

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

ассистент				В. В. Жукалин
должность, уч. степень, звание		подпись, дата		инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

ВВЕДЕНИЕ В WEB-ПРОГРАММИРОВАНИЕ

Вариант 3

по курсу: ОБРАБОТКА ДАННЫХ HTML-ФОРМ В NODE.JS

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4128			Воробьев В. А.
			подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1 Введение	3
1.1 Цель работы	3
2 Выполнение работы	4
ВЫВОД	8
ПРИЛОЖЕНИЕ	9

1 Введение

1.1 Цель работы

Создать папку с проектом, содержащую страницу с HTML-формой, каскадной таблицей стилей (CSS) и JS-скриптом в среде Node.JS.

- Разработать web-страницу, которая должна предлагать пользователю HTML-форму, включающую различные элементы (текстовые поля, списки, кнопки и т.д.). HTML-форма должна содержать не менее 10 различных элементов. HTML-форма должна иметь дизайн, разработанный средствами CSS. Тематика, для которой создается форма, определяется в соответствии с приложением А. HTML-форма должна иметь обязательные поля (input type=“text”) в соответствии с разделом «Варианты для лабораторных работ». РЕКОМЕНДУЕТСЯ использовать форму из Лабораторной работы № 5 «Создание форм в web-документах» по дисциплине «Web-технологии».

2 Выполнение работы

В результате получилось создать страницы, изображенные на рис. 1-3. При неправильном вводе или отсутствии ввода там, где это необходимо, выводится соответствующее сообщение об этом. Для создания таблицы стилей для формы была использована технология flexbox. Результирующий код представлен в Приложении.

Registration Form

Name:

sdasdas

Email:

Password:

Gender:

Male

Birthdate:

02/12/2024

Country:

Bio:

Subscribe to newsletter:

☐

Profile Picture:

Choose File no file selected

Submit

Рисунок 2.1 - Главный экран ввода формы

Registration Successful

Name: sdasdas

Email: askdjosijdj@dsam.com

Gender: male

Birthdate: 2024-12-10

Country: dsaasd

Bio: dsadsaadsads

Newsletter: Not Subscribed



Рисунок 2.2 - Экран отправки формы

Registration Form

Name:

saddsaasd

Email:

sadsadsad

Enter an email address

Password:

••••••••

Gender:

Male

Birthdate:

02/12/2024

Country:

Bio:

Subscribe to newsletter:

☐

Profile Picture:

Choose File no file selected

Submit

Рисунок 2.3 - Валидация формы

ВЫВОД

В ходе выполнения лабораторной работы были изучены основные принципы работы с сервером на базе Node.js. Был настроен сервер для обработки данных, отправленных через HTML-форму методом POST, что позволило эффективно обрабатывать ввод пользователя и формировать динамическую страницу с результатами. В процессе работы использовался модуль `body-parser` для корректного парсинга данных формы, что обеспечило корректную обработку информации, полученной от пользователя.

Данная работа позволила закрепить навыки организации серверной части приложения, обработки POST-запросов для получения данных из HTML-формы и дальнейшего формирования страницы с результатами.

ПРИЛОЖЕНИЕ

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-
   scale=1.0">
7   <title>Web Form</title>
8   <link rel="stylesheet" href="styles.css">
9 </head>
10
11 <body>
12   <div class="container">
13     <h1>Registration Form</h1>
14     <form id="registrationForm" action="/submit" method="POST
      ">
15       <label for="name">Name:</label>
16       <input type="text" id="name" name="name" required><br>
17
18       <label for="email">Email:</label>
19       <input type="email" id="email" name="email" required><
        br>
20
21       <label for="password">Password:</label>
22       <input type="password" id="password" name="password"
        required><br>
23
24       <label for="gender">Gender:</label>
25       <select id="gender" name="gender">
26         <option value="male">Male</option>
27         <option value="female">Female</option>
28         <option value="other">Other</option>
29       </select><br>
30
31       <label for="birthdate">Birthdate:</label>
32       <input type="date" id="birthdate" name="birthdate"><br>
33
34       <label for="country">Country:</label>
35       <input type="text" id="country" name="country"><br>
36
```

```

37     <label for="bio">Bio:</label>
38     <textarea id="bio" name="bio"></textarea><br>
39
40     <label for="newsletter">Subscribe to newsletter:</label>
41     <input type="checkbox" id="newsletter" name="newsletter"
42           "><br>
43
44     <label for="profilePicture">Profile Picture:</label>
45     <input type="file" id="profilePicture" name="
46           profilePicture"><br>
47
48     <input type="submit" value="Submit">
49     </form>
50 </div>
51 <script src="script.js"></script>
52 </body>
53 </html>

```

