Министерство транспорта Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Отчёт

по практике

«Учебная практика»

­

Работа с файлами с помощью JavaScript

Выполнил: ст. гр. ТУУ-211

Дидушок Д.А.

Проверил: доц. Сафронов А.И.

Москва ­– 2019 г.

Цель работы:

Изучать принцип работы с файлами с помощью JavaScript

**Задание:**

Разработать интерфейс, содержащий кнопки загрузки файлов, скачивания файла, изменения его расширения и добавление файлов в архив и скачивание его.

**Блок-схема:**

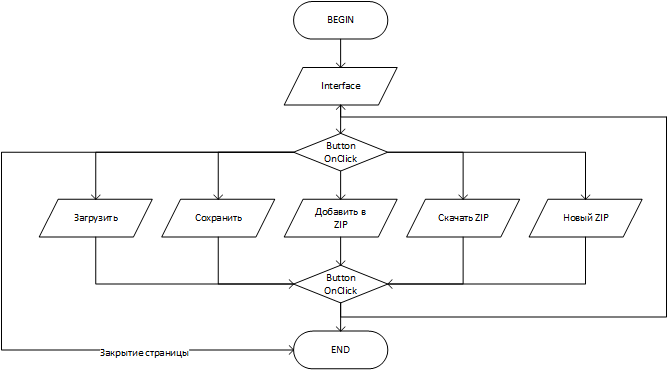
****

Рисунок 1. Идея взаимодействия

Выполнение работы:

* **Код:**

**HTML**

index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<link rel="icon" href="image/favicon.ico" type="image/x-icon">

<link rel="stylesheet" href="style/main.css">

<title>Работа с файлами</title>

</head>

<body>

<script src="scripts/Vue.js"></script>

<script src="scripts/jszip.js"></script>

<script src="scripts/first\_page.js"></script>

<script type="text/x-template" id="firstPage">

<div id="vueFile">

<div id="app">

<div id="main">

Введите имя файла:

<div id="fileName" contenteditable></div>

Введите расширение файла:

<div id="fileExpansion" contenteditable></div>

<br>

<input id="fileInput" type="file" @change="readfile($event)" multiple>

<br>

<button id="select" @click="readerButton()">Загрузить</button>

<button id="download" @click="writefile()">Сохранить</button>

<p>

<button id="add" @click="addZip()">Добавить в Zip</button>

<button id="create" @click="createZip()">Скачать Zip</button>

<button id="clear" @click="clearZip()">Новый Zip</button></p>

</div>

<div id="result">Файлы в архиве:</div>

<div id="fileOutput" contenteditable>Введите текст</div>

</div>

</div>

</script>

<div id="vueFile">

<component v-bind:is="getPage"></component>

</div>

<script>

var myVueFileReader = new Vue(

{

el: '#vueFile',

data:

{

zip: '',

selPage: 'firstPage'

},

computed:

{

getPage: function()

{

zip = new JSZip();

if (this.selPage == 'firstPage')

{

return 'firstPage';

}

else

{

if (this.selPage == 'secondPage')

{

return 'secondPage';

}

else

{

if (this.selPage == 'direct')

{

return 'direct';

}

else

{

if (this.selPage == 'archiving')

{

return 'archiving';

}

}

}

}

}

}

})

</script>

</body>

</html> <div id="myCatalog">

<b>Выберите элемент из списка</b>

<br>

<select v-model="selected">

<option v-for="el in this.$root.fileData" v-bind:value="el.description.effect"> {{el.productname}} </option>

</select>

<span v-if="selected!==''"> <b>Описание:</b> <i>{{selected}}</i> </span>

<br>

<button @click="back">Назад</button>

</div>

</script>

<div id="myCatalog">

<component v-bind:is="getPage"></component>

</div>

<script>

myVueCatalog = new Vue(

{

el: '#myCatalog',

data:

{

fileData: '',

pageN: 'fSelect'

},

computed:

{

getPage: function()

{

if (this.pageN == 'fSelect')

{

return 'selectf';

}

else

{

if (this.pageN == 'NForm')

{

return 'noform';

}

else

{

if (this.pageN == 'Form')

{

return 'format';

}

else

{

if (this.pageN == 'Spisok')

{

return 'nspis';

}

}

}

}

}

},

methods:

{

},

})

