

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Алтайский государственный технический университет им. И. И. Ползунова»  
Университетский технологический колледж

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_ *С. В. Умбетов*

«\_\_\_\_\_» \_\_\_\_\_ 2025 г.

Отчёт по практической работе №1  
по теме:  
«ROT13 Шифратор/Дешифратор.»

Студент группы 1ИСП-21  
группа

С.Б. Рем  
и.о., фамилия

Преподаватель ассистент, к. т. н.  
должность, ученая степень

С. В. Умбетов  
и.о., фамилия

БАРНАУЛ 2025

## Практическая работа №1

### Задания к лабораторной работе:

#### Функционал

1. **HTML5** страница + **JS** с реализацией ROT13.
2. **Шифровка** (для чётных вариантов) / **Дешифровка** (для нечётных).
3. **ROT13** – сдвиг латинских букв на **13 позиций**, остальные символы без изменений.

#### Ограничения

- Запрещены: `replace`, `charCodeAt`, `fromCharCode`, `join`, `split`.
- **Алфавит** – только массив (`['a','b',...,'z']`).
- **Вывод алфавита, операций и результата** на страницу.

#### Интерфейс

- **Левая часть** – исходный текст.
- **Правая часть** – зашифрованный/расшифрованный текст.

#### Технические требования

- Валидный **HTML5**.
- Обработка регистра ( $A \rightarrow N, a \rightarrow n$ ).
- **Без использования** стандартных строковых методов.

Ы

Задание принял:



Рем С.Б.

