

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ
(КАФЕДРА 43)

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

Старший преподаватель
должность, уч. степень, звание

подпись, дата

Н.А. Соловьева
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

«Серверные сценарии. Форма и обработка get и post запросов»

по дисциплине: Web-технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4236
номер группы

подпись, дата

Л. Мвале
инициалы, фамилия

Санкт-Петербург 2025

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

1 Варианты задания

Таблица 1 – Тема сайта

№ варианта	Тема
21	Промышленные роботы

Таблица 2 – Вариант задания

№	Метод	Оформление вывода данных	Дополнительный элемент формы
21	Post	Горизонтальная таблица	file

2 Скриншоты

Главная Промышленный робот Используемые источники лр3 анкета таблица из БД

Заполните анкету

Фамилия:

Имя:

Отчество:

Группа:

Какой язык программирования вы предпочитаете? ☐ Python ☐ Java ☐ C++

Какие технологии вы используете? ☐ IoT ☐ AI ☐ Robotics ☐ Automation

Email:

Profile Picture: Файл не выбран

Загрузите файл: Файл не выбран

Рисунок 1 – Исходная страница формы form.html

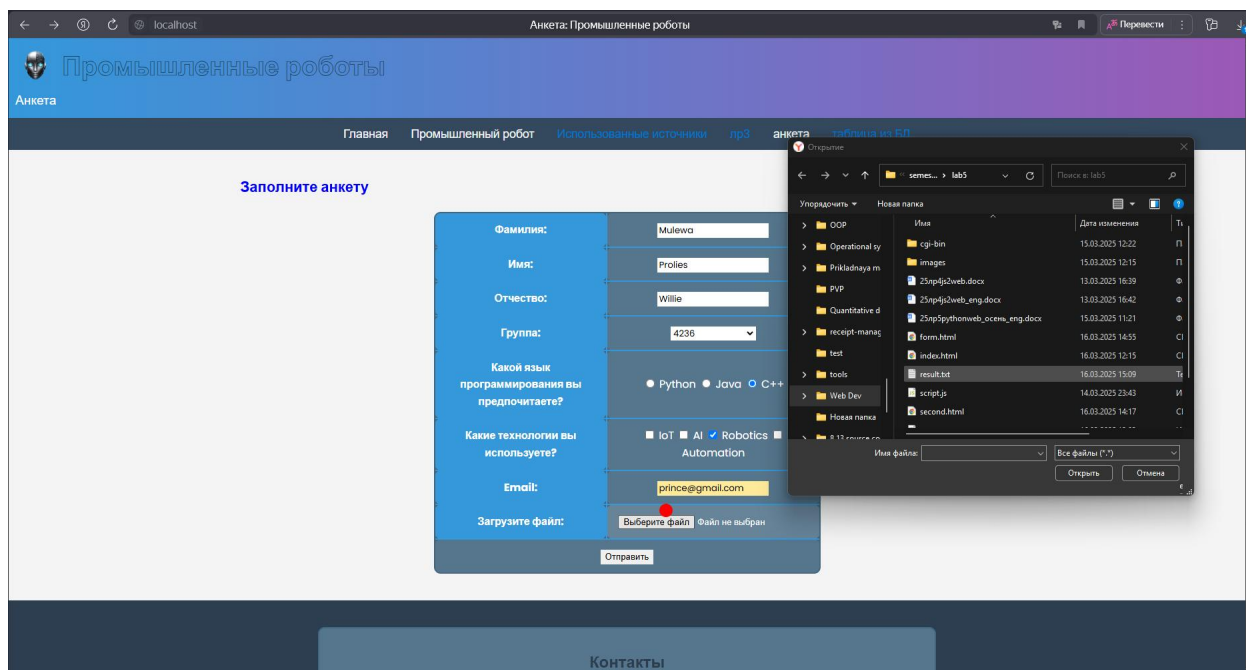


Рисунок 2 – Заполненная форма

