**0МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №6

по дисциплине: **«Современные технологии *Front-end* разработки»**

на тему: **«**Использование технологии *AJAX* для взаимодействия с серверной частью *web*-приложения**»**

Выполнил: студент гр. ИТП-31

Сетко А.И.

Принял: преподаватель-стажер

Васюкова В.О.

Гомель 2020

**Цель**: Научиться использовать метод *fetch* для отправки и получения данных от *web-*сервера.

**Ход работы**

# **Задание:**

Реализовать *CRUD* операции с использованием технологии *ajax* для произвольной предметной области. Реализовать сервер с помощью веб-фреймворка Express для приложений *node.js*. Хранение данных организовать с помощью базы данных либо на стороне сервера.

**Верификация программы:**

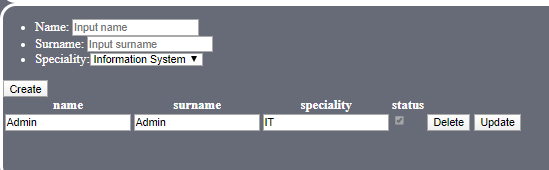


Рисунок 1.1 – Главное окно программы с таблицей пользователей

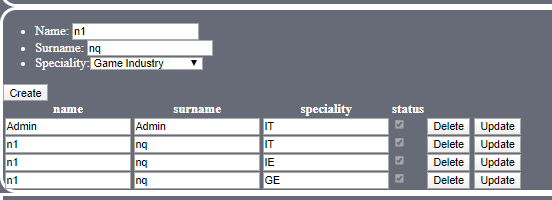


Рисунок 1.2 – Добавление нового пользователя и вывод в общий список

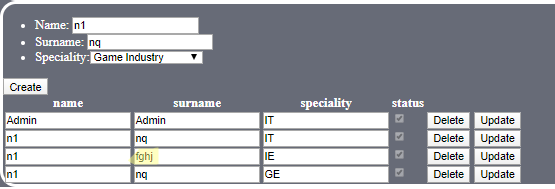


Рисунок 1.3 – Редактирование данных пользователя

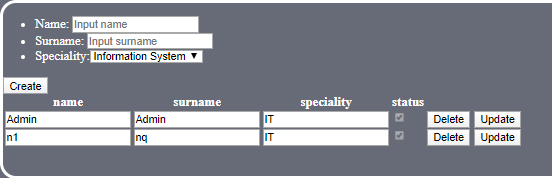


Рисунок 1.3 – Удаление пользователя из системы

**Вывод:** в ходе выполнения лабораторной работы был изучен фреймворк *express* в *node.js*. Данный фреймворк позволяет расширить стандартный функционал сервера, предоставляя разработчику возможность обработки поступающих от клиента запросов. Для отправки запросов использовался метод *fetch*, который работает по принципу *promis*-ов.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Листинг программы**

**Lab6.html**

<**div id="update-block"**>  
 <**form class="contact\_form"**>  
 <**ul**>  
 <**li**>  
 <**label for="text-name"**>Name: </**label**><**input type="text" id="text-name" placeholder="Input name"**>  
 </**li**>  
 <**li**>  
 <**label for="text-surname"**>Surname: </**label**><**input type="text" id="text-surname" placeholder="Input surname"**>  
 </**li**>  
 <**li**>  
 <**label**>Speciality:<**select id="specialities"**>  
 <**option value="IT" selected="selected"**>Information System</**option**>  
 <**option value="IE"**>Industry Electronic</**option**>  
 <**option value="GE"**>Game Industry</**option**>  
  
 </**select**></**label**>  
 </**li**>  
 </**ul**>  
 </**form**>  
</**div**>  
<**input type="button" value="Create" onclick="*controller***.create()**"**>  
<**div id="result-block"**>  
  
</**div**>  
<**div id="students-block"**>  
  
</**div**>  
<**script**>***gstu***.selectStudents()</**script**>

**Lab6.js**

**const *gstu*** = { *//хранит методы* selectStudents: **function** () {  
 *fetch*(**'/user/select'**, {  
 **method**: **'POST'**,  
 **headers**: {  
 **'Content-Type'**: **'application/json'** },  
 **mode**: **'cors'**,  
 **cache**: **'default'** }).then(**function** (resp) {  
 **return** resp.json();  
 }).then(objects => {  
 **let** o = ***JSON***.parse(objects);  
 **let** stb = ***document***.getElementById(**"students-block"**);  
 stb.appendChild(*generateTableStructure*(**"students-table"**, o)); *//формирует таблицу на основе полученных данных* })  
 },  
 deleteStudent: **async function**(id) {  
 **const** url = **'/user/del'**;  
 **const** data = { **id**: id };  
  