## Ход работы

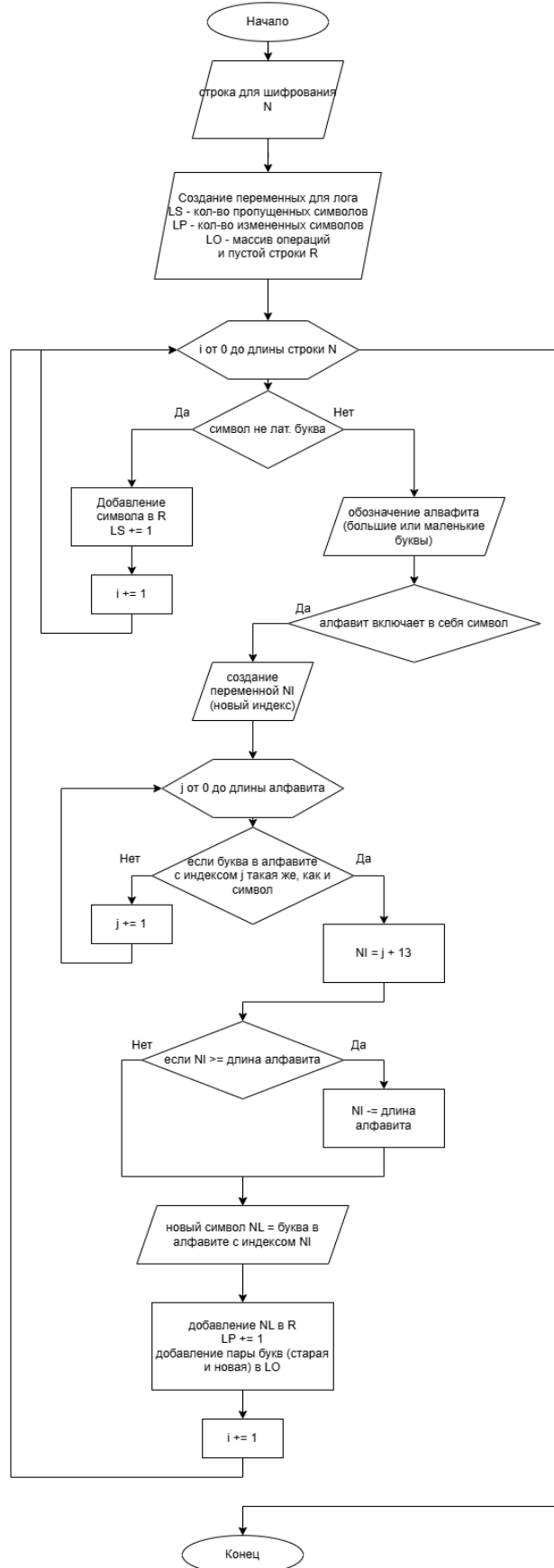


Рисунок 1 – Блок схема

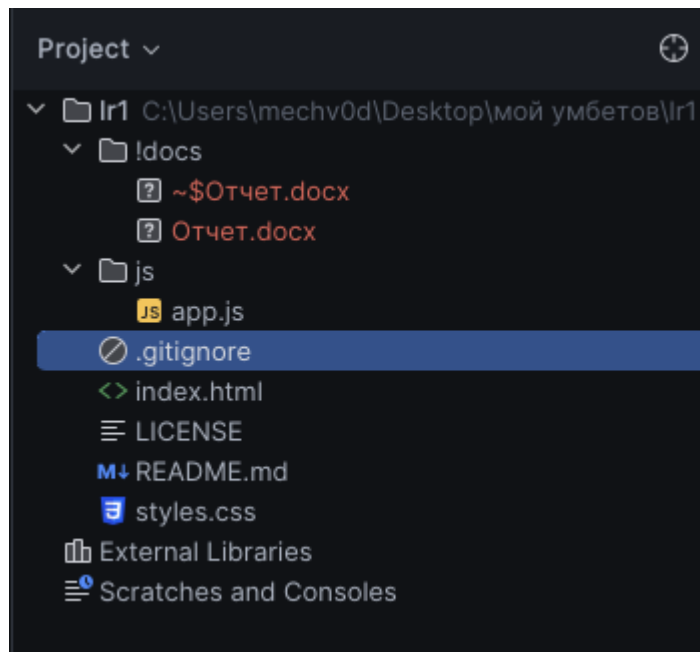


Рисунок 2 – Иерархия проекта

```
<> index.html x <> 404.html error.css styles.css app.js color.css
1 <!DOCTYPE html>
2
3 <html lang="ru">
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport"
7         content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
8     <meta http-equiv="X-UA-Compatible" content="ie=edge">
9
10    <link rel="preconnect" href="https://fonts.googleapis.com">
11    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
12    <link href="https://fonts.googleapis.com/css2?family=Victor+Mono:ital,wght@0,100..700;1,100..700&display=swap"
13        rel="stylesheet">
14
15    <link rel="stylesheet" href="css/color.css">
16    <link rel="stylesheet" href="css/styles.css">
17
18    <script src="js/app.js"></script>
19
20    <title>ДЕ/ШИФРАТОР</title>
21
22    <!-- this work belongs to the student of IISP-21 Rem Sergey.-->
23    <!-- All rights reserved.-->
24    <!-- Don't eat yellow snow!-->
25 </head>
26 <body>
27 <div id="#page-wrapper" class="page-wrapper">
28     <h1>ДЕ<span class="pink"></span>ШИФРАТОР</h1>
29     <h2>Формат: ROT13</h2>
30 <div>
31     <section id="forms" class="forms-container">
32     <div class="circles">
33         <svg class="circle" xmlns="http://www.w3.org/2000/svg">
34             <circle cx="6" cy="6" r="6"/>
35         </svg>
36         <svg class="circle yellow" xmlns="http://www.w3.org/2000/svg">
37             <circle cx="6" cy="6" r="6"/>
38         </svg>
39         <svg class="circle green" xmlns="http://www.w3.org/2000/svg">
40             <circle cx="6" cy="6" r="6"/>
41         </svg>
42
43         <svg class="circle black" xmlns="http://www.w3.org/2000/svg">
44             <a href="https://t.me/mechv0d" target="_blank" rel="noopener noreferrer">
45                 <circle cx="6" cy="6" r="6"/>
46             </a>
47         </svg>
48     </div>
49     <div class="line">
50     <div class="encrypt-div">
51     <label for="user_text">Текст для шифровки</label>
```