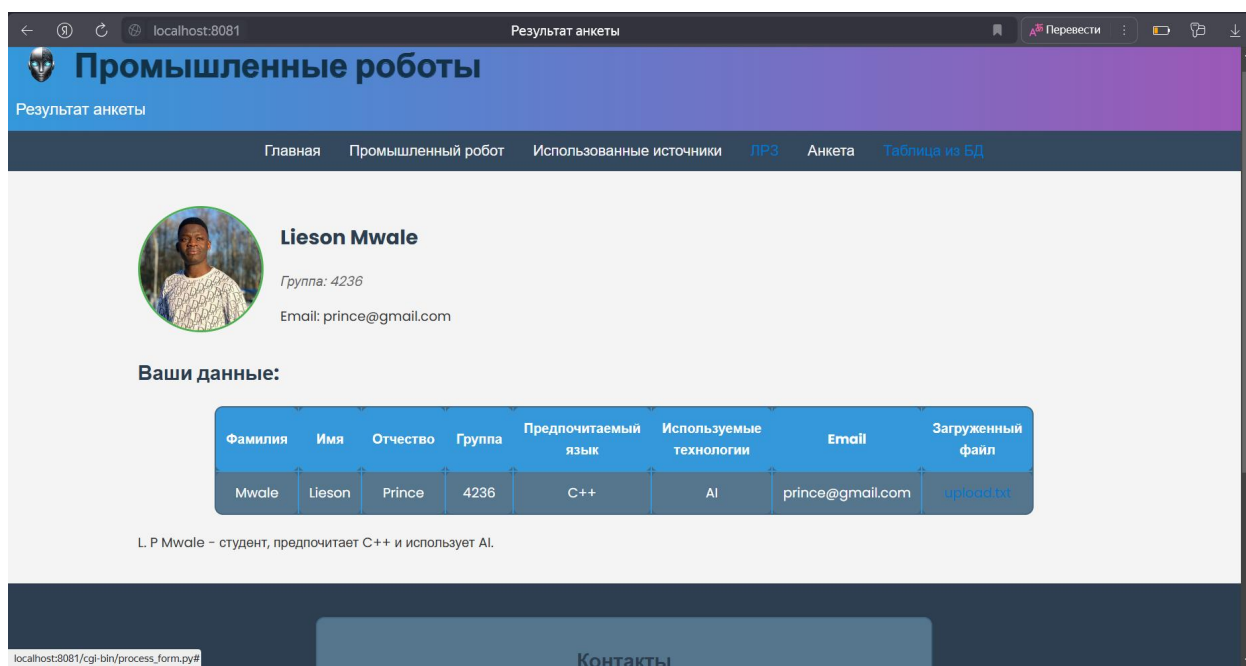


Рисунок 3 – Страница, сформированная веб-сервером 1 часть

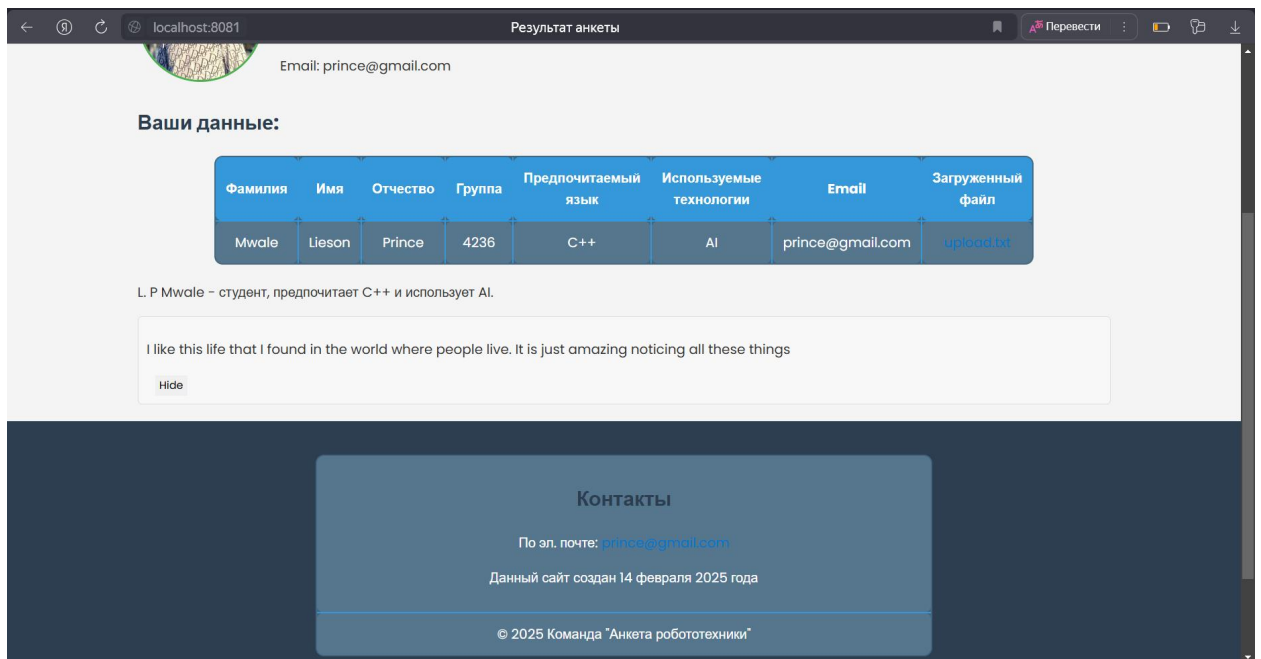


Рисунок 3 – Страница, сформированная веб-сервером 2 часть

3 Листинг

Листинг функции проверки валидации

```
function formVal(form) {
    // Validate surname
    if (form.surname.value.length < 4 || form.surname.value.length > 30) {
        alert("Фамилия должна содержать от 4 до 30 символов.");
        form.surname.focus();
        return false;
    }
    if (!form.surname.value.match(/^[a-zA-Za-яA-Я]+$/i)) {
        alert("Фамилия должна содержать только буквы.");
        form.surname.focus();
        return false;
    }

    // Validate name
    if (form.name.value.length < 4 || form.name.value.length > 30) {
        alert("Имя должно содержать от 4 до 30 символов.");
        form.name.focus();
        return false;
    }
    if (!form.name.value.match(/^[a-zA-Za-яA-Я]+$/i)) {
        alert("Имя должно содержать только буквы.");
        form.name.focus();
        return false;
    }
}
```

```

    }

    // Validate patronymic
    if (form.patron.value.length < 4 || form.patron.value.length > 30) {
        alert("Отчество должно содержать от 4 до 30 символов.");
        form.patron.focus();
        return false;
    }
    if (!form.patron.value.match(/^[a-zA-Za-яА-Я]+$/i)) {
        alert("Отчество должно содержать только буквы.");
        form.patron.focus();
        return false;
    }

    // Validate group selection
    if (form.group.selectedIndex == 0) {
        alert("Пожалуйста, выберите вашу группу.");
        return false;
    }

    // Validate radio buttons
    if (!form.question1[0].checked && !form.question1[1].checked
    && !form.question1[2].checked) {
        alert("Пожалуйста, выберите вариант ответа.");
        return false;
    }

    // Validate checkboxes
    if (!form.question2[0].checked && !form.question2[1].checked
    && !form.question2[2].checked && !form.question2[3].checked) {
        alert("Пожалуйста, выберите хотя бы один вариант ответа.");
        return false;
    }

    // Validate email
    if (!form.email.value.match(/^[^\s@]+@[^\s@]+\.[^\s@]+$/)) {
        alert("Пожалуйста, введите корректный email.");
        form.email.focus();
        return false;
    }

    return true;
}

```

Листинг страницы формы form.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta
    charset="UTF-8"

```

