**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

по дисциплине «Современные технологии *FRONT*-*END* разработки»

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

Выполнил: студент гр. ИТИ-41   
 Давидовский Д.Е.  
 Принял: преподаватель

Титова Л.К.

Гомель 2022

**Цель работы:** Научиться использовать объекты типа *Promise* для формирования цепи последовательных вызовов функций.

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

Релазивать *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.

Сервер использует технологию *node*.*js* и реализует систему сохранения информации о студентах на собственной стороне. Система реализуется при помощи фреймворка *express*. На рисунке 1.1 показан этап запуска сервера и подгрузки файлов разметки.

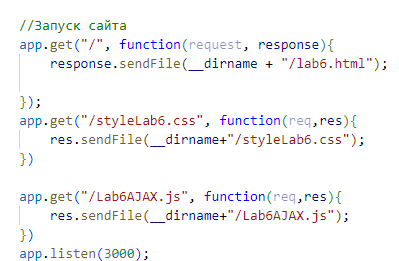


Рисунок 1.1 – Запуск сервера

*CRUD* операции подразумевают систему создания, редактирования, удаления, добавления информации. В данном случае используется информация о студентах. На рисунке 1.2 показан сайт с возможностью добавления и редактирования информации о студентах.

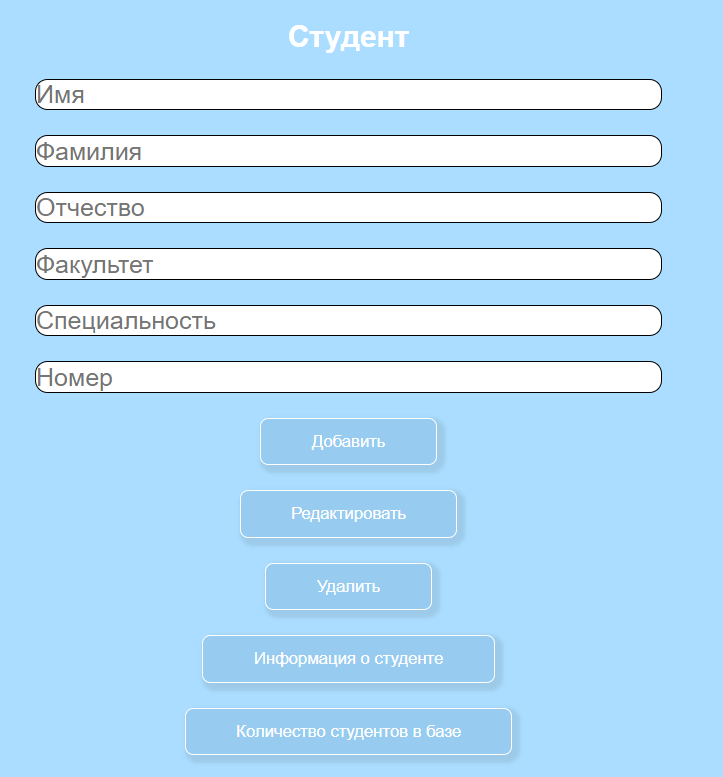


Рисунок 1.2 – Сайт с добавлением информации о студентах

Система хранения данных реализована на стороне сервера. Сервер принимает *html* запросы, в которых в *json* форме отправляется информация о студентах. Метод отправления информации показан на рисунке 1.3.



Рисунок 1.3 – Отправление информации на сервер

Таким образом мы передаем всю информацию о студенте. В случае получения информации используется отдельная функция, отправляющая запрос и считывающая ответ с сервера в *json* форме. Функция считывания ответа с сервера показана на рисунке 1.4.



Рисунок 1.4 – Получение информации с сервера

**Вывод:** В результате работы было реализовано приложение, организующее систему *CRUD* для создания объектов студентов, их добавления в базу, удаления, редактирования.

**ПРИЛОЖЕНИЕ А**

(обязательное)

Листинг программы

Lab6.js

*class* *Student*{

*constructor*(*name*,*surname*,*lastname*,*faculty*,*speciality*,*index*){

*this*.*name* = *name*;

*this*.*surname* = *surname*;

*this*.*lastname* = *lastname*;

*this*.*faculty* = *faculty*;

*this*.*speciality* = *speciality*;

*this*.*index* = *index*

    };

*ConvertToJSON*(){

*let* *jsonElement* = *JSON*.*stringify*(*this*);

*return* *jsonElement*;

    }

*InfoAbout*(){

*let* *str* = "*Student*: "+*this*.*name*+" "+*this*.*surname*+" "+*this*.*lastname*+" *studying* *on* "+*this*.*speciality*+" *from* "+*this*.*faculty*;

*return* *str*;

    }

}

*function* *Redact*(){

*SendStudent*("/*redact*");

}

*function* *SendStudent*(*info*){

      // получаем данные формы

*var* *inputMass* = *document*.*getElementsByClassName*("*inputText*");