Рисунок 3 – HTML код (часть 1)

```
<> index.html x <> 404.html error.css styles.css app.js color.css
3 <html lang="ru">
26 <body>
27 <div id="#page-wrapper" class="page-wrapper">
30 <div>
31 <section id="forms" class="forms-container">
48
49 </div>
50 <div class="line">
51 <div class="encrypt-div">
52 <label for="user_text">Текст для шифровки</label>
53 <textarea id="user_text" class="main-text-area" placeholder="Введите текст..."></textarea>
54 <button type="submit" onclick="submitButtonLogic(this)">Зашифровать
55 </button>
56 </div>
57 <div id="output-div" class="encrypt-div" style="background-color: #f0f0f0">
58 <label>Готовый шифр:</label>
59 <p id="output-text" class="main-text-area"></p>
60 <button onclick="copyButtonLogic(this)">Скопировать
61 </button>
62 </div>
63 </div>
64 <div id="operations-info" class="line alphabet-info">
65 <p class="alphabet">Алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z</p>
66 <div class="op-count">
67 <p id="op-all">Операций:</p>
68 <p id="op-proceed">Выполнено:</p>
69 <p id="op-skip">Пропущено:</p>
70 </div>
71 <div id="op-list" class="op-list">
72 </div>
73
74 </div>
75 </section>
76 </div>
77
78 <section class="info-container">
79 
80 <p class="description">ROT13 – это простой шифр подстановки, который заменяет каждую букву алфавита на 13-ю по
81 счёту после неё:<br>(A →
82 N, B → O, ..., M → Z, N → A и т. д.). <br><br>Он работает только с латинскими буквами, а цифры и символы
83 оставляет
84 без изменений. <br><br>Особенность ROT13 в том, что это обратимый шифр: повторное применение ROT13
85 возвращает
86 исходный текст. Изначально он использовался в интернете для сокрытия спойлеров и шуток, а не для реальной
87 защиты данных.</p>
88 </section>
89 </div>
90 <footer>
91 <p class="year">mecyv0d · 2025</p>
92 </footer>
93 </body>
94 </html>
```

Рисунок 4 – HTML код (часть 2)

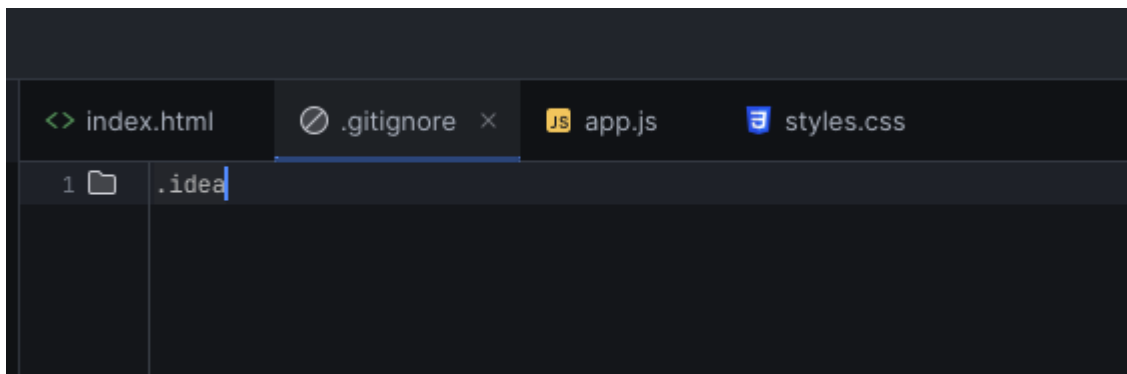


Рисунок 5 – Файл «.gitignore»

```

1  const alphabetUpper : string[] = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'];
2  const alphabetLower : string[] = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'];
3  const rot13_range : number = 13;
4
5  const isLetter = (str) => alphabetUpper.includes(str) || alphabetLower.includes(str); Show usages  mechv0d
6
7  function encrypt_ROT13(text) : {...} { Show usages  mechv0d *
8      let l_proceed : number = 0;
9      let l_skipped : number = 0;
10     let l_ops : any[] = [];
11     let newText : string = '';
12
13     for (let i : number = 0; i < text.length; i++) {
14         let letter = text[i];
15
16         if (!isLetter(letter)) {
17             newText += letter;
18             l_skipped++;
19         } else {
20             let targetArray : string[] = alphabetLower.includes(letter) ? alphabetLower : alphabetUpper;
21
22             if (targetArray.includes(letter)) {
23                 let newIndex : number = 0;
24                 for (let j : number = 0; j < targetArray.length; j++) {
25                     if (targetArray[j] === letter) {
26                         newIndex = j + rot13_range;
27                         break;
28                     }
29                 }
30
31                 if (newIndex >= targetArray.length) newIndex -= targetArray.length;
32                 let newLetter : string = targetArray[newIndex];
33                 newText += newLetter;
34
35                 l_proceed++;
36                 l_ops.push([letter, newLetter])
37             }
38         }
39     }
40
41     return {
42         textEncrypted: newText,
43         opAll: l_proceed + l_skipped,
44         opProceed: l_proceed,
45         opSkip: l_skipped,
46         opList: l_ops
47     };
48 }
49
50 function outputEncrypted(data) : void { Show usages  mechv0d
51     let opInfoContainer : Element = document.querySelector(selectors: '#operations-info');
52     encrypt_ROT13()