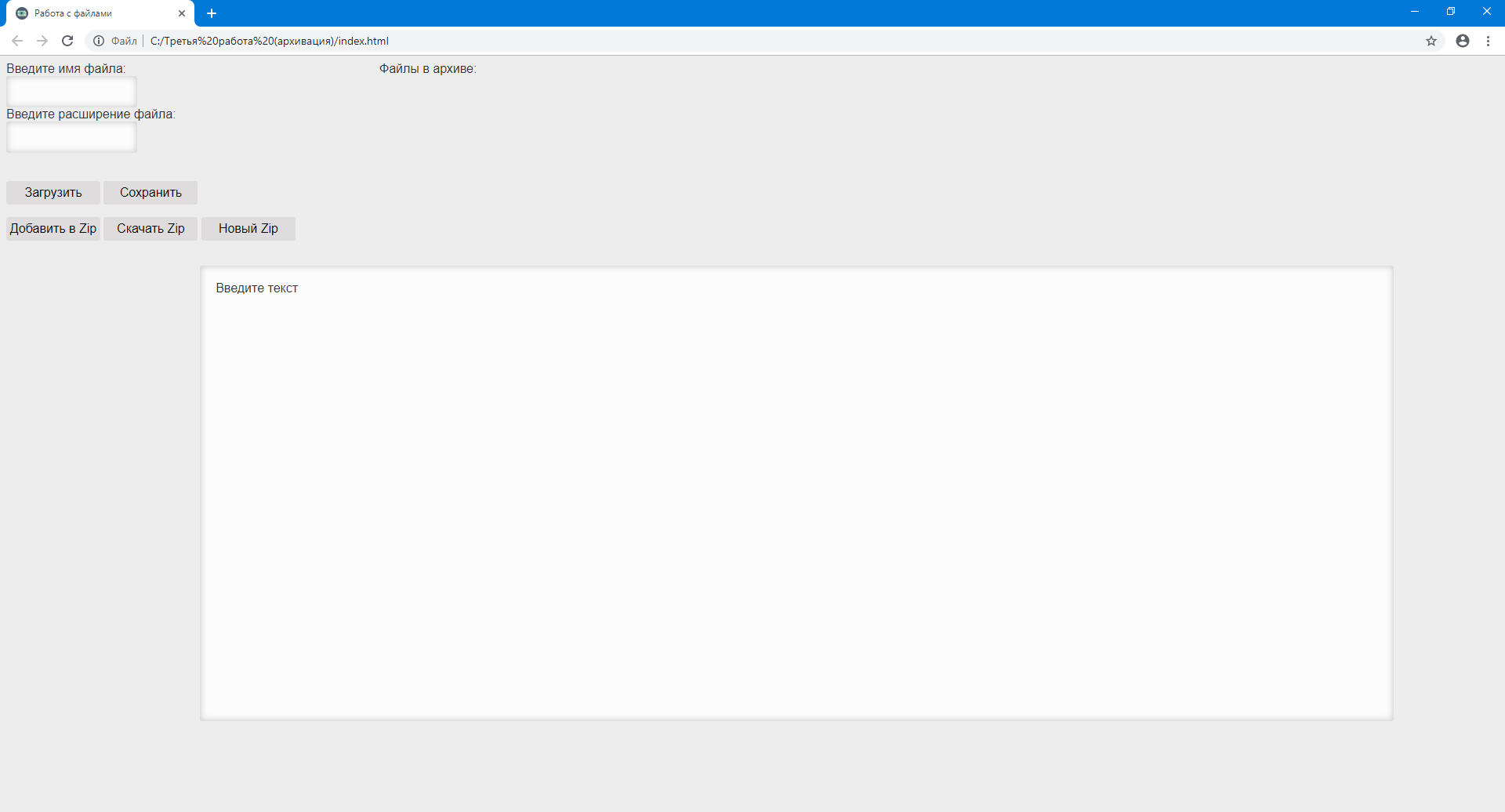
</script>

</body>

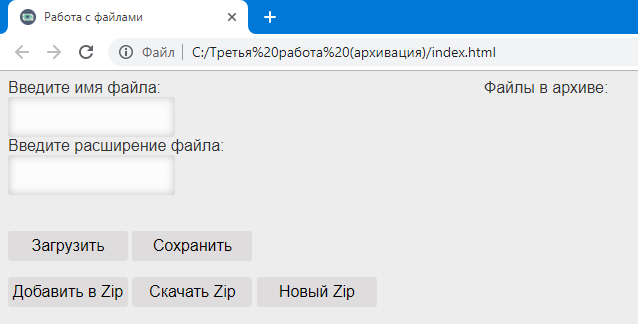
</html>

|  |
| --- |
| **JS** |
| first\_page.js |
| Vue.component('firstPage',  {  template: '#firstPage',  methods:  {  writefile: function ()  {  let content = document.getElementById('fileOutput').innerText;  let name = document.getElementById('fileName').innerText;  let expansion = document.getElementById('fileExpansion').innerText;  let blob = new Blob([content]);  let url = URL.createObjectURL(blob);  function download(url, name){  let a = document.createElement("a");  a.download = name;  a.href = url;  document.body.appendChild(a);  a.click();  a.remove();  }  download(url, name + '.' + expansion);  URL.revokeObjectURL(url);  },  readfile: function (e)  {  let file = e.target.ownerDocument.getElementById('fileInput').files[0];  tmp = file.name.split('.');  name = tmp[0];  expansion = tmp[1];  document.getElementById('fileName').innerText = name;  document.getElementById('fileExpansion').innerText = expansion;  let reader = new FileReader();  reader.onload = function (en) {  let output = document.getElementById("fileOutput");  output.textContent = en.target.result;  };  reader.readAsText(file);  },  readerButton: function()  {  let fileInput = document.getElementById('fileInput');  fileInput.click();  },  createZip: function()  {  zip.generateAsync({type:"base64"})  .then(function (content) {  location.href="data:application/zip;base64,"+content;  });  },  addZip: function()  {  let content = document.getElementById('fileOutput').innerText;  let name = document.getElementById('fileName').innerText;  let expansion = document.getElementById('fileExpansion').innerText;  zip.file(name + '.' + expansion, content);  let list = document.getElementById('result');  let block = document.createElement('div');  block.id = 'resultBlock';  if (document.getElementById('resultBlock') != null)  {  block = document.getElementById('resultBlock');  let li = document.createElement('li');  li.innerHTML = name + '.' + expansion;  