

Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

ЛАБОРАТОРНАЯ РАБОТА № 6

по дисциплине

«Эргономические веб-технологии»

на тему «Обработка HTML форм»

Выполнил:

студенты группы

910101

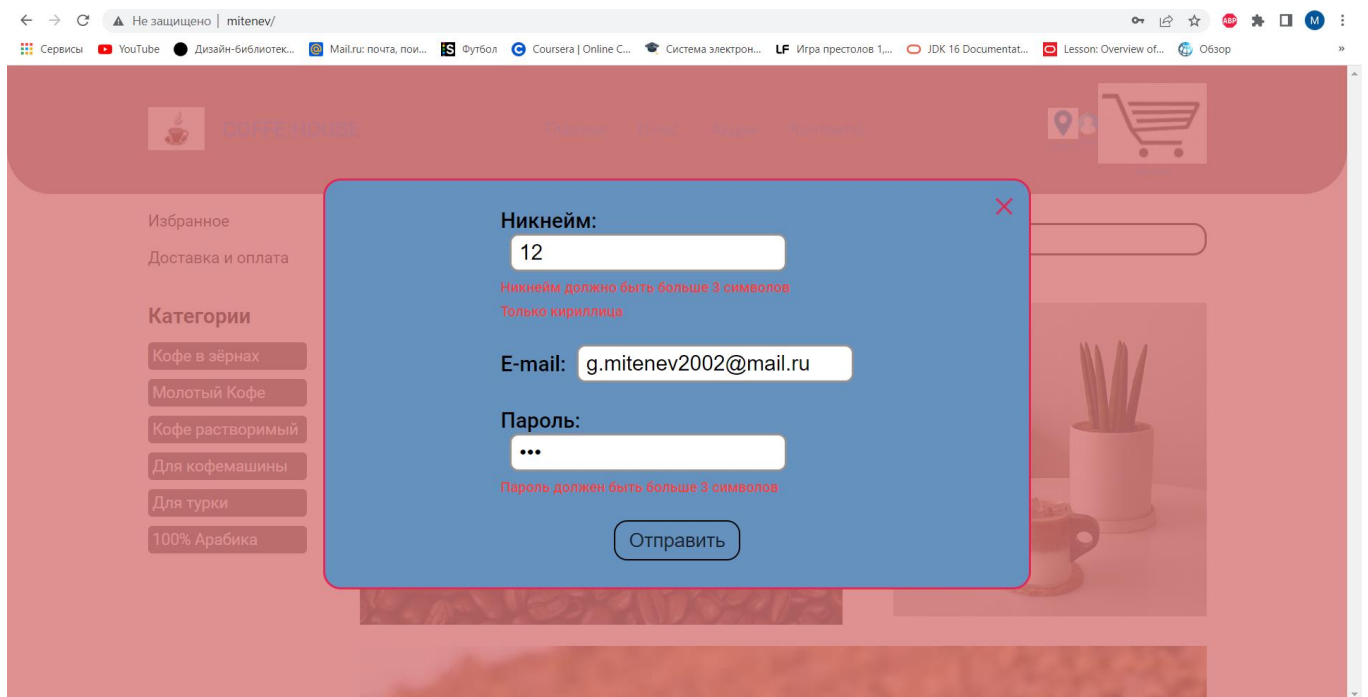
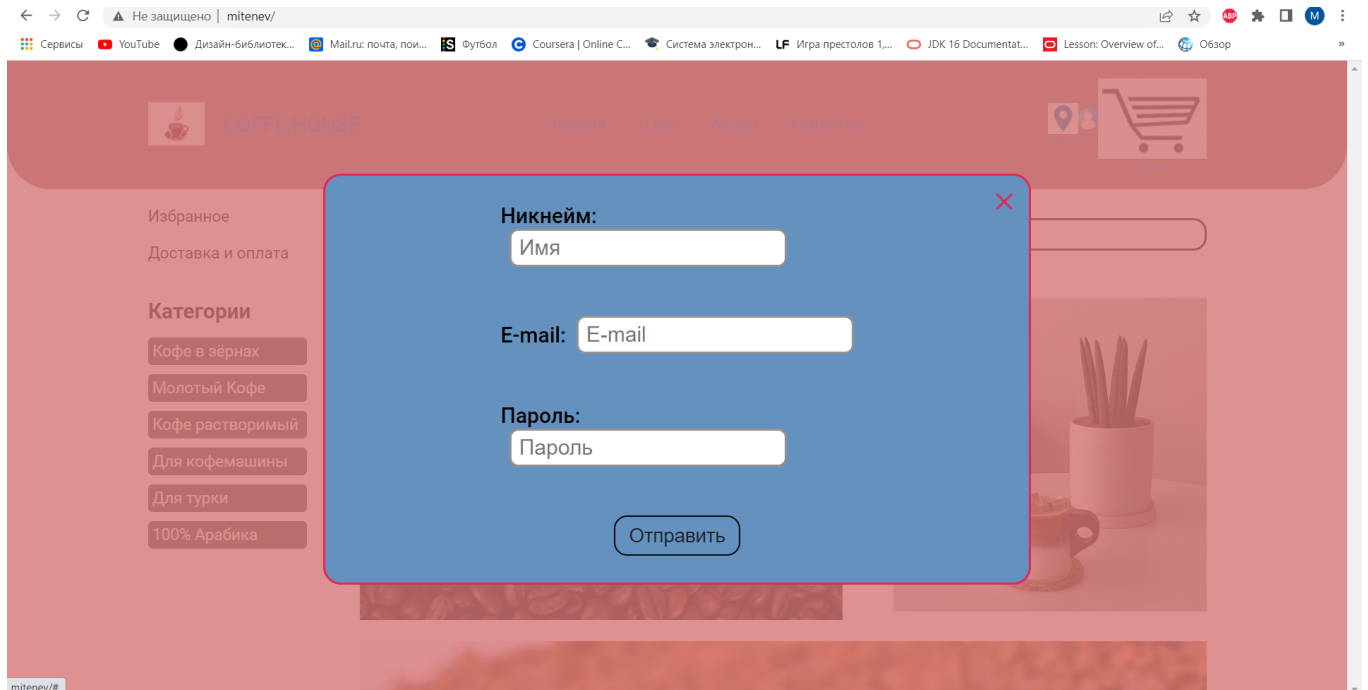
Митенёв Г.Г.