 **try** {  
 **const** response = **await** *fetch*(url, {  
 **method**: **'POST'**,  
 **body**: ***JSON***.stringify(data),  
 **headers**: {  
 **'Content-Type'**: **'application/json'** }  
 });  
 **return await** response.json();  
 } **catch** (error) {  
 ***console***.error(**'Ошибка:'**, error);  
 }  
 },  
 createStudent: **async function**(data) {  
 **const** url = **'/user/create'**;  
  
 **try** {  
 **const** response = **await** *fetch*(url, {  
 **method**: **'POST'**,  
 **body**: ***JSON***.stringify(data),  
 **headers**: {  
 **'Content-Type'**: **'application/json'** }  
 });  
 **return await** response.json();  
 } **catch** (error) {  
 **console**.error(**'Ошибка:'**, error);  
 }  
 },  
  
 updateStudent: **async function**(data) {  
 **const** url = **'/user/update'**;  
  
 **try** {  
 **const** response = **await** *fetch*(url, {  
 **method**: **'POST'**,  
 **body**: ***JSON***.stringify(data),  
 **headers**: {  
 **'Content-Type'**: **'application/json'** }  
 });  
 **return await** response.json();  
 } **catch** (error) {  
 **console**.error(**'Ошибка:'**, error);  
 }  
 },  
 **specialities**: [**"IT"**, **"IE"**, **"GI"**]  
};  
  
**const *controller*** = {  
 create: **function** () {  
 **let** name = ***document***.getElementById(**"text-name"**).**value**;  
 **let** surname = ***document***.getElementById(**"text-surname"**).**value**;  
 **let** spec = ***document***.getElementById(**"specialities"**).**value**;  
 **let** data = {  
 **name**: name,  
 **surname**: surname,  
 **speciality**: spec  
 };  
 ***gstu***.createStudent(data).then(r => {  
  
 **let** tbody = ***document***.getElementById(**"main-tbody"**);  
 tbody.appendChild(*createRow*(***JSON***.parse(r)));  
 });  
 },  
 update: **function** (index) {  
 **let** row = ***document***.getElementById(**"row-"** + index);  
 **let** data = [index];  
 **for** (**let** cell **of** row.**cells**) {  
 **if** (cell.**children**[0].**type** === **"text"**) {  
 data.push(cell.**children**[0].**value**);  
 }  
 }  
 ***gstu***.updateStudent(data).then(r => **console**.log(**'Успех:'**, ***JSON***.stringify(r)));  
 */\*  
 document.getElementById("text-name").value = elem.name;  
 document.getElementById("text-surname").value = elem.surname;  
 document.getElementById("specialities").value = elem.speciality;\*/* },  
 **delete**: **function** (id) {  
 **if** (id > 0) {  
 ***gstu***.deleteStudent(id).then(r => **console**.log(**'Успех:'**, ***JSON***.stringify(r)));  
 ***gstu***.selectStudents();  
 }  
 }  
};  
  
**function** *createHeader*(tbody, titleCollection) {  
 **let** header = ***document***.createElement(**"tr"**);  
 titleCollection.forEach(title => {  
 **let** th = ***document***.createElement(**"th"**);  
 th.**innerHTML** = title;  
 header.appendChild(th)  
 });  
 tbody.appendChild(header);  
}  
  
**function** *generateTableStructure*(tableId, collection) {  
 *isTableExists*(tableId);  
  
 **let** table = ***document***.createElement(**"table"**);  
 table.**id** = tableId;  
 table.**className** = **"table\_dark"**;  
 **let** tbody = ***document***.createElement(**"tbody"**);  
 tbody.**id** = **"main-tbody"**;  
  
 *createHeader*(tbody, [**"name"**, **"surname"**, **"speciality"**, **"status"**]);  
  
 collection.forEach(f => {  
 tbody.appendChild(*createRow*(f, f.**id**));  
 });  
 table.appendChild(tbody);  
  
 **return** table;  
}  
  
**function** *createRow*(element, index) {  
 **let** row = ***document***.createElement(**"tr"**);  
 row.**id** = **"row-"** + index;  
  
 **for** (**let** i = 0; i < 6; i++) {  
 row.appendChild(*createTd*(element, i, index));  
 }  
 **return** row;  
}  
  
**function** *createTd*(element, col, index) {  
 **let** td = ***document***.createElement(**"td"**);  
  
 **switch** (col) {  
 **case** 0: td.appendChild(*createElement*(**"input"**, **"text"**, element.**name**)); **break**;  
 **case** 1: td.appendChild(*createElement*(**"input"**, **"text"**, element.**surname**)); **break**;  
 **case** 2: td.appendChild(*createElement*(**"input"**, **"text"**, element.**speciality**)); **break**;  
 **case** 3:  
 **let** cb = *createElement*(**"input"**, **"checkbox"**, element.**speciality**);  
 cb.**checked** = element.**isActive**;  
 cb.**disabled** = **true**;  
 td.appendChild(cb); **break**;  
 **case** 4: td.appendChild(*createBtn*(**"Delete"**, ***controller***.delete, index)); **break**;  
 **case** 5: td.appendChild(*createBtn*(**"Update"**, ***controller***.update, index)); **break**;  
 }  
  
