

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

В.В. Жукалин

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

РАБОТА С СЕССИЯМИ И ФАЙЛАМИ COOKIE В NODE.JS

по курсу: WEB-ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4128

подпись, дата

В.А. Воробьев

инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Изучить функции языка для работы с сессиями, файлами cookie, а также работу с несколькими файлами.

Задание

- 1 Реализовать ввод данных для HTML-формы из лабораторной работы 3 в 3 шага (этапа), используя технологию сессий. Каждый шаг (этап) должен находиться на отдельной странице. Переключение страниц осуществить с помощью метода GET в среде Node.JS.
- 2 Написать JS-скрипт в среде Node.JS, который применяет функции управления сессией для запоминания того, какие страницы уже посещались пользователем. Вывести список ссылок на все посещенные страницы с указанием количества посещений.
- 3 Реализовать в среде Node.JS возможность сохранения данных, введенных пользователем в прошлый раз в HTML-форме с помощью сессий, чтобы их повторно не вводить.
- 4 Модернизировать JS-скрипт путем добавления нулевого (стартового) шага (этапа). На нулевом (стартовом) шаге (этапе) предложить пользователю указать свое имя и выбрать цвет фона страниц сайта. Сохранить эту информацию, используя файлы cookie. Далее, используя файлы cookie, на всех шагах (этапах), кроме нулевого, приветствовать пользователя по имени и использовать выбранный пользователем цвет фона. При приветствии запрещается использовать всплывающие окна

Выполнение работы

При выполнении работы была написана новая входная точка `intro`, которая позволяет пользователю выбрать имя и цвет фона. При нажатии на кнопку перекидываем на шаги, которые кешируют ввод пользователя.

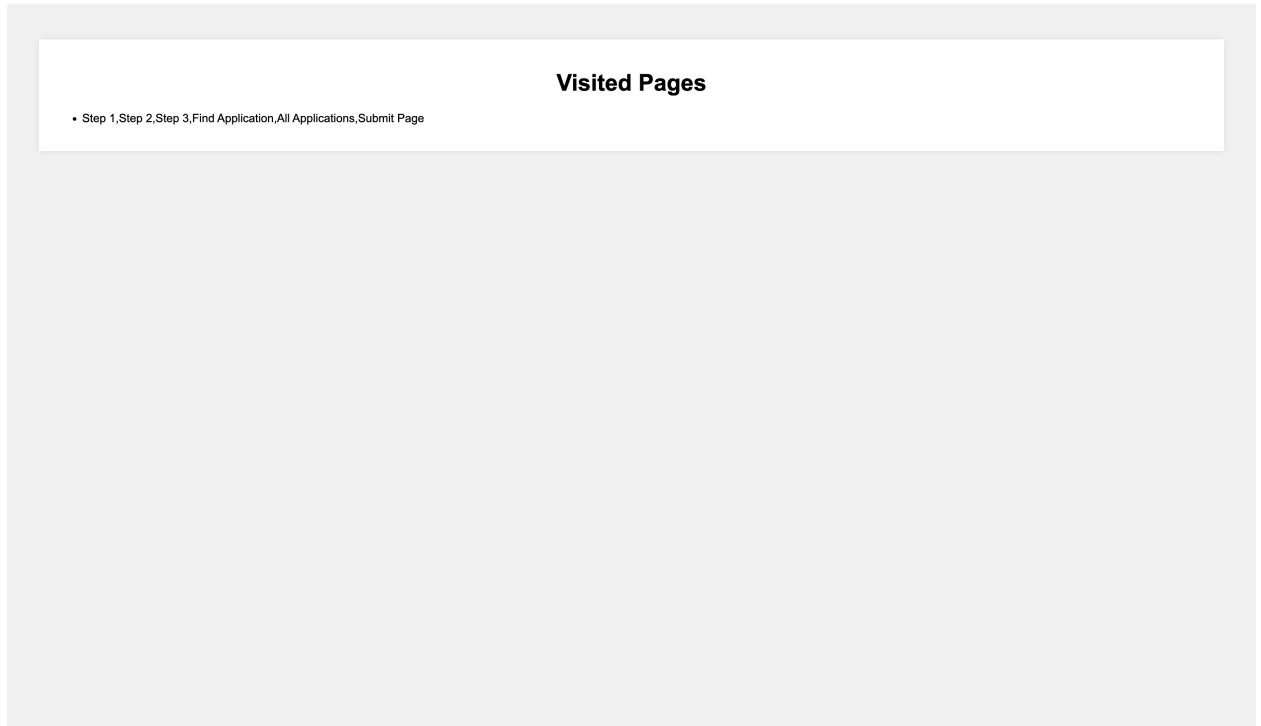


Рисунок 1 – Список посещенных страниц

Welcome

Name:

sdasddasasd

Background Color:

Next

Рисунок 2 – Начальная страница

Welcome, sdasddasasd!