block.appendChild(li);  }  else  {  list.appendChild(block);  block = document.getElementById('resultBlock');  let li = document.createElement('li');  li.innerHTML = name + '.' + expansion;  block.appendChild(li);  }  },  clearZip: function()  {  zip = new JSZip();  let list = document.getElementById('resultBlock').remove();  },  next: function()  {  myVueFileReader.selPage = 'direct';  }  }  }) |
| **CSS** |
| main.css |
| body {  font-family: Arial, Verdana, sans-serif; /\* Семейство шрифтов \*/  font-size: 12pt; /\* Размер основного шрифта в пунктах \*/  background-color: #ededed; /\* Цвет фона веб-страницы \*/  color: #333; /\* Цвет основного текста \*/  }  button{  width: 120px;  height: 30px;  border-radius: 3px;  background: #dedcdc;  border: none;  box-sizing: border-box;  padding: 0;  font-size: 16px;  color: #000;  cursor: pointer;  outline: none;  }  button:hover {  border: 1px solid #000;  color: #000;  }  #fileOutput {  position: relative;  top: 260px;  right: 800px;  width: 80%;  height: 60vh;  margin: 0 auto;  padding: 20px;  background: #fcfcfc;  border-radius: 3px;  box-shadow: inset 0 0 8px 1px rgba(0,0,0,.2);  box-sizing: border-box;  overflow: hidden;  word-break: break-all;  outline: none;  resize: none;  flex: none;  }  #fileName {  position: relative;  width: 35%;  height: 0;  padding: 20px;  background: #fcfcfc;  border-radius: 3px;  box-shadow: inset 0 0 8px 1px rgba(0,0,0,.2);  box-sizing: border-box;  overflow: hidden;  word-break: break-all;  outline: none;  resize: none;  }  #fileExpansion {  position: relative;  width: 35%;  height: 0;  padding: 20px;  background: #fcfcfc;  border-radius: 3px;  box-shadow: inset 0 0 8px 1px rgba(0,0,0,.2);  box-sizing: border-box;  overflow: hidden;  word-break: break-all;  outline: none;  resize: none;  }  #fileInput {  display: none;  }  #app {  display: flex;  }  #main {  flex: none;  width: 25%;  }  #result {  flex: none;  width: 30%;  }  #next {  position: relative;  top: 30px;  }  #back {  position: relative;  top: 30px;  } |
|  |