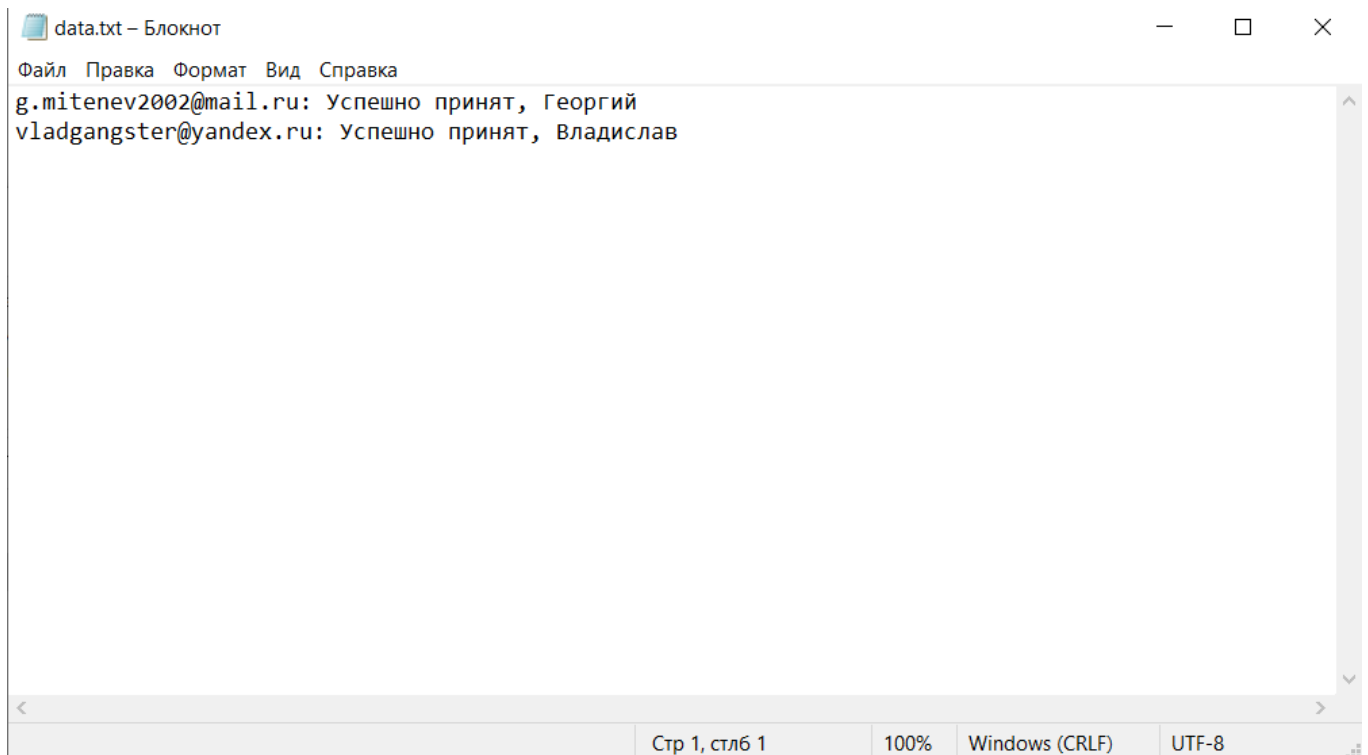