Registration Step 1

Email:

dsasadsadasd@asddsa.comdsaadsasdaad

Password:

Next

Рисунок 3 – Первый шаг с выбранным цветом

Welcome, sdasddasasd!

Registration Step 2

Gender:

Male

Birthdate:

12.12.2024

Next

Рисунок 4 –Второй шаг с выбранным цветом

Welcome, sdasddasasd!

Registration Step 3

Country:

sdasdasd

Bio:

dsdasadsaad

Subscribe to newsletter:

☐

Next

Рисунок 5 – Третий шаг с выбранным цветом

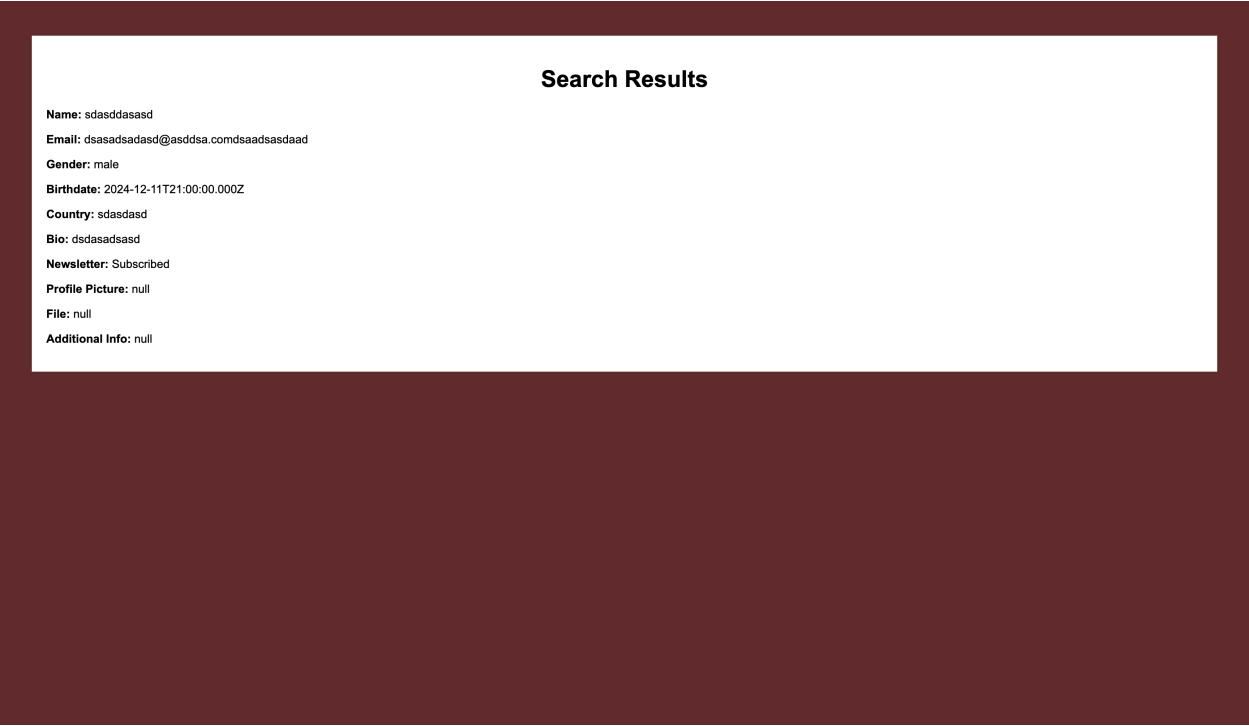


Рисунок 6 – Результат заполнения

Вывод

В процессе выполнения лабораторной работы были изучены ключевые аспекты работы с сессиями и куки в среде Node.js, а также их практическое применение для улучшения пользовательского опыта в веб-приложении. Была создана HTML-форма, разделенная на три этапа, каждый из которых находился на отдельной странице. Для навигации между этапами использовался метод GET. Введенные данные сохранялись с помощью сессий, что позволило пользователям заполнять форму поэтапно без потери информации.