```

        name="viewport"
        content="width=device-width, initial-scale=1.0" />
<title>Анкета</title>
<link rel="stylesheet" href="css/styles.css">
<link rel="icon" href="images/icon.ico" type="image/x-icon">
<script type="text/javascript" src="js/events.js"></script>
</head>
<body>
<!-- Шапка сайта с названием и логотипом -->
<header>
    <a href="#" class="logo-container">
        
        <span class="logo-text">Промышленные роботы</span>
    </a>
    <p>Анкета</p>
</header>

<!-- Навигационная панель с ссылками на разделы сайта -->
<nav align="center" >
    <a href="index.html">Главная</a>
    <a href="second.html">Промышленный робот</a>
    <a href="source.html">Использованные источники</a>
    <a href="#">лр3</a>
    <a href="form.html">анкета</a>
    <a href="#">таблица из БД</a>
</nav>
<main>
    <h2>Заполните анкету</h2>
    <form method="POST" action="http://localhost:8081/cgi-bin/process_form.py"
enctype="multipart/form-data" onsubmit="return formVal(this);">
        <table>
            <tr>
                <th>Фамилия:</th>
                <td><input type="text" name="surname" placeholder="Введите
фамилию"></td>
            </tr>
            <tr>
                <th>Имя:</th>
                <td><input type="text" name="name" placeholder="Введите имя"></td>
            </tr>
            <tr>
                <th>Отчество:</th>
                <td><input type="text" name="patron" placeholder="Введите
отчество"></td>
            </tr>
            <tr>
                <th>Группа:</th>
                <td>
                    <select name="group">
                        <option value="">Выберите группу</option>
                        <option value="4236">4236</option>
                        <option value="Z0431">Z0431</option>
                    </select>
                </td>
            </tr>
        </table>
    </form>

```

```

        <option value="Z0431K">Z0431K</option>
    </select>
</td>
</tr>
<tr>
<th>Какой язык программирования вы предпочитаете?</th>
<td>
    <input type="radio" name="question1" value="Python"> Python
    <input type="radio" name="question1" value="Java"> Java
    <input type="radio" name="question1" value="C++"> C++
</td>
</tr>
<tr>
<th>Какие технологии вы используете?</th>
<td>
    <input type="checkbox" name="question2" value="IoT"> IoT
    <input type="checkbox" name="question2" value="AI"> AI
    <input type="checkbox" name="question2" value="Robotics"> Robotics
    <input type="checkbox" name="question2" value="Automation">
Automation
</td>
</tr>
<tr>
<th>Email:</th>
<td><input type="email" name="email" placeholder="Введите
email"></td>
</tr>
<tr>
<th>Profile Picture:</th>
<td><input type="file" name="profile_pic" accept="image/*"></td>
</tr>
<tr>
<th>Загрузите файл:</th>
<td><input type="file" name="file"></td>
</tr>
<tr>
<td colspan="2"><input type="submit" value="Отправить"></td>
</tr>
</table>
</form>
</main>
<!-- Подвал сайта с контактной информацией -->
<footer>
    <table width="100%" border="0">
        <tr>
            <td>
                <address>
                    <h2>Контакты</h2>
                    <p>C
                        <button class="author-button" name="author-
info">АВТОРОМ</button> сайта можно связаться:</p>
                    <address>

```

```

        По эл. почте: <a
href="mailto:lieson.mwale22.ru">lieson.mwale22@gmail.com</a><br />
        По телефону: <a href="tel:+78005553535">+7(800)555-35-35</a>
        </address>
        <p>Данный сайт создан 14 февраля 2025 года</p>
        </address>
    </td>
</tr>
<tr>
<td>&copy; 2025 Команда "История робототехники"</td>
</tr>
</table>
</footer>
</body>
<script src="script.js" defer></script>
</html>

```

Листинг серверного сценария

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import cgi
import cgitb
import codecs
import os
import html

cgitb.enable() # Enable error reporting

# Ensure stdout uses UTF-8
import sys
sys.stdout = codecs.getwriter("utf-8")(sys.stdout.detach())

# Create directories if they don't exist
if not os.path.exists("uploads"):
    os.makedirs("uploads")
if not os.path.exists("profile_pics"):
    os.makedirs("profile_pics")

# Get form data
form = cgi.FieldStorage()

# Extract values and decode to UTF-8
def get_utf8_value(field_name, default="(unknown)"):
    value = form.getvalue(field_name, default)
    return value.encode("latin1").decode("utf-8") if isinstance(value, str) else default

# Extract values
surname = get_utf8_value('surname')