```

Рисунок 6 – JS код (часть 1)

```
<> index.html <> 404.html error.css styles.css app.js x color.css
function encrypt_ROT13(text) : {...} { Show usages mechv0d *
49 }
50
51 function outputEncrypted(data) : void { Show usages mechv0d
52   let opInfoContainer : Element = document.querySelector( selectors: '#operations-info');
53   let outputDiv : Element = document.querySelector( selectors: '#output-div');
54   let outputText : Element = document.querySelector( selectors: '#output-text');
55   let opAll : Element = document.querySelector( selectors: '#op-all');
56   let opProceed : Element = document.querySelector( selectors: '#op-proceed');
57   let opSkip : Element = document.querySelector( selectors: '#op-skip');
58   let opList : Element = document.querySelector( selectors: '#op-list');
59
60   if (data === null) {
61     outputDiv.style.visibility = 'collapse';
62     opInfoContainer.style.display = 'none';
63     return;
64   }
65
66   outputDiv.style.visibility = 'visible';
67   opInfoContainer.style.display = 'flex';
68
69   outputText.innerText = data.textEncrypted;
70   opAll.innerText = 'Операций: ' + data.opAll
71   opProceed.innerText = 'Выполнено: ' + data.opProceed
72   opSkip.innerText = 'Пропущено: ' + data.opSkip
73   opList.innerText = ''
74
75   for (const opPair of data.opList) {
76     opList.innerHTML += `<p class="op-list-el">${opPair[0]} -> ${opPair[1]}</p>`;
77   }
78   // opList.innerHTML = opList.innerHTML.slice(0, -1)+'.';
79 }
80
81 function submitButtonLogic() : void { Show usages mechv0d
82   let textArea : Element = document.querySelector( selectors: '#user_text');
83
84   if (textArea.value !== '') {
85     let data : {...} = encrypt_ROT13(textArea.value);
86     outputEncrypted(data)
87
88     // outputText.innerText = data.textEncrypted;
89     if (window.screen.width <= 450) {
90       const element : Element = document.querySelector( selectors: '#forms');
91       const topPosition : number = element.getBoundingClientRect().top + window.scrollY - 32;
92       window.scrollTo( options: {
93         top: topPosition,
94         behavior: 'smooth'
95       });
96     }
97   }
98 }
99
100 function copyButtonLogic(el) : void { Show usages mechv0d
101   const readyText : Element = document.querySelector( selectors: '#output-text');
102   navigator.clipboard.writeText(readyText.innerText).then(() : void => {
103     el.innerHTML = 'Скопировано, Босс!';
104     setTimeout( handler: () : string => el.innerHTML = 'Скопировать', timeout: 1666);
105   });
106 }
```

Рисунок 7 – JS код (часть 2)



