Минобрнауки России

федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный технологический институт (технический университет)»

СПбГТИ(ТУ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Направление подготовки | | 09.03.01 | Информатика и вычислительная техника | | | |
| Направленность программы | |  | Автоматизированные системы обработки информации и управления  Очная | | | |
|  | |  |  | | | |
| Факультет | |  | Информационных технологий и управления | | | |
| Кафедра | |  | Систем автоматизированного  проектирования и управления | | | |
| Учебная дисциплина | |  | Интернет-технологии | | | |
|  |  | | |  |  |

Курс 3 Группа 404

**Отчет по контрольной работе № 2**

|  |  |
| --- | --- |
| **Тема:** | **Создание интерфейса веб-сайта**  **с использованием HTML, CSS и JS** |

Студент гр. 404 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. К. Азаров

(подпись, дата) (инициалы, фамилия)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Иванов

(подпись, дата) (инициалы, фамилия)

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись преподавателя)

Санкт-Петербург

2023

#### **1 Описание идеи проекта. Формирование функциональных требований к проекту**

В данной контрольной работе осуществляется продолжение развитие идеи проекта, описанной в контрольной работе №1.

**Цель проекта:** разработать веб-приложение для шифрования.

**Функциональные требования к разрабатываемому веб-приложению:**

1) Возможность переключения между алгоритмами.

2) Возможность генерации ключа.

3) Возможность отправки данных на сервер.

#### **2 Обзор и обоснование выбора технологий реализаций**

Для реализации бизнес-логики в клиентской части разрабатываемого веб-приложения будет использоваться нативный JavaScript.

**Ниже представлены популярные фреймворки JS:**

***React JS***

React - это JavaScript-библиотека с открытым исходным кодом, которая широко применяется при разработке пользовательских интерфейсов. Библиотека хорошо принимается UX-дизайнерами, SEO специалистами и имеет положительную репутацию среди разработчиков.

***Angular***

Фреймворк был впервые представлен компанией Google в 2012 году и долгое время находился на первом месте по популярности, однако в последнее время потерял лидирующие позиции.

***Vue JS***

Vue - это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментами и дополнительными библиотеками.

***Next JS***

Открытый JavaScript фреймворк, созданный поверх React.js для создания веб-приложений, созданный компанией Vercel. Фреймворк был предназначен для решения проблемы React.js, связанную с отрисовкой приложения на стороне сервера - SSR. Работает на сервере и в браузере.

***Ember JS***

Ember JS - это свободный каркас веб-приложений на JavaScript, реализующий шаблон MVC и нацеленный на упрощение создания масштабируемых одностраничных веб-приложений.

***JQuery***

jQuery - это набор функций JavaScript, фокусирующийся на взаимодействии JS и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX. Разработка jQuery ведётся командой добровольцев на пожертвования [[1].](#_СПИСОК_ЛИТЕРАТУРЫ)

Таким образом, существует множество библиотек, из которых в первую очередь была выбрана JQuery, так как она:

* решает много проблем с кросс-браузером JS;
* выполняет сложные операции при добавлении небольшого количества кода;
* легко добавить функционал AJAX;
* есть встроенные библиотеки UI и эффектов.

#### **3 Разработка UML-диаграммы вариантов использования программы**

На рисунке 1 представлена UML-диаграммы вариантов использования веб-приложения для пользователя.

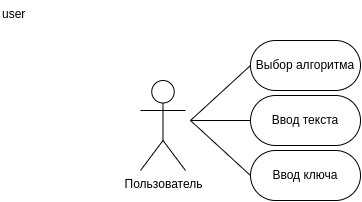


Рисунок 1 – UML-диаграммы вариантов использования веб-приложения

#### **4 Разработка блок-схемы алгоритма решения задачи**

На рисунке 2 представлены блок-схемы алгоритмов решения задач.



Рисунок 2 – Блок-схемы алгоритма решения задачи ввода отзыва

#### **5 Реализация с использованием JS проекта**

С помощью JS были реализованы следующие подзадачи:

1. Переключение алгоритмов.
2. Генерация ключа.
3. Валидация ключа.
4. Отправка данных на сервер.
5. Получение данных с сервера и вывод из пользователю.

Ниже представлена реализация этих задач.

window.onload = init

function init()

{

// Изменение шифра

var topMenuChilds = document.getElementById("topmenu").children

for (let el of topMenuChilds)

{

el.addEventListener("click", \_ => changeCipherType(el.textContent))

}

// Генерация ключа

let generateKeyBtn = document.getElementById("generate\_key")

generateKeyBtn.addEventListener("click", setGenerateKeyOnTexBox)

// let keyTextBox = document.getElementById("key\_box")

// keyTextBox.setCustomValidity("Что-то не правильно")

// Кодирование

let cipherBtn = document.getElementById("ciph\_btn")

cipherBtn.addEventListener("click", \_ => ciphText(false))

let decipherBtn = document.getElementById("deciph\_btn")

decipherBtn.addEventListener("click", \_ => ciphText(true))

}

function changeCipherType(name)

{

let header = document.getElementById("cipher\_name")

header.textContent = name

let shownAlghoritm = document.getElementsByClassName("show")[0]

shownAlghoritm.className = "hide"

let alghoritmManual = document.getElementById(name + "-alghoritm")

alghoritmManual.className = "show"

}

function setGenerateKeyOnTexBox()

{

let keyTextBox = document.getElementById("key\_box")

keyTextBox.value = generateKey()

}

function generateKey()

{

let shownAlghoritm = document.getElementById("cipher\_name").textContent

let count = getAlgoKeyLenght(shownAlghoritm)

let min = Math.pow(10, 16);

let max = Math.pow(10, 17)

let val = "";

for(let i = 0; i < 10; ++i)

{

val += Math.floor(Math.random() \* (max - min) + min).toString(36)

}

return val.slice(-count)

}

function getAlgoKeyLenght(name)

{

let count = 0;

switch (name)

{

case "AES":

count = 128 //128/192/256 бит

break

case "BLOWFISH":

count = 32 //от 32 до 448 бит

break

case "CAST":

count = 64 //40-128 бит

break

case "DES":

count = 64 //56 бит + 8 проверочных бит

break

}

let v = count / 8

return count/8

}

function ciphText(isDecipher)

{

console.log(isDecipher)

let keyText = document.getElementById("key\_box")

let typeCiph = document.getElementById("cipher\_name").textContent

if (!checkKeyValidity(keyText.value, typeCiph))

{

keyText.setCustomValidity(`Ключ должен содержать ${getAlgoKeyLenght(typeCiph)} символов`);

keyText.reportValidity()

return

}

let inText = document.getElementById("in\_text").value

let outText = document.getElementById("out\_text")

let req =

{

isDeciph: isDecipher,

type: typeCiph,

text: inText,

key: keyText.value

}

fetch("/",

{

method: 'POST',

headers: { 'Content-Type': 'application/json;charset=utf-8' },

body: JSON.stringify(req)

})

.then(res => res.text())

.then(text => outText.textContent = text)

}

function checkKeyValidity(key, ciphType)

{

return key.length == getAlgoKeyLenght(ciphType)

}

# **СПИСОК ЛИТЕРАТУРЫ**

* 1. Learn Javascript: Популярные JavaScript фреймворки и библиотеки в 2023 году: сайт. – Москва, 2023 –.– URL: [https://learn-javascript.ru/article/top-javascript-framework](https://learn-javascript.ru/article/top-javascript-framework%20) (Дата обращения: 29.03.2023).
  2. JQuery: JQuery Api Documentation: сайт. –.– URL: <https://api.jquery.com/>.