```

1 body {
2   font-family: Arial, sans-serif;
3   background-color: #f4f4f4;
4   margin: 0;
5   padding: 0;
6 }
7
8 .container {
9   max-width: 600px;
10  margin: 50px auto;
11  padding: 20px;
12  background-color: #fff;
13  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
14 }
15
16 h1 {
17   text-align: center;
18 }
19
20 form {
21   display: flex;
22   flex-direction: column;

```

```

23     }
24
25     label {
26         margin-top: 10px;
27     }
28
29     input, select, textarea {
30         padding: 10px;
31         margin-top: 5px;
32         border: 1px solid #ccc;
33         border-radius: 4px;
34     }
35
36     input[type="submit"] {
37         background-color: #4CAF50;
38         color: white;
39         border: none;
40         cursor: pointer;
41     }
42
43     input[type="submit"]:hover {
44         background-color: #45a049;
45     }
46
47     img {
48         display: block;
49         margin: 20px auto;
50         max-width: 100%;
51         height: auto;
52     }

```

```

1 document
2   .getElementById("registrationForm")
3   .addEventListener("submit", function () {
4       alert("Form submitted successfully!")
5   })

```

```

1 document.addEventListener("DOMContentLoaded", () => {
2     const urlParams = new URLSearchParams(window.location.
        search)
3
4     document.getElementById("name").textContent = urlParams.get
        ("name")

```

```

5    document.getElementById("email").textContent = urlParams .
      get("email")
6    document.getElementById("gender").textContent = urlParams .
      get("gender")
7    document.getElementById("birthdate").textContent =
      urlParams.get("birthdate")
8    document.getElementById("country").textContent = urlParams .
      get("country")
9    document.getElementById("bio").textContent = urlParams.get
      ("bio")
10   document.getElementById("newsletter").textContent =
11     urlParams.get("newsletter") === "true" ? "Subscribed" : "
      Not Subscribed"
12
13   const imageSrc =
14     urlParams.get("newsletter") === "true" ? "checked.png" :
      "unchecked.png"
15   document.getElementById("subscriptionImage").src = imageSrc
16 })

```

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-
      scale=1.0">
6    <title>Result Page</title>
7    <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10   <div class="container">
11     <h1>Registration Successful</h1>
12     <p><strong>Name:</strong> <span id="name"></span></p>
13     <p><strong>Email:</strong> <span id="email"></span></p>
14     <p><strong>Gender:</strong> <span id="gender"></span></p>
15     <p><strong>Birthdate:</strong> <span id="birthdate"></
      span></p>
16     <p><strong>Country:</strong> <span id="country"></span></
      p>
17     <p><strong>Bio:</strong> <span id="bio"></span></p>
18     <p><strong>Newsletter:</strong> <span id="newsletter"></
      span></p>

```

```

19     <img id="subscriptionImage" src="" alt="Subscription
        Status">
20   </div>
21   <script src="resultScript.js"></script>
22 </body>
23 </html>

```

```

1  const express = require("express")
2  const bodyParser = require("body-parser")
3  const path = require("path")
4  const app = express()
5  const PORT = 3000
6
7  app.use(bodyParser.urlencoded({ extended: true }))
8  app.use(express.static(path.join(__dirname, "public")))
9
10 app.get("/", (req, res) => {
11   res.sendFile(path.join(__dirname, "public", "index.html"))
12 })
13
14 app.post("/submit", (req, res) => {
15   const formData = req.body
16   const errors = validateForm(formData)
17
18   if (errors.length > 0) {
19     res.send(
20       '<h1>Validation Errors</h1><ul>${errors
21         .map((error) => '<li>${error}</li>')
22         .join("")}</ul>'
23     )
24   } else {
25     const queryParams = new URLSearchParams(formData).
26       toString()
27     res.redirect(`/result.html?${queryParams}`)
28   }
29 })
30
31 app.listen(PORT, () => {
32   console.log(`Server is running on http://localhost:${PORT}
33     `)
34 })

```

```
34 function validateForm(formData) {  
35     const errors = []  
36     if (!formData.name) errors.push("Name is required.")  
37     if (!formData.email) errors.push("Email is required.")  
38     if (!formData.password) errors.push("Password is required  
    .")  
39     return errors  
40 }
```