Кроме того, был разработан JavaScript-скрипт для управления сессиями, который отслеживал посещенные пользователем страницы и отображал их список с указанием количества посещений. Также была реализована функция сохранения введенных данных: при повторном посещении формы данные автоматически загружались из сессий.

Для улучшения удобства и персонализации был добавлен начальный этап, на котором пользователь вводил свое имя и выбирал цвет фона для страниц. Эти данные сохранялись в куки и использовались на последующих этапах для приветствия пользователя по имени и отображения страниц в выбранном цвете фона. Выполнение работы позволило приобрести практические навыки разработки многостраничных приложений с использованием сессий и куки.

ПРИЛОЖЕНИЕ

```
const express = require("express");

const bodyParser = require("body-parser");

const path = require("path");

const multer = require("multer");

const mysql = require("mysql2");

const cookieParser = require("cookie-parser");


const app = express();

const PORT = 3000;


const uploadDir = path.join(__dirname, "uploads");


app.use(bodyParser.urlencoded({ extended: true }));

app.use(express.static(path.join(__dirname, "public")));

app.use('/uploads', express.static(uploadDir));

app.use(cookieParser());


const storage = multer.diskStorage({

  destination: (_, __, cb) => {

    cb(null, uploadDir);
```



```
    },  
  
    filename: (_, file, cb) => {  
  
        cb(null, `${Date.now()}-${file.originalname}`);  
  
    }  
  
});  
  
const fileFilter = (req, file, cb) => {  
  
    const fileExtension = path.extname(file.originalname).toLowerCase();  
  
    if (fileExtension !== '.7z') {  
  
        req.fileValidationError = 'Only .7z files are allowed!';  
  
        return cb(null, false);  
  
    }  
  
    cb(null, true);  
  
};  
  
const upload = multer({  
  
    storage: storage,  
  
    fileFilter: fileFilter,  
  
    limits: { fileSize: 1024 * 1024 * 5 }  
  
});
```

```
const db = mysql.createConnection({  
  host: "localhost",  
  user: "root",  
  password: "",  
  database: "registration_db"  
});
```

```
db.connect(err => {  
  if (err) {  
    console.error("Error connecting to the database:", err);  
    return;  
  }  
  console.log("Connected to the database");  
});
```

```
app.get("/", (_, res) => {  
  res.sendFile(path.join(__dirname, "public", "main.html"));  
});
```

```
app.get("/submit", (req, res) => {  
  const formData = {
```

```
name: req.cookies.name,  
  
email: req.cookies.email,  
  
password: req.cookies.password,  
  
gender: req.cookies.gender,  
  
birthdate: req.cookies.birthdate,  
  
country: req.cookies.country,  
  
bio: req.cookies.bio,  
  
newsletter: req.cookies.newsletter === "true",  
  
};
```

```
const errors = validateForm(formData);
```

```
if (errors.length > 0) {
```

```
  res.send(  
    `<h1>Validation Errors</h1><ul>${errors
```

```
      .map((error) => `<li>${error}</li>`)
```

```
      .join("")}</ul>`
```

```
  );
```

```
};
```

```
} else {
```

```
  const query = `INSERT INTO users (name, email, password, gender,  
    birthdate, country, bio, newsletter, profilePicture, file, additionalInfo)  
    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)`;
```

```