*var* *student* = *new* *Student*(*inputMass[*0*]*.*value*,*inputMass[*1*]*.*value*,*inputMass[*2*]*.*value*,*inputMass[*3*]*.*value*,*inputMass[*4*]*.*value*,*inputMass[*5*]*.*value*);

   // сериализуем данные в *json*

*student* = *student*.*ConvertToJSON*();

*let* *request* = *new* *XMLHttpRequest*();

   // посылаем запрос на адрес "/*user*"

*request*.*open*("*POST*", *info*, *true*);

*request*.*setRequestHeader*("*Content*-*Type*", "*application*/*json*");

*request*.*addEventListener*("*load*", *function* () {

       // получаем и парсим ответ сервера

*alert*(*request*.*response*);

    });

*request*.*send*(*student*);

}

*function* *Delete*(){

*SendStudent*("/*delete*");

}

*function* *Add*(){

*SendStudent*("/*user*");

};

*function* *RecieveInfo*(){

      // получаем данные формы

*var* *inputMass* = *document*.*getElementsByClassName*("*inputText*");

*var* *student* = *new* *Student*(*inputMass[*0*]*.*value*,*inputMass[*1*]*.*value*,*inputMass[*2*]*.*value*,*inputMass[*3*]*.*value*,*inputMass[*4*]*.*value*,*inputMass[*5*]*.*value*);

   // сериализуем данные в *json*

*student* = *student*.*ConvertToJSON*();

*let* *request* = *new* *XMLHttpRequest*();

*request*.*open*("*POST*", "/*info*", *true*);

*request*.*setRequestHeader*("*Content*-*Type*", "*application*/*json*");

*request*.*addEventListener*("*load*", *function* () {

*var* *result* = *request*.*response*;

*result* = *JSON*.*parse*(*result*);

*result* = *ConvertFromJson*(*result*);

*inputMass[*0*]*.*value* = *result*.*name*;

*inputMass[*1*]*.*value* = *result*.*surname*;

*inputMass[*2*]*.*value* = *result*.*lastname*;

*inputMass[*3*]*.*value* = *result*.*faculty*;

*inputMass[*4*]*.*value* = *result*.*speciality*;

*inputMass[*5*]*.*value* = *result*.*index*;

    });

*request*.*send*(*student*);

};

*function* *ConvertFromJson*(*JSONobject*){

*var* *student* = *new* *Student*(*JSONobject*.*name*,*JSONobject*.*surname*,*JSONobject*.*lastname*,*JSONobject*.*faculty*,*JSONobject*.*speciality*, *JSONobject*.*index*);

*return* *student*;

}

*function* *AmountOfStudents*(){

*var* *request* = *new* *XMLHttpRequest*();

*request*.*open*("*POST*", "/*amount*",*true*);

*request*.*addEventListener*("*load*", ()=>{

*alert*(*request*.*response*);

    });

*request*.*send*();

}

*var* *Students* = *[]*

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

*const* *app* = *express*();

// создаем парсер для данных в формате *json*

*const* *jsonParser* = *express*.*json*();

*app*.*post*("/*amount*", *function* (*request*, *response*) {

*console*.*log*("*Amount* *send*");

*response*.*json*(*Students*.*length*);

});

*app*.*post*("/*info*",*jsonParser*, *function* (*request*, *response*) {

*console*.*log*("---------*RECIEVED* *FOR* *INFO*----------");

*console*.*log*(*request*.*body*);

*console*.*log*("-------------------");

*if*(!*request*.*body*) *return* *response*.*sendStatus*(400);

*var* *student* = *request*.*body*;

*console*.*log*(*student*);

*response*.*json*(*Students[student*.*index]*);

});

*app*.*post*("/*delete*",*jsonParser*, *function* (*request*, *response*) {

*console*.*log*("---------*RECIEVED* *FOR* *DELETE*----------");

*console*.*log*(*request*.*body*);

*console*.*log*("-------------------");

*if*(!*request*.*body*) *return* *response*.*sendStatus*(400);

*var* *student* = *request*.*body*;

*student*.*index* = *Students*.*length*-1;

*Students*.*splice*(*student*.*index*, *student*.*index*);

*response*.*json*("*Deleted*"); // отправляем пришедший ответ обратно

});

*app*.*post*("/*redact*", *jsonParser*, *function* (*request*, *response*) {

*console*.*log*("---------*RECIEVED* *FOR* *REDACT*----------");

*console*.*log*(*request*.*body*);

*console*.*log*("-------------------");

*if*(!*request*.*body*) *return* *response*.*sendStatus*(400);

*var* *student* = *request*.*body*;

*Students[student*.*index]* = (*student*);

*response*.*json*("*Redacted*"); // отправляем пришедший ответ обратно

});

*app*.*post*("/*user*", *jsonParser*, *function* (*request*, *response*) {

*console*.*log*("---------*RECIEVED* *FOR* *NEW* *USER*----------");

*console*.*log*(*request*.*body*);

*console*.*log*("-------------------");

*if*(!*request*.*body*) *return* *response*.*sendStatus*(400);