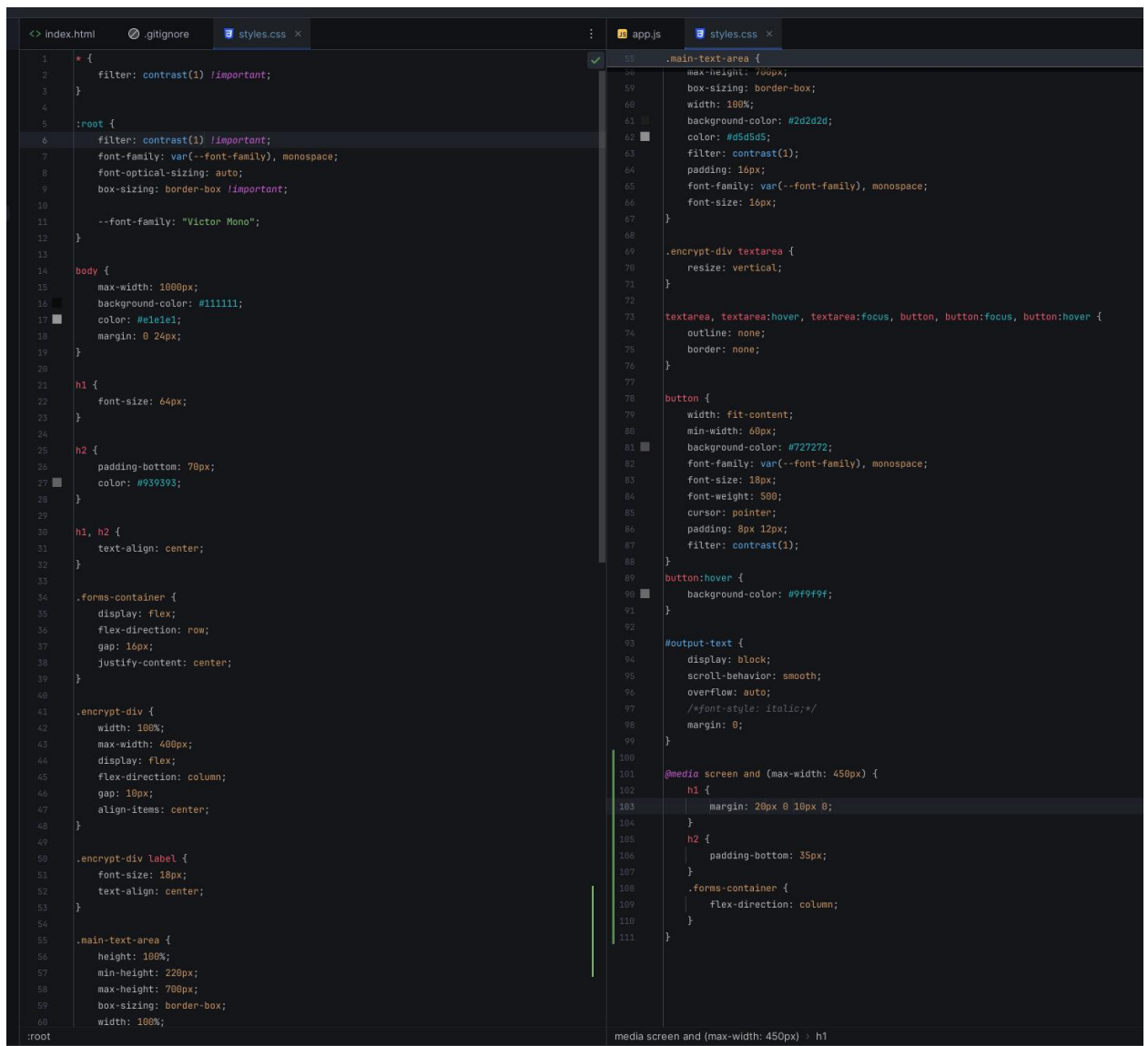


Рисунок 8 – Стили CSS

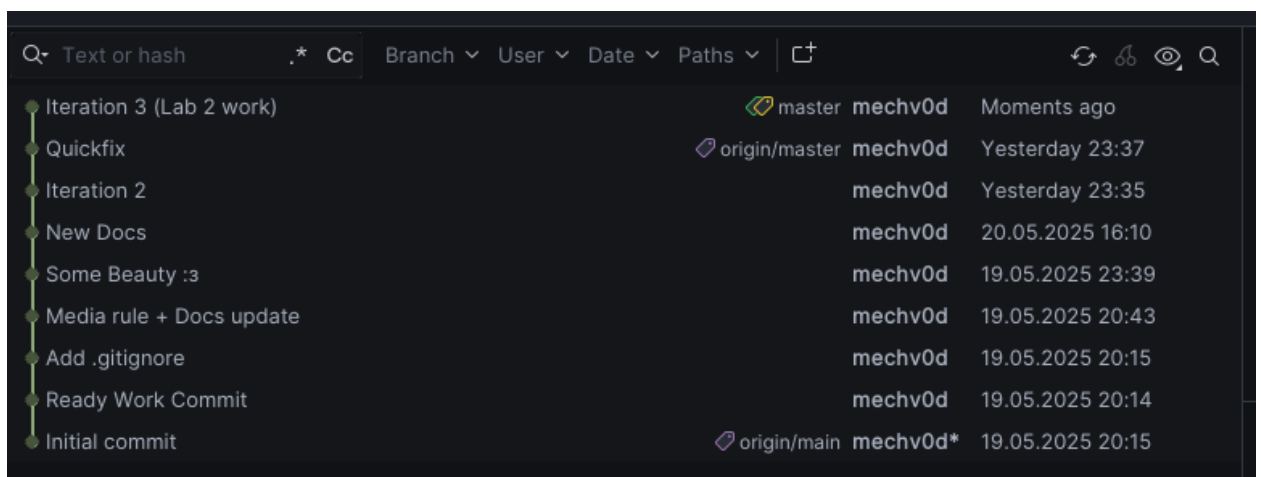


Рисунок 9 – История коммитов

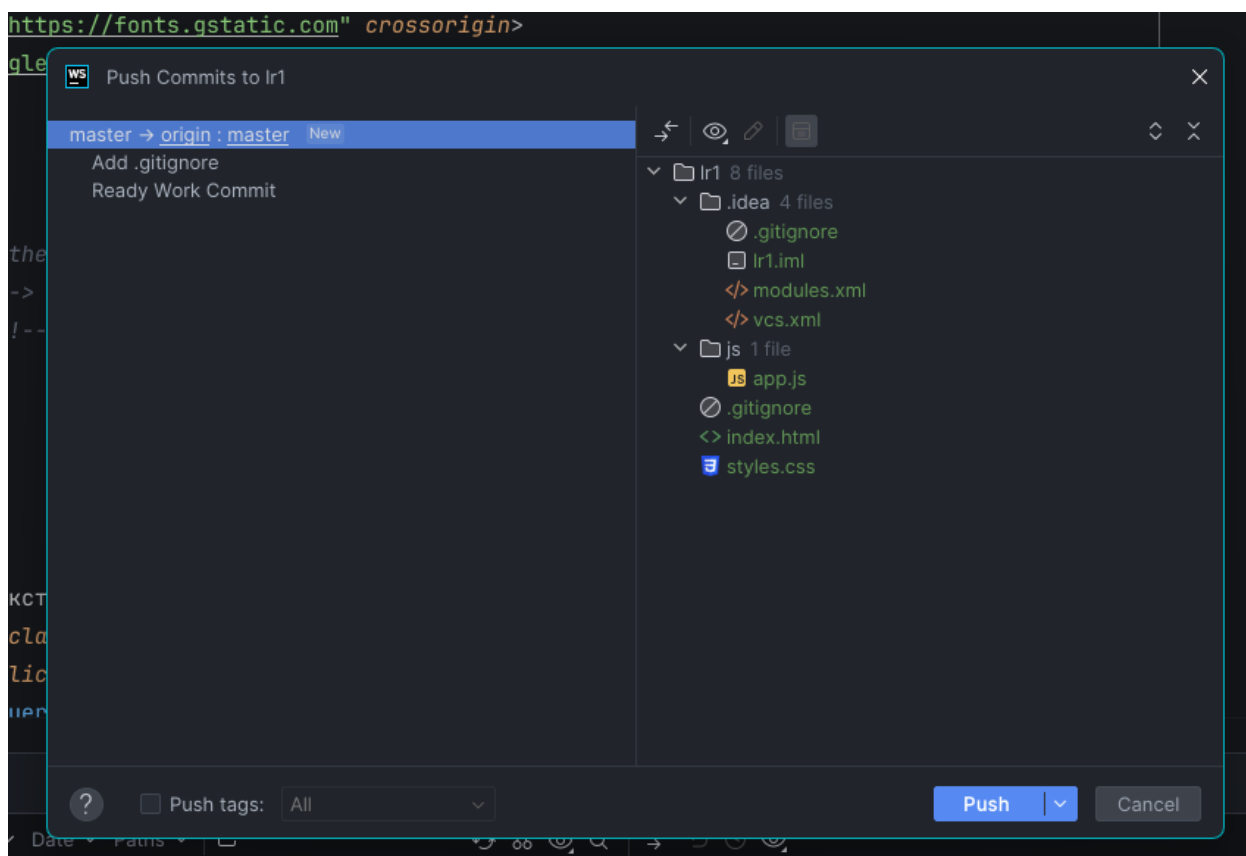


Рисунок 10 – Push на удаленный репозиторий

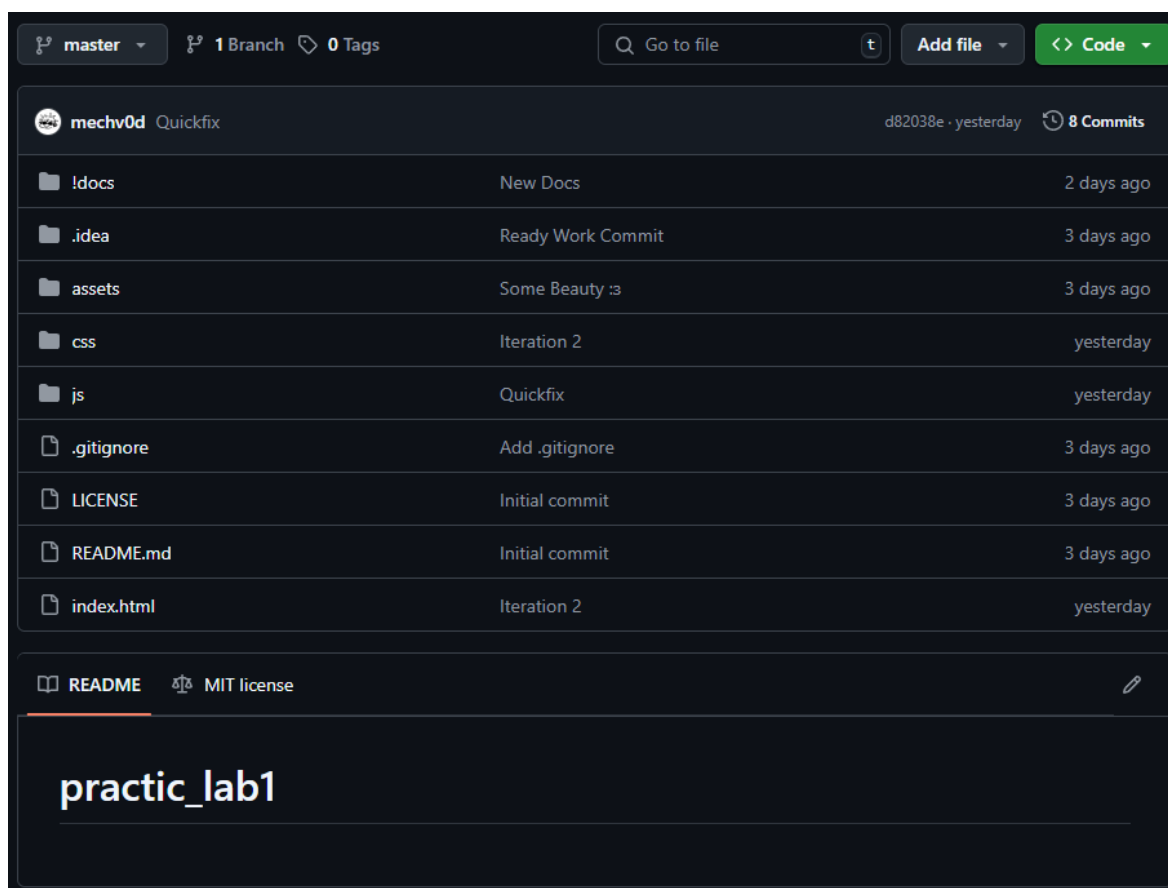


Рисунок 11 – Удаленный репозиторий

# ДЕ/ШИФРАТОР

Формат: ROT13

Текст для шифровки

HPrivet Mir!

Зашифровать

Готовый шифр:

UCevirg Zve!

Скопировать

Алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Операций: 12    Выполнено: 10    Пропущено: 2

H → U P → C r → e i → v v → i e → r t → g M → Z i → v r → e



ROT13 – это простой шифр подстановки, который заменяет каждую букву алфавита на 13-ю по счёту после неё: (A → N, B → O, ..., M → Z, N → A и т. д.).

Он работает только с латинскими буквами, а цифры и символы оставляет без изменений.