const values = [  
  
  formData.name,  
  
  formData.email,  
  
  formData.password,  
  
  formData.gender,  
  
  formData.birthdate,  
  
  formData.country,  
  
  formData.bio,  
  
  formData.newsletter,  
  
  formData.profilePicture,  
  
  formData.file,  
  
  formData.additionalInfo  
];  
  
db.query(query, values, (err, result) => {  
  
  if (err) {  
  
    return res.send(`<script>alert("Database error: ${err.message}");  
window.location.href="/";</script>`);  
  
  }  
  
  res.clearCookie("name");  
  
  res.clearCookie("email");  
  
  res.clearCookie("password");  
}
```

```
    res.clearCookie("gender");

    res.clearCookie("birthdate");

    res.clearCookie("country");

    res.clearCookie("bio");

    res.clearCookie("newsletter");

    res.clearCookie("profilePicture");

    res.clearCookie("file");

    res.clearCookie("additionalInfo");

    res.redirect(`/searchResults.html?searchParam=${result.insertId}`);

  });

}

});

app.get("/step1", (req, res) => {

  res.sendFile(path.join(__dirname, "public", "step1.html"));

});

app.post("/step1", (req, res) => {

  const { email, password } = req.body;

  res.cookie("email", email);

  res.cookie("password", password);
```

```
    res.redirect("/step2");

});

app.get("/step2", (req, res) => {

    res.sendFile(path.join(__dirname, "public", "step2.html"));

});

app.post("/step2", (req, res) => {

    const { gender, birthdate } = req.body;

    res.cookie("gender", gender);

    res.cookie("birthdate", birthdate);

    res.redirect("/step3");

});

app.get("/step3", (req, res) => {

    res.sendFile(path.join(__dirname, "public", "step3.html"));

});

app.post("/step3", (req, res) => {

    const { country, bio, newsletter } = req.body;

    res.cookie("country", country);
```

```
res.cookie("bio", bio);

res.cookie("newsletter", newsletter);

res.redirect("/submit");

});

app.post("/upload", upload.single('file'), (req, res) => {

  if (req.fileValidationError) {

    return res.status(400).json({ error: req.fileValidationError });

  }

  if (req.file) {

    res.cookie('file', req.file.filename, { path: '/' });

    return res.status(200).json({ message: "File uploaded successfully!",
filePath: req.file.filename });

  }

  res.status(400).json({ error: "File upload failed." });

});

app.get("/allApplications", (req, res) => {

  const query = "SELECT * FROM users";

  db.query(query, (err, results) => {

    if (err) {
```

```
        return res.send(`<script>alert("Database error: ${err.message}");
window.location.href="/";</script>`);
    }
}
```

```
let html = `
```

```
    <!DOCTYPE html>
```

```
    <html lang="en">
```

```
    <head>
```

```
        <meta charset="UTF-8">
```

```
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
        <title>All Applications</title>
```

```
        <link rel="stylesheet" href="styles.css">
```

```
    </head>
```

```
    <body>
```

```
        <div class="container">
```

```
            <h1>All Applications</h1>
```

```
            <table>
```

```
                <thead>
```

```
                    <tr>
```

```
                        <th>ID</th>
```

```
                        <th>Name</th>
```

```
                        <th>Email</th>
```