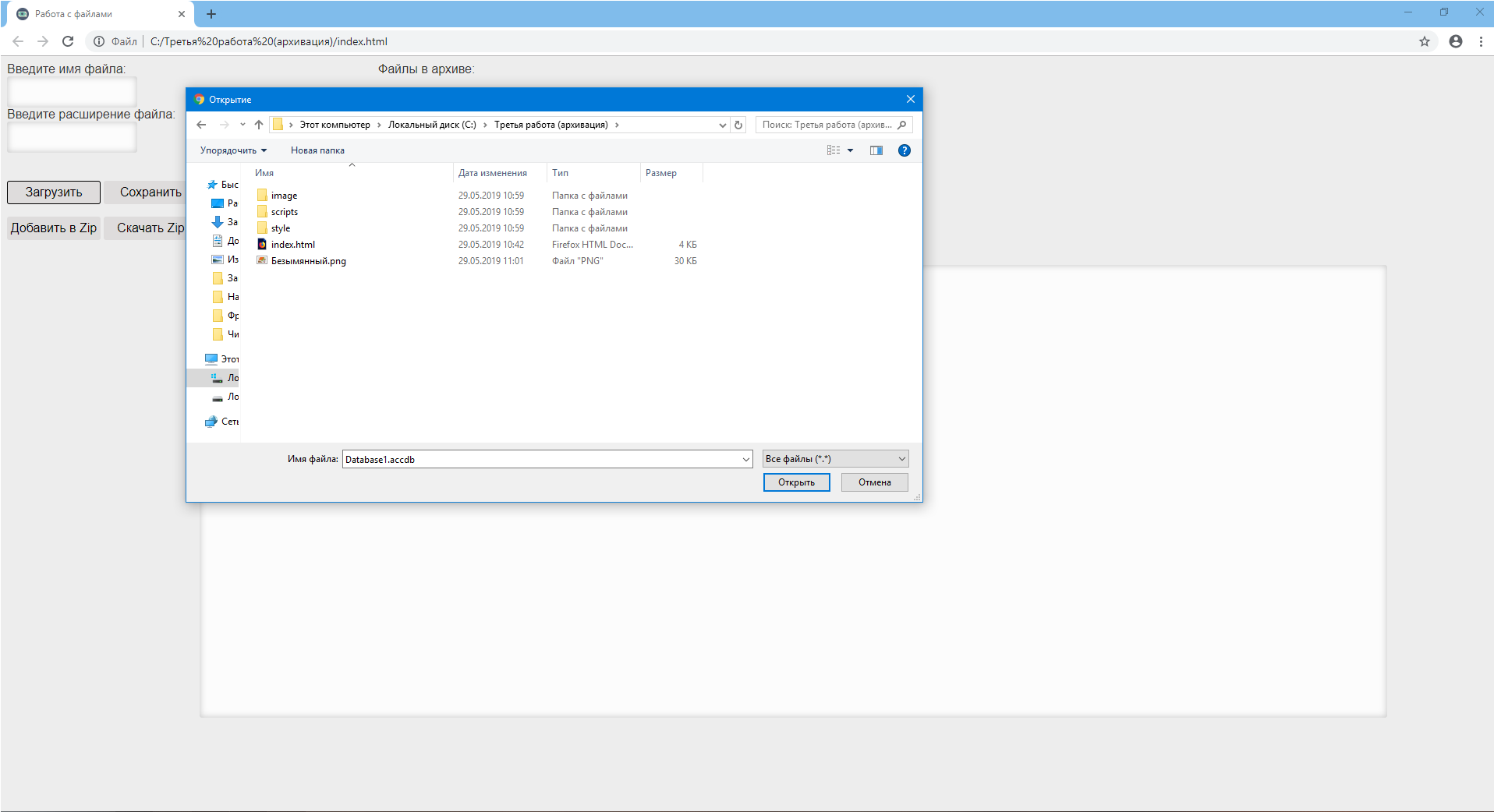
**Результат:**

****

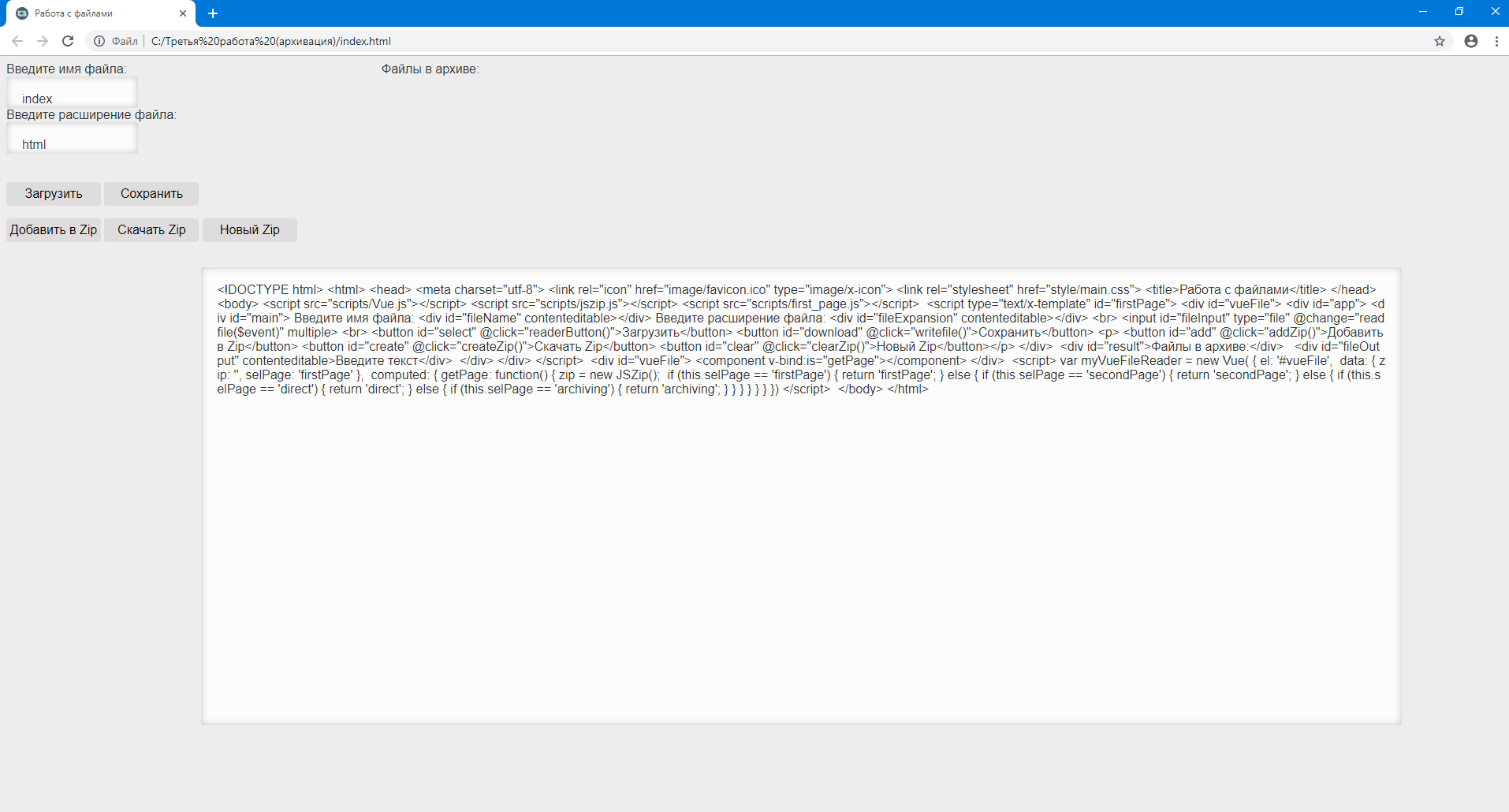
**Скриншот 1.Начальное окно**



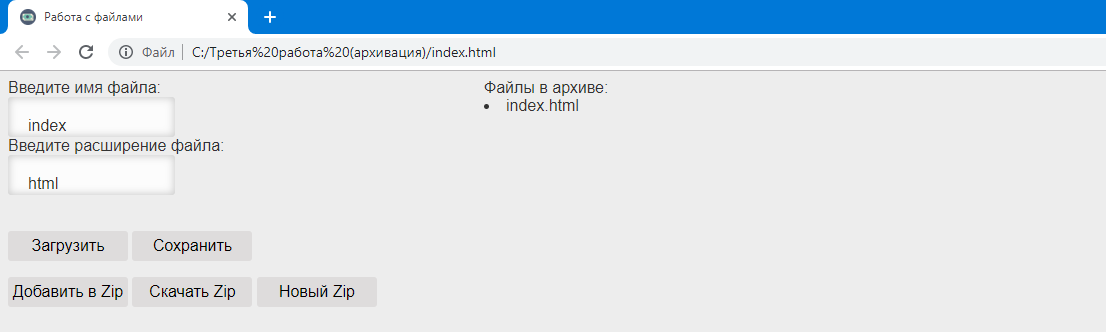
**Скриншот 2. Панель инструментов**

****

**Скриншот 3. Загрузка файла.**

****

Скриншот 4. Отображение данных из файла



Скриншот 5. Отображение файлов в архиве

Вывод:

Изучены принципы работы c файлами с помощью JavaScript.