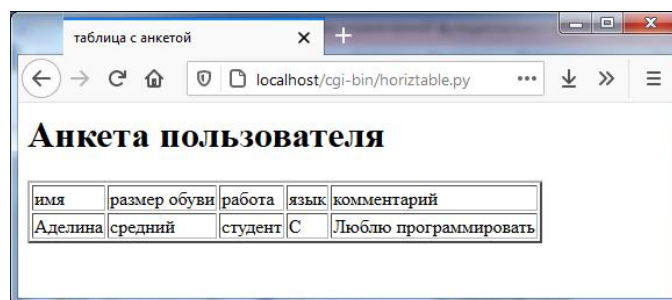


Рис. 5 Ответ сервера. Вывод введенных данных анкеты в виде горизонтальной таблицы.

Таблица 1. **Варианты заданий. Лр 5**

№	Метод	Оформление вывода данных	Дополнительный элемент формы
1	Get	Построчно	file
2	Get	Горизонтальная таблица	file
3	Get	Вертикальная таблица	file
4	Post	Построчно	file
5	Post	Горизонтальная таблица	file
6	Post	Вертикальная таблица	file
7	Get	Построчно	hidden
8	Get	Горизонтальная таблица	hidden
9	Get	Вертикальная таблица	hidden
10	Post	Построчно	hidden
11	Post	Горизонтальная таблица	hidden
12	Post	Вертикальная таблица	hidden
13	Get	Построчно	image
14	Get	Горизонтальная таблица	image
15	Get	Вертикальная таблица	image
16	Post	Построчно	image
17	Post	Горизонтальная таблица	image
18	Post	Вертикальная таблица	image
19	Get	Построчно	password
20	Get	Горизонтальная таблица	password
21	Get	Вертикальная таблица	password
22	Post	Построчно	password
23	Post	Горизонтальная таблица	password
24	Post	Вертикальная таблица	password
25	Get	Построчно	reset
26	Get	Горизонтальная таблица	reset
27	Get	Вертикальная таблица	reset
28	Post	Построчно	reset
29	Post	Горизонтальная таблица	reset
30	Post	Вертикальная таблица	reset

Распределение вариантов для лабораторной работы № 5 по курсу «Web-технологии»

Весна 2025

Вариант	Гр 4231	Гр 4232
1	Алексеев Леонид Алексеевич	Баранов Даниил Григорьевич
2	Алин Даниил Сергеевич	Козлов Антон Вячеславович
3	Бельденко София Енисеевна	Коновалова Анастасия Васильевна
4	Васильев Святослав Сергеевич	Костылев Владимир Антонович
5	Гаврилов Дмитрий Васильевич	Криковцов Юрий Алексеевич
6	Ермаков Вячеслав Андреевич	Кузнецов Степан Дмитриевич
7	Ефремов Андрей Сергеевич	Кулагина Полина Андреевна
8	Ильин Фёдор Александрович	Куриш Михаил Викторович
9	Кашколда Роман Сергеевич	Лавров Святослав Александрович
10	Кириллов Руслан Юрьевич	Лапин Ярослав Анатольевич
11	Козлов Артем Борисович	Майоров Арсений Андреевич
12	Козырев Никита Александрович	Матвеев Андрей Владимирович
13	Кондратьев Кирилл Вадимович	Матюшков Григорий Петрович
14	Кочемиров Максим Алексеевич	Махмудова Милена Ильясовна
15	Овчинникова Лилия Николаевна	Мельник Матвей Ильич
16	Потапов Сергей Антонович	Михайлов Даниил Игоревич
1	Проурзин Павел Сергеевич	Нефедьев Илья Иннокентьевич
2	Тимофеев Александр Вячеславович	Островский Максим Борисович
3	Ткачев Иван Александрович	Порохняк Михаил Дмитриевич
4	Ушаков Даниил Павлович	Савельев Владислав Максимович
5	Чеверда Илья Александрович	Салаев Никита Леонидович
6	Черных Дарья Юрьевна	Семенов Александр Алексеевич
7	Чистякова Кристина Александровна	Сотов Вадим Сергеевич
8	Шалья Владислав Юрьевич	Спицов Андрей Николаевич
9	Шведов Егор	Уткина Екатерина Алексеевна

	Александрович	
10	Эль-Хейба Денис Фади Абдаллаевич	Цыбин Дмитрий Андреевич
11	Бельчиков Никита Андреевич	Юзвенко Алина Дмитриевна
12	Саидов Саъди Толибович	Юрагин Никита Валерьевич
13	Мазур Богдан Владиславович	Чемоданова Виктория Николаевна