<th>Gender</th>

<th>Birthdate</th>

<th>Country</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

`;

results.forEach(user => {

html += `

<tr>

<td>\${user.id}</td>

<td>\${user.name}</td>

<td>\${user.email}</td>

<td>\${user.gender}</td>

<td>\${user.birthdate}</td>

<td>\${user.country}</td>

<td>

<a

href="/searchResults.html?searchParam=\${user.id}">View

<form action="/deleteApplication" method="POST"

style="display:inline;">

```

        <input type="hidden" name="id" value="\${user.id}">

        <input type="hidden" name="email"
value="\${user.email}">

        <button type="submit">Delete</button>

    </form>

    <form action="/updateApplication" method="POST"
style="display:inline;">

        <input type="hidden" name="id" value="\${user.id}">

        <button type="submit">Update</button>

    </form>

</td>

</tr>

`;

});

html += `

    </tbody>

</table>

</div>

</body>

</html>

`;

res.send(html);

```

```
});  
  
});  
  
app.get("/findApplication", (req, res) => {  
    res.sendFile(path.join(__dirname, "public", "findApplication.html"));  
});
```

```
app.get("/searchApplication", (req, res) => {  
    const searchParam = req.query.searchParam;  
    res.redirect(`/searchResults.html?searchParam=${searchParam}`);  
});
```

```
app.get("/getUserByParam", (req, res) => {  
    const searchParam = req.query.searchParam;  
    const query = "SELECT * FROM users WHERE id = ? OR email = ?";  
    db.query(query, [searchParam, searchParam], (err, result) => {  
        if (err) {  
            return res.status(500).json({ error: "Database error" });  
        }  
        res.json(result[0]);  
    });
```

```
});
```

```
app.get("/deleteApplication", (req, res) => {  
  
  const { id, email } = req.body;  
  
  const query = "DELETE FROM users WHERE id = ? AND email = ?";  
  
  db.query(query, [id, email], (err, result) => {  
  
    if (err) {  
  
      return res.send(`<script>alert("Database error:  
${err.message}");</script>`);  
  
    }  
  
    res.send(`<script>alert("Record deleted successfully!");</script>`);  
  
  });  
  
});
```

```
app.post("/updateApplication", (req, res) => {  
  
  const id = req.body.id;  
  
  const query = `UPDATE users SET name = 'REDACTED', email =  
'REDACTED', password = 'REDACTED', gender = 'Male', birthdate =  
NULL, country = 'REDACTED', bio = 'REDACTED', additionalInfo =  
'REDACTED' WHERE id = ?`;  
  
  db.query(query, [id], (err, result) => {  
  
    if (err) {
```

```
        return res.send(<script>alert("Database error:
${err.message}");</script>`);
    }

    res.send(<script>alert("Record updated successfully!");</script>`);
});

});
```

```
app.get("/getUserById", (req, res) => {

    const id = req.query.id;

    const query = "SELECT * FROM users WHERE id = ?";

    db.query(query, [id], (err, result) => {

        if (err) {

            return res.status(500).json({ error: "Database error" });

        }

        res.json(result[0]);

    });

});

app.listen(PORT, () => {

    console.log(`Server is running on http://localhost:${PORT}/intro.html`);

});
```

```
function validateForm(formData) {  
  
    const errors = [];  
  
    if (!formData.name) errors.push("Name is required.");  
  
    if (!formData.email) errors.push("Email is required.");  
  
    if (!formData.password) errors.push("Password is required.");  
  
    return errors;  
  
}
```

```
document.getElementById("registrationForm").addEventListener("submit",  
function () {  
  
    alert("Form submitted successfully!");  
  
});
```

```
document.getElementById("uploadFileButton").addEventListener("click",  
function () {  
  
    const fileInput = document.getElementById("file");  
  
    const file = fileInput.files[0];  
  
    if (!file) {  
  
        alert("Please select a file to upload.");  
  
        return;  
  
    }  
  
}
```

```
const formData = new FormData();  
  
formData.append("file", file);  
  
fetch("/upload", {  
  
    method: "POST",  
  
    body: formData  
})  
  
    .then(response => response.json())  
  
    .then(data => {  
  
        if (data.error) {  
  
            alert(data.error);  
  
        } else {  
  
            alert(data.message);  
  
        }  
  
    })  
  
    .catch(error => {  
  
        console.error("Error:", error);  
  
        alert("File upload failed.");  
  
    });  
  
});
```

```
document.addEventListener("DOMContentLoaded", () => {

    const name = getCookie("name");

    const backgroundColor = getCookie("backgroundColor");

    if (name) {

        document.getElementById("greeting").innerText = `Welcome,
${name}!`;

    }

    if (backgroundColor) {

        document.body.style.backgroundColor = backgroundColor;

    }

});
```

```
function getCookie(name) {

    const value = `; ${document.cookie}`;

    const parts = value.split(`; ${name}=`);

    if (parts.length === 2) return parts.pop().split(';').shift();

}
```

```
<!DOCTYPE html>
```


<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>All Applications</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container" style="max-width: 80vw; width: 80vw">

<h1>All Applications</h1>

<table>

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Gender</th>

<th>Birthdate</th>

<th>Country</th>

```

        <th>Actions</th>

    </tr>

</thead>

<tbody>

    {{#each users}}

    <tr>

        <td>{{id}}</td>

        <td>{{name}}</td>

        <td>{{email}}</td>

        <td>{{gender}}</td>

        <td>{{birthdate}}</td>

        <td>{{country}}</td>

        <td>

            <a href="/findApplication?id={{id}}">View</a>

            <form action="/deleteApplication" method="POST"
style="display:inline;">

                <input type="hidden" name="id" value="{{id}}">

                <input type="hidden" name="email" value="{{email}}">

                <button type="submit">Delete</button>

            </form>

            <form action="/updateApplication" method="POST"
style="display:inline;">

```

```
<input type="hidden" name="id" value="{{id}}">

<button type="submit">Update</button>

</form>

</td>

</tr>

{{/each}}

</tbody>

</table>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Delete Application</title>
```

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h1>Delete Application</h1>

<p>This page is under construction.</p>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Find Application</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h1>Find Application</h1>

<form id="searchForm" action="/searchApplication" method="GET">

<label for="searchParam">Enter ID or Email:</label>

**<input type="text" id="searchParam" name="searchParam"
required>**

<button type="submit">Search</button>

</form>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Web Form</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h1>Registration Form</h1>