```



```

name = get_utf8_value('name')
patron = get_utf8_value('patron')
group = get_utf8_value('group')
question1 = get_utf8_value('question1')
question2 = form.getlist('question2')
email = get_utf8_value('email')

# Convert checkboxes list to UTF-8
question2 = [q.encode("latin1").decode("utf-8") for q in question2]

# Handle file upload
uploaded_filename = "(no file uploaded)"
file_content = ""
if "file" in form:
    uploaded_file = form["file"]
    if uploaded_file.filename:
        file_path = os.path.join("uploads", uploaded_file.filename)
        try:
            with open(file_path, "wb") as f:
                f.write(uploaded_file.read())
            uploaded_filename = uploaded_file.filename
            # Read file content for display
            try:
                with open(file_path, "r", encoding="utf-8") as f:
                    file_content = f.read()
            except:
                file_content = "[Binary file - content not displayable]"
        except Exception as e:
            uploaded_filename = f"Error saving file: {e}"

# Handle profile picture upload
profile_pic = ""
if "profile_pic" in form:
    pic_file = form["profile_pic"]
    if pic_file.filename:
        pic_path = os.path.join("profile_pics", pic_file.filename)
        try:
            with open(pic_path, "wb") as f:
                f.write(pic_file.read())
            profile_pic = pic_file.filename
        except Exception as e:
            profile_pic = f"Error saving profile picture: {e}"

# Save data to a UTF-8 file
try:
    with open("result.txt", "a", encoding="utf-8") as f:
        f.write(f'{surname}|{name}|{patron}|{group}|{question1}|{','.join(question2)}|{email}\n')
except Exception as e:
    print("Content-Type: text/plain\r\n\r\n")
    print(f"Error saving data: {e}")
    exit()

```

```

# Prepare profile picture HTML
if profile_pic and not profile_pic.startswith("Error"):
    profile_html = f''
else:
    profile_html = '<div class="profile-pic" style="background:#eee; display:flex; align-
items:center; justify-content:center;">No Image</div>'

# Prepare file content HTML
if uploaded_filename != "(no file uploaded)":
    escaped_filename = html.escape(uploaded_filename)
    file_link_html = f'<a href="#" onclick="toggleFileContent('{escaped_filename}')';
return false;">{uploaded_filename}</a>'
    file_content_html = f'
    <div id="file-content-{escaped_filename}" class="file-content">
        <pre>{html.escape(file_content)}</pre>
        <button onclick="hideFileContent('{escaped_filename}')" class="toggle-
btn">Hide</button>
    </div>
    '
else:
    file_link_html = uploaded_filename
    file_content_html = ""

# Generate response
print("Content-Type: text/html; charset=utf-8\r\n\r\n")
print(f'<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Результат анкеты</title>
    <link rel="stylesheet" href="../styles.css">
    <style>
        .profile-container {{
            display: flex;
            gap: 20px;
            margin: 20px 0;
            align-items: center;
        }}
        .profile-pic {{
            width: 150px;
            height: 150px;
            border-radius: 50%;
            object-fit: cover;
            border: 3px solid #4CAF50;
        }}
        .profile-info {{
            flex-grow: 1;
        }}
        .file-content {{
            display: none;