*var* *student* = (*request*.*body*);

*student*.*index* = *Students*.*length*;

*Students*.*push*(*student*);

*console*.*log*(*student*);

*response*.*json*("*Added*"); // отправляем пришедший ответ обратно

});

//Запуск сайта

*app*.*get*("/", *function*(*request*, *response*){

*response*.*sendFile*(*\_\_dirname* + "/*lab*6.*html*");

});

*app*.*get*("/*styleLab*6.*css*", *function*(*req*,*res*){

*res*.*sendFile*(*\_\_dirname*+"/*styleLab*6.*css*");

})

*app*.*get*("/*Lab*6*AJAX*.*js*", *function*(*req*,*res*){

*res*.*sendFile*(*\_\_dirname*+"/*Lab*6*AJAX*.*js*");

})

*app*.*listen*(3000);

Lab6.css

\* {

*margin*:0;

*padding*:0;

*font*-*family*: '*Open* *Sans*', *sans*-*serif*;

}

.*header*-*horizontal*{

*position*: *relative*;

*height*: 55*px*;

*background*-*color*: #97*ccf*0;

*box*-*shadow*: 0 0 10*px* *rgba*(39, 39, 39, 0.5);

}

.*Text* *a*{

*display*: *inline*-*block*;

*width*: 100%;

*text*-*align*: *center*;

*font*-*family*: *fantasy*;

*font*-*weight*: 700;

*color*: *white*;

*font*-*size*: 1.5*em*;

*padding*-*top*: 8*px*;

*font*-*family*: '*Nanum* *Gothic*', *sans*-*serif*;

*text*-*decoration*: *none*;

}

.*Text* *a*:*visited*{

*text*-*decoration*: *none*;

*color*: *white*;

}

.*header*-*horizontal* *ul* *li*{

*color*: *white*;

}

*main*{

*background*-*color*: #*aaddff*;

*width*: 100%;

*height*: 900*px*;

*display*:*flex*;

}

.*testzone*{

*display*: *inline*-*block*;

*width*: 100%;

*height*: 100%;

*text*-*align*: *center*;

}

.*testzone* *h*2{

*margin*-*top*: 30*pt*;

*color*: *white*;

}

.*button*{

*margin*-*top*: 15*pt*;

*border*: 1*px* *solid*;

*padding*: 10*px* 40*px*;

*border*-*radius*: 0.5*em*;

*background*-*color*: #97*ccf*0;

*box*-*shadow*: 3*px* 3*px* 3*px* 3*px* *rgba*(39, 39, 39, 0.1);

*color*: *white*;

}

.*inputText*{

*margin*-*top*: 15*pt*;

*border*: 1*px* *solid*;

*border*-*radius*: 0.5*em*;

*font*-*size*: 20*px*;

}

.*inputAnagram*{

*margin*-*top*: 15*pt*;

*border*: 1*px* *solid*;

*border*-*radius*: 0.5*em*;

*font*-*size*: 20*px*;

}

Lab6.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Давидовский Даниил</title>

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <link href="styleLab6.css" rel="stylesheet" type="text/css">

    <title>Document</title>

    <script src="Lab6AJAX.js"></script>

</head>

<body>

    <header>

        <nav class="header-horizontal">

            <ul>

                <li class="Text"><a href="D:\\Учеба\\FrontEnd\\Project\\index.html">Лабораторные работы</a></li>

            </ul>

        </nav>

    </header>

    <main>

        <form name="registerForm"   class="testzone" id="test">

            <h2>Студент</h2>

            <p class="TaskInfo">Релазивать CRUD операции с использованием технологии ajax для произвольной предметной области.<br>

                 Реализовать сервер с помощью веб-фреймворка Express для приложений node.js.<br>

                  Хранение данных организовать с помощью базы данных либо на стороне сервера.</p>

            <div><input type="text" name = "name" maxlength="200" class="inputText" placeholder="Имя"></div>

            <div><input type="text" name = "surname" maxlength="200" class="inputText" placeholder="Фамилия"></div>

            <div><input type="text" name = "lastname" maxlength="200" class="inputText" placeholder="Отчество"></div>

            <div><input type="text" name = "faculty" maxlength="200" class="inputText" placeholder="Факультет"></div>

            <div><input type="text" name = "speciality" maxlength="200" class="inputText" placeholder="Специальность"></div>

            <div><input type="text" name = "speciality" maxlength="200" class="inputText" id="StudentId" placeholder="Номер"></div>

            <p><input type="button" class = "button" onclick="Add()" value="Добавить"></p>

            <p><input type="button" class = "button" onclick="Redact()" value="Редактировать"></p>

            <p><input type="button" class = "button" onclick="Delete()" value="Удалить"></p>

            <p><input type="button" class = "button" onclick="RecieveInfo()" value="Информация о студентe"></p>

            <p><input type="button" class = "button" onclick="AmountOfStudents()" value="Количество студентов в базе"></p>

            <div class="peopleArray"></div>

        </form>

    </main>

</body>

</html>