**<form id="registrationForm" action="/submit" method="POST"
enctype="multipart/form-data">**

<label for="name">Name:</label>

**<input type="text" id="name" name="name" required>
**

<label for="email">Email:</label>

**<input type="email" id="email" name="email" required>
**

<label for="password">Password:</label>

**<input type="password" id="password" name="password"
required>
**

<label for="gender">Gender:</label>

<select id="gender" name="gender">

<option value="male">Male</option>

<option value="female">Female</option>

<option value="other">Other</option>

**</select>
**

<label for="birthdate">Birthdate:</label>

**<input type="date" id="birthdate" name="birthdate">
**

<label for="country">Country:</label>

**<input type="text" id="country" name="country">
**

<label for="bio">Bio:</label>

**<textarea id="bio" name="bio"></textarea>
**

<label for="newsletter">Subscribe to newsletter:</label>

**<input type="checkbox" id="newsletter" name="newsletter">
**

<label for="profilePicture">Profile Picture:</label>

**<input type="file" id="profilePicture" name="profilePicture">
**

<label for="file">Upload File (.7z only):</label>

**<input type="file" id="file" name="file" accept=".7z">
**

<label for="additionalInfo">Additional Info:</label>

```
<textarea id="additionalInfo"
name="additionalInfo"></textarea><br>
```

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

```
</form>
```

```
</div>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Intro Page</title>
```

```
<link rel="stylesheet" href="styles.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```



```
<h1>Welcome</h1>
```

```
<form id="introForm" action="/intro" method="POST">
```

```
  <label for="name">Name:</label>
```

```
  <input type="text" id="name" name="name" required><br>
```

```
  <label for="backgroundColor">Background Color:</label>
```

```
  <input type="color" id="backgroundColor"
name="backgroundColor" required><br>
```

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

```
</form>
```

```
</div>
```

```
<script src="introScript.js"></script>
```

```
</body>
```

```
</html>
```

```
document.addEventListener("DOMContentLoaded", () => {
```

```
  const nameInput = document.getElementById("name");
```

```
  const backgroundColorInput =
document.getElementById("backgroundColor");
```

```
  const name = getCookie("name");
```

```
const backgroundColor = getCookie("backgroundColor");
```

```
if (name) {
```

```
    nameInput.value = name;
```

```
}
```

```
if (backgroundColor) {
```

```
    backgroundColorInput.value = backgroundColor;
```

```
    document.body.style.backgroundColor = backgroundColor;
```

```
}
```

```
document.getElementById("introForm").addEventListener("submit",  
function (event) {
```

```
    event.preventDefault();
```

```
    document.cookie = `name=${nameInput.value}; path=/`;
```

```
    document.cookie = `backgroundColor=${backgroundColorInput.value};  
path=/`;
```

```
    window.location.href = "/step1";
```

```
});
```

```
});
```

```
function getCookie(name) {
```

```
const value = `; ${document.cookie}`;

const parts = value.split(`; ${name}=`);

if (parts.length === 2) return parts.pop().split(';').shift();
}

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Main Page</title>

  <link rel="stylesheet" href="styles.css">

</head>

<body>

  <div class="container">

    <h1>Main Page</h1>

    <button onclick="findApplication()">Найти заявку</button>

    <button onclick="allApplications()">Все заявки</button>

    <button onclick="deleteApplication()">Удалить заявку</button>

  </div>
```

<script>

function findApplication() {

window.location.href = "findApplication";

}

function allApplications() {

window.location.href = "allApplications";

}

function deleteApplication() {

window.location.href = "deleteApplication";

}

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Result Page</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

  <div class="container">

    <h1>Registration Successful</h1>

    <p><strong>Name:</strong> <span id="name"></span></p>

    <p><strong>Email:</strong> <span id="email"></span></p>

    <p><strong>Gender:</strong> <span id="gender"></span></p>

    <p><strong>Birthdate:</strong> <span id="birthdate"></span></p>

    <p><strong>Country:</strong> <span id="country"></span></p>

    <p><strong>Bio:</strong> <span id="bio"></span></p>

    <p><strong>Newsletter:</strong> <span id="newsletter"></span></p>

    <img id="subscriptionImage" src="" alt="Subscription Status">

    <button id="openFileButton">Open File</button>

  </div>

  <script src="resultScript.js"></script>

</body>
```

</html>