Вариант	Гр 4233	
1	Ануфриев Максим Антонович	
2	Букаев Сергей Артемович	
3	Гожев Дмитрий Сергеевич	
4	Голанова Софья Владимировна	
5	Григорьев Данила Алексеевич	
6	Далибаева Алина Джамалидиновна	
7	Ерыгин Даниил Юрьевич	
8	Зорихин Кирилл Николаевич	
9	Зубков Илья Ярославович	
10	Зуев Руслан Дмитриевич	
11	Косташ Ренат Вадимович	
12	Котков Владислав Владимирович	
13	Кузнецов Владислав Дмитриевич	
14	Ларионовский Роман Эдуардович	
15	Ларченкова Анна Александровна	
16	Лузанов Никита Алексеевич	
1	Магеров Антон Сергеевич	
2	Осинкин Евгений Антонович	
3	Павлов Сергей Романович	
4	Патрухин Алексей Ильич	
5	Романовский Кирилл Аркадьевич	
6	Садриев Амир Радимович	
7	Симонов Дмитрий Алексеевич	
8	Смирнов Владислав Дмитриевич	
9	Типеева Аделина Ринатовна	
10	Чавычалов Максим Евгеньевич	
11	Стадник Евгений Петрович	
12	Эркинов Дониёр Бахтиёр Угли	

Вариант	Гр 4236	Гр 4331
1	Атюцкий Никита Михайлович	Басавин Игорь Владимирович
2	Батин Владислав Сергеевич	Бердицкий Дмитрий Игоревич
3	Борщ Никита Сергеевич	Вартанян Давид Эдуардович
4	Жуковский Сергей Александрович	Голдашевский Никита Степанович
5	Завгородний Александр Романович	Денисов Ерофей Павлович
6		Зиновьев Даниил Артемович
7	Карпова Мария Евгеньевна	Карлов Всеволод Григорьевич
8	Кельбина Ксения Анатольевна	Костин Максим Юрьевич
9	Колесникова Лариса Эдуардовна	Кумпель Артём Геннадьевич
10	Кудряшов Роман Сергеевич	Кушигина Анастасия Антоновна
11	Литвинова Анастасия Михайловна	Ламбрехт Михаил Олегович
12	Лукина Анастасия Андреевна	Лекомцева Юлия Максимовна
13	Лулаков Даниил Феликсович	Маленков Михаил Сергеевич
14	Мачульский Даниил Романович	Микадзе Виктория Валерьевна
15	Николаев Артём Александрович	Новицкий Роман Александрович
16	Панин Артур Константинович	Орлов Богдан Иванович
1	Полпудников Владислав Олегович	Пресняков Егор Степанович
2	Супруненко Георгий Александрович	Сергеева Елена Александровна
3	Щукин Антон Александрович	Смирнов Константин Алексеевич
4	Юрьев Роман Андреевич	Трубников Александр Вячеславович
5	Мвале Лисон	Хазиев Ренат Рамазанович
6	Панченко Андрей Викторович	Чернов Дмитрий Андреевич
7	Валиева Дарина Рафаиловна	Исупов Дмитрий Русланович
8	Назарова Дарья Вадимовна	
9		
10		

Вариант	Гр 4332	Гр 4333
1	Артикуленко Владимир Алексеевич	Аргунов Артем Никитич
2	Беседин Роман Михайлович	Быков Александр Сергеевич
3	Болдакова Александра Евгеньевна	Власюк Анастасия Николаевна
4	Будковой Илья Павлович	Володин Глеб Владимирович
5	Васильев Артём Олегович	Волокитин Евгений Юрьевич
6	Верещагин Никита Борисович	Ворошилов Даниил Николаевич
7	Дегтярь Роман Иванович	Гаршина Ангелина Элизбаровна
8	Добрынин Сергей Дмитриевич	Герасимов Сергей Андреевич
9	Каташов Владислав -	Дорофеев Александр Вячеславович
10	Кононенко Святослав Вадимович	Еничев Егор Олегович
11	Коржавин Константин Сергеевич	Кичкина Евгения Денисовна
12	Крапивка Сергей Игоревич	Коломейцев Сергей Алексеевич
13	Лютон Артем Александрович	Корда Александр Евгеньевич
14	Мартынов Александр Александрович	Лобанов Артём Олегович
15	Мигунов Максим Сергеевич	Мигунова Наталия Денисовна
16	Никулина Юлия Эдуардовна	Могилатов Семен Игоревич
1	Панаев Владимир Владимирович	Сон Эдвард Сергеевич
2	Понушков Александр Алексеевич	Сулейманов Раджаб Русланович
3	Саиткулов Дмитрий Игоревич	Сыров Даниель Анджеевич
4	Топорина Анна Владимировна	Хаткевич Александр Евгеньевич
5	Чубарь Дмитрий Васильевич	
6		
7		
8		
9		
10		