Особенность ROT13 в том, что это обратимый шифр: повторное применение ROT13 возвращает исходный текст. Изначально он использовался в интернете для сокрытия спойлеров и шуток, а не для реальной защиты данных.

Рисунок 12 – Вид страницы с ПК

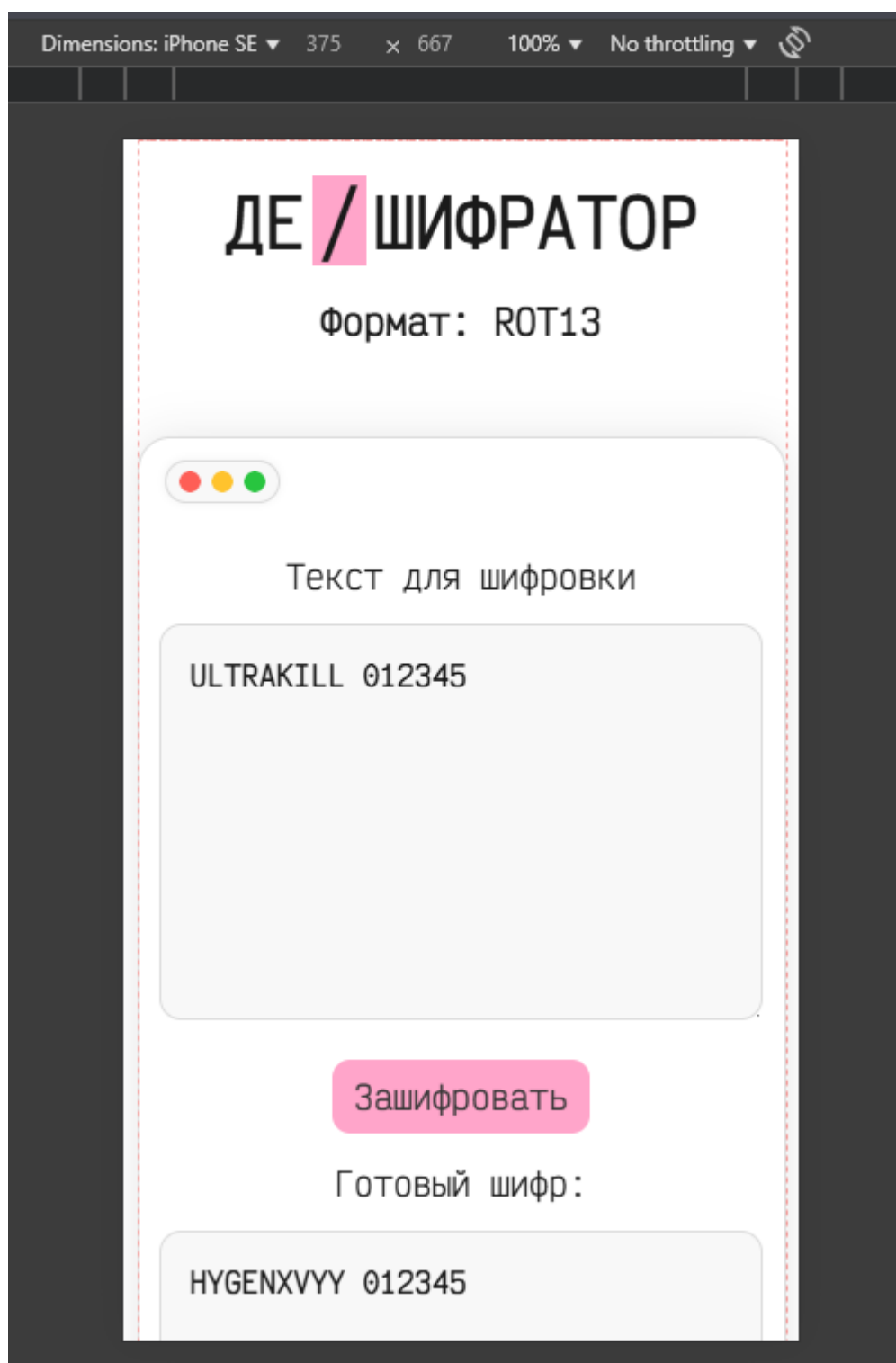


Рисунок 13 – Вид страницы с мобильного устройства (1)