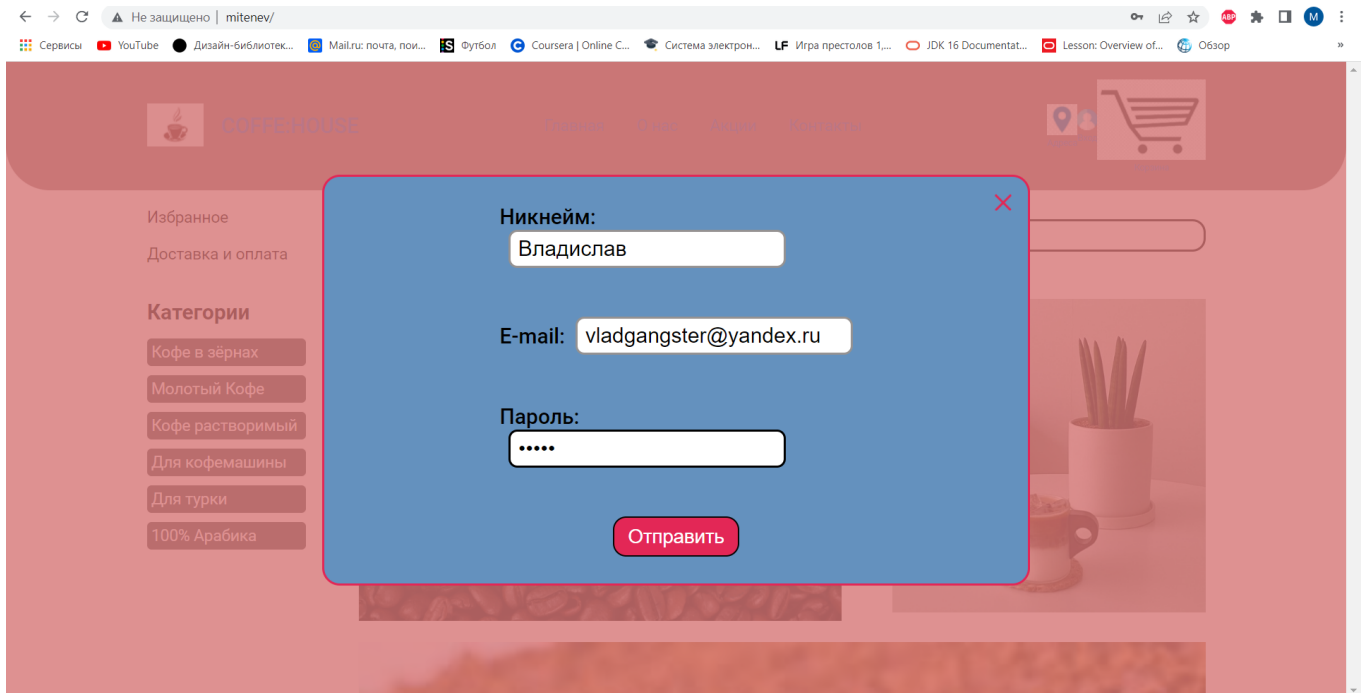
Сенкевич В.В.