```

```

        background: #f8f8f8;
        padding: 10px;
        border: 1px solid #ddd;
        margin-top: 10px;
        border-radius: 5px;
    }}
    .toggle-btn {{
        margin-left: 10px;
        padding: 2px 5px;
        font-size: 0.8em;
    }}
    .file-display-section {{
        margin-top: 20px;
        padding: 15px;
        background: #f5f5f5;
        border-radius: 5px;
    }}
</style>
</head>
<body>
    <!-- Шапка сайта -->
    <header>
        <a href="#" class="logo-container">
            
            <span class="logo-text">Промышленные роботы</span>
        </a>
        <p>Результат анкеты</p>
    </header>

    <!-- Навигация -->
    <nav align="center">
        <a href="../index.html">Главная</a>
        <a href="../second.html">Промышленный робот</a>
        <a href="../source.html">Использованные источники</a>
        <a href="#">ЛР3</a>
        <a href="../form.html">Анкета</a>
        <a href="#">Таблица из БД</a>
    </nav>

    <main>
        <!-- Profile Section -->
        <div class="profile-container">
            <div>
                {profile_html}
            </div>
            <div class="profile-info">
                <h2>{name} {surname}</h2>
                <p>Группа: {group}</p>
                <p>Email: {email}</p>
            </div>
        </div>
    </main>

```

```

<h2>Ваши данные:</h2>
<table border="1">
  <tr>
    <th>Фамилия</th>
    <th>Имя</th>
    <th>Отчество</th>
    <th>Группа</th>
    <th>Предпочитаемый язык</th>
    <th>Используемые технологии</th>
    <th>Email</th>
    <th>Загруженный файл</th>
  </tr>
  <tr>
    <td>{surname}</td>
    <td>{name}</td>
    <td>{patron}</td>
    <td>{group}</td>
    <td>{question1}</td>
    <td>{' , '.join(question2)}</td>
    <td>{email}</td>
    <td>{file_link_html}</td>
  </tr>
</table>

```

<p>{name[0]}. {patron[0]} {surname} - студент, предпочитает {question1} и использует {' , '.join(question2)}.</p>

```

<!-- File content display section -->
{file_content_html if uploaded_filename != "(no file uploaded)" else ""}
</main>

<footer>
  <table width="100%" border="0">
    <tr>
      <td>
        <address>
          <h2>Контакты</h2>
          <address>
            По эл. почте: <a href="mailto:lieson.mwale22.ru">{email}</a><br />
          </address>
          <p>Данный сайт создан 14 февраля 2025 года</p>
        </address>
      </td>
    </tr>
    <tr>
      <td>&copy; 2025 Команда "Анкета робототехники"</td>
    </tr>
  </table>
</footer>

```

```

<script>
  function toggleFileContent(filename) {{

```

```

        const contentDiv = document.getElementById(`file-content-
${{encodeURIComponent(filename)}}`);
        if (contentDiv.style.display === 'block') {{
            contentDiv.style.display = 'none';
        }} else {{
            contentDiv.style.display = 'block';
        }}
    }}

    function hideFileContent(filename) {{
        document.getElementById(`file-content-
${{encodeURIComponent(filename)}}`).style.display = 'none';
    }}
</script>
</body>
</html>""")

```

Листинг сервера

```

import os
import sys
from http.server import HTTPServer, CGIHTTPRequestHandler

class MyCGIHandler(CGIHTTPRequestHandler):
    cgi_directories = ["/cgi-bin"] # Ensure this is correctly set

# Set the web directory and port
webdir = '.'
port = 8081

if len(sys.argv) > 1:
    webdir = sys.argv[1]
if len(sys.argv) > 2:
    port = int(sys.argv[2])

print(f'Web directory: {webdir}, Port: {port}')

os.chdir(webdir) # Change to web directory

server_address = ("", port)
httpd = HTTPServer(server_address, MyCGIHandler) # Use the custom handler
print(f'Server running on port {port}...')
httpd.serve_forever()

```

Вывод

В ходе выполнения лабораторной работы был получен опыт по написанию и применению серверных сценариев для обработки данных, введенных пользователем. А также выполнены дополнительные условия из расширенного задания.