 **return** td;  
}  
  
**function** *createElement*(elementName, type, val) {  
 **let** elem = ***document***.createElement(elementName);  
 elem.**value** = val;  
 elem.**type** = type;  
  
 **return** elem;  
}  
  
**function** *createSelectObject*(className, tr) {  
 **let** td = ***document***.createElement(**"td"**);  
 **let** select = ***document***.createElement(**"select"**);  
 select.**id** = className;  
  
 **for** (**let** i = 0; i < ***gstu***.**specialities**.**length**; i++) {  
 select.**options**[i] = **new *Option***(***gstu***.**specialities**[i], ***gstu***.**specialities**[i]);  
 }  
  
 td.appendChild(select);  
 tr.appendChild(td);  
}  
  
**function** *createBtn*(btnText, func, parameters) {  
 **let** btn = ***document***.createElement(**"button"**);  
 btn.**className** = **'btn btn-dark'**;  
 **let** textInBtn = ***document***.createTextNode(btnText);  
 btn.appendChild(textInBtn);  
  
 **if** (func !== **undefined**) {  
 btn.**onclick** = **function** () {  
 func(parameters);  
 };  
 }  
 **return** btn;  
}  
  
**function** *isTableExists*(name) {  
 **let** t;  
 **if** ((t = ***document***.getElementById(name)) !== **null**) {  
 t.remove();  
 }  
}

**Index.js**

**var** express = **require**(**'express'**),  
 app = express();   
app.set(**'port'**, 8080);  
app.use(express.**static**(\_\_dirname));  
app.listen(app.get(**'port'**), **function**() {  
 **console**.log(**'Server started: http://localhost:'** + app.get(**'port'**) + **'/'**);  
});  
**const** jsonParser = express.**json**();  
  
app.get(**"/"**, **function**(request, response){  
 response.sendFile(\_\_dirname + **"/index.html"**);  
});  
  
app.post(**"/user/select"**, jsonParser, **function** (request, response) {  
 **if**(!request.**body**) **return** response.sendStatus(400);  
  
 response.json(***JSON***.stringify(users.filter(f => f.**isActive**))); *// отправляем пришедший ответ обратно*});  
  
app.get(**"/user/select"**, jsonParser, **function** (request, response) {  
 **if**(!request.**body**) **return** response.sendStatus(400);  
  
 response.json(***JSON***.stringify(users.filter(f => f.**isActive**))); *// отправляем пришедший ответ обратно*});  
  
app.post(**"/user/update"**, jsonParser, **function** (request, response) {  
 **if**(!request.**body**) **return** response.sendStatus(400);  
  
 users.forEach(f => {  
 **if** (f.**id** === request.**body**[0]) {  
 f.**name** = request.**body**[1];  
 f.**surname** = request.**body**[2];  
 f.**speciality** = request.**body**[3];  
 }  
 });  
  
 response.json(***JSON***.stringify(**true**)); *// отправляем пришедший ответ обратно*});  
  
app.post(**"/user/del"**, jsonParser, **function** (request, response) {  
 **if**(!request.**body**) **return** response.sendStatus(400);  
 users.forEach(f => {  
 **if** (f.**id** === request.**body**.**id**) f.**isActive** = **false**;  
 });  
 response.json(***JSON***.stringify(**true**)); *// отправляем пришедший ответ обратно*});  
  
app.post(**"/user/create"**, jsonParser, **function** (request, response) {  
 **if**(!request.**body**) **return** response.sendStatus(400);  
 **let** flag = **true**;  
 users.forEach(f => {  
 **if** (f.**name** === request.**body**.**name** &&  
 f.**surname** === request.**body**.**surname** &&  
 f.**speciality** === request.**body**.**speciality**)  
 flag = **false**;  
 });  
 **if** (flag) {  
 **let** user = {  
 **id**: users.**length** + 1,  
 **name**: request.**body**.**name**,  
 **surname**: request.**body**.**surname**,  
 **speciality**: request.**body**.**speciality**,  
 **isActive**: **true** };  
 users.push(user);  
 response.json(***JSON***.stringify(user)); *// отправляем пришедший ответ обратно* }  
});  
  
**let** admin = {  
 **id**: 1,  
 **name**: **"Admin"**,  
 **surname**: **"Admin"**,  
 **speciality**: **"IT"**,  
 **isActive**: **true**};  
  
**let** users = [admin];