Проверил: Кабариха В.А.

Минск БГУИР 2022

Результаты работы:





HTML разметка pop-up form:

```
<div class="pop-up pop-up--hidden pop-up__registration">
  <div class="pop-up__body">
    <div class="pop-up__content">
      <div class="pop-up__cross-wrapper">
        <div class="pop-up__cross">
          <div class="pop-up__cross-inner">
            <div class="cross__string cross__left"></div>
            <div class="cross__string cross__right"></div>
```

```
        </div>
    </div>
</div>
<form class="pop-up__form">
    <label class="pop-up_input pop-up_input--name" name="name">Имя: <input
value="" placeholder="Имя" /></label>
```

```

        <label class="pop-up__input pop-up__input--email" name="email">E-mail: <input
value="" placeholder="E-mail" /></label>
        <label class="pop-up__input pop-up__input--password" name="password">Пароль:
<input value="" type="password" placeholder="Пароль" /></label>
        <button class="pop-up__form-btn">Отправить</button>
    </form>
</div>
</div>
</div>

```

CSS pop-up form:

```

.pop-up {
    display: block;

    position: fixed;
    top: 0%;
    left: 0%;
    z-index: 3;

    width: 100%;
    height: 100%;

    background-color: rgba(77, 72, 72, 0.8);

    overflow-y: scroll;
    visibility: visible;
    opacity: 1;

    transition: linear 0.25s;
}

.pop-up.pop-up--hidden {
    visibility: hidden;
    opacity: 0;
}

.pop-up__body {
    display: flex;
    justify-content: center;
    align-items: center;

    min-height: 100%;
    padding: 50px 30px;
}

.pop-up__content {
    width: 55%;
    padding: 30px;

    border: 3px solid #e32756;
}

```

```
        border-radius: 20px;
        background-color: white;
    }

    .pop-up__cross-wrapper {
        position: relative;
        width: 100%;
    }

    .pop-up__cross {
        position: absolute;
        top: -15px;
        right: -15px;
        z-index: 5;

        width: 25px;
        height: 25px;

        cursor: pointer;
    }

    .pop-up__cross-inner {
        position: relative;
        width: 100%;
        height: 100%;
    }

    .cross__string {
        position: absolute;
        top: 50%;
        left: 50%;

        width: 100%;
        height: 2px;

        border-radius: 3px;
        background-color: #e32756;
    }

    .cross__left {
        transform: translateX(-50%) rotate(-45deg);
    }

    .cross__right {
        transform: translateX(-50%) rotate(45deg);
    }

    .pop-up__form {
        display: flex;
        flex-direction: column;
        justify-content: space-between;
        align-items: center;
        width: 500px;
        height: 350px;
    }
```

```
    margin: 0 auto;
}

.pop-up__input {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
    align-items: center;

    width: 100%;
    color: black;
    font-size: 24px;
    font-weight: 500;
}

.pop-up__input > input {
    margin-left: 10px;
    padding: 5px 8px;

    color: black;
    font-size: 24px;
    font-weight: 500;

    border: 3px solid rgb(149, 146, 146);
    border-radius: 10px;

    transition: .25s linear;
    outline: none;
}

.pop-up__input > input:focus {
    border: 3px solid black;
}

.pop-up__form-btn {
    padding: 8px 15px;

    color: black;
    font-size: 22px;
    font-weight: 500;

    background: transparent;

    border: 0;
    border: 2px solid #e32756;
    border-radius: 15px;

    transition: .25s linear;
    cursor: pointer;
    outline: none;
}

.pop-up__form-btn:hover {
```

```

    color: white;
    background-color: #e32756;
}

.pop-up__form-error {
    color: #F04040;
    font-size: 16px;
    font-weight: 500;

    margin-top: 8px;
    transition: .25s linear;
}

```

JS index.js:

```

//DOM-elements for registration pop-up
const popUp = document.querySelector(".pop-up__registration");
const popUpCross = document.querySelector(".pop-up__cross");
const submitBtn = document.querySelector(".pop-up__form-btn");
const registerBtn = document.querySelector(".header__info-item--register");

const nameInput = document.querySelector(".pop-up__input--name > input");
const emailInput = document.querySelector(".pop-up__input--email > input");
const passwordInput = document.querySelector(".pop-up__input--password > input");

let errors = {
    name: [],
    email: [],
    password: []
};

//refreshes popUp
function renewPopUp() {
    errors = { name: [], email: [], password: [] }
    for(let error of document.querySelectorAll('.pop-up__form-error')) error.remove();
    nameInput.value = ''
    emailInput.value = ''
    passwordInput.value = ''
}

//event handler for showing pop-up form
function showPopUpForm(event) {
    event.preventDefault();

    document.body.classList.add("lock");
    popUp.classList.remove("pop-up--hidden");
}

//event handler that hides pop-up form
function hidePopUpForm(event) {
    if(event.target.classList.contains('pop-up__body') ||
        event.target.closest('.pop-up__cross')) {
        popUp.classList.add("pop-up--hidden");
        setTimeout(() => {

```



```

        document.body.classList.remove("lock")
        renewPopUp()
    }, 250);
}
}

//event handler for sumbiting form
function submitHandler(event) {
    event.preventDefault();

    const data = {
        name: nameInput.value,
        email: emailInput.value,
        password: passwordInput.value
    }

    errors = {
        name: [],
        email: [],
        password: []
    }

    errors = { ...validateForm({ ...data }) };

    if(canBeSubmitted(errors)) {
        popUp.classList.add("pop-up--hidden");
        setTimeout(async () => {
            document.body.classList.remove("lock")
            renewPopUp()

            try {
                let response = await fetch('http://localhost:3000/register/', {
                    mode: 'no-cors',
                    method: 'POST',
                    headers: {
                        'Accept': 'application/jspn',
                        'Content-Type': 'application/json'
                    },
                    body: JSON.stringify(data)
                });
                console.log(response);
            } catch(e) {
                console.log(e);
            }
        }, 250);
    } else {
        for(let error of document.querySelectorAll('.pop-up__form-error')) error.remove();

        for(let label of document.querySelectorAll('.pop-up__input')) {
            labelName = label.getAttribute('name')
            labelErrors = errors[labelName]

            for(let labelError of labelErrors) {
                error_p = document.createElement('p')

```

```

        error_p.classList.add('pop-up_form-error')
        error_p.textContent = labelError
        label.appendChild(error_p)
    }
}

//form validation
function validateForm({ name, email, password }) {
    new_errors = { name: [], email: [], password: [] };

    if(name.length <= 3) {
        new_errors.name.push("Никнейм должно быть больше 3 символов");
    }

    if (!/^[a-яА-Я]*$/i.test(name)) {
        new_errors.name.push("Только кириллица");
    }

    if(!/^([a-z0-9_-]+\.)*[a-z0-9_-]+@[a-z0-9_-]+(\.[a-z0-9_-]+)*\.[a-z]{2,6}$/i.test(email)) {
        new_errors.email.push("Ошибка в email!");
    }

    if(password.length <= 3) {
        new_errors.password.push("Пароль должен быть больше 3 символов");
    }
}

//defines if form can be submitted
function canBeSubmitted(errorObj) {
    for(let error of Object.values(errorObj)){
        if(error.length != 0) return false;
    }
    return true
}

//event for register btn in header
registerBtn?.addEventListener("click", showPopUpForm);
//event listener for pop-up cross
popUpCross?.addEventListener("click", hidePopUpForm);
//event listener for pop-up
popUp?.addEventListener("click", hidePopUpForm);
//event for submit btn
submitBtn?.addEventListener("click", submitHandler);

```

JS server.js:

```
const http = require('http');
```

```

const fs = require('fs');

http.createServer(async (request, response) => {

  if(request.url === '/') {
    fs.readFile('../index.html', 'utf-8', (e, data) => {
      response.end(data);
    });
  } else if(request.url === '/shirts/') {
    fs.readFile('../src/pages/shirts.html', 'utf-8', (e, data) => {
      response.end(data);
    });
  } else if(request.url === '/register/') {
    const buffer = [];

    for await (const chunk of request) {
      buffer.push(chunk);
    }

    const data = Buffer.concat(buffer).toString();
    const dataObj = JSON.parse(data);
    const stringToWriteInFile = `${dataObj?.email} -- Приветствую тебя,
    ${dataObj?.name}\n`;

    if(!fs.existsSync('data.txt')) {
      fs.writeFile('data.txt', stringToWriteInFile, (e) => {
        if(e) throw e;
      });
    } else {
      fs.appendFile('data.txt', stringToWriteInFile, (e) => {
        if(e) throw e;
      });
    }

  } else if (request.url.split('/').length > 2) {
    path = request.url.split('/')[1] === 'src' ? `../${request.url}` :
    `../src${request.url}`
    fs.readFile(path, (e, data) => {
      response.end(data);
    });
  } else {
    response.writeHead(404, { 'Content-Type': 'text/plain' });
    response.end("Page Not Found!");
  }

}).listen(3000, () => console.log("Сервер запущен по адресу http://localhost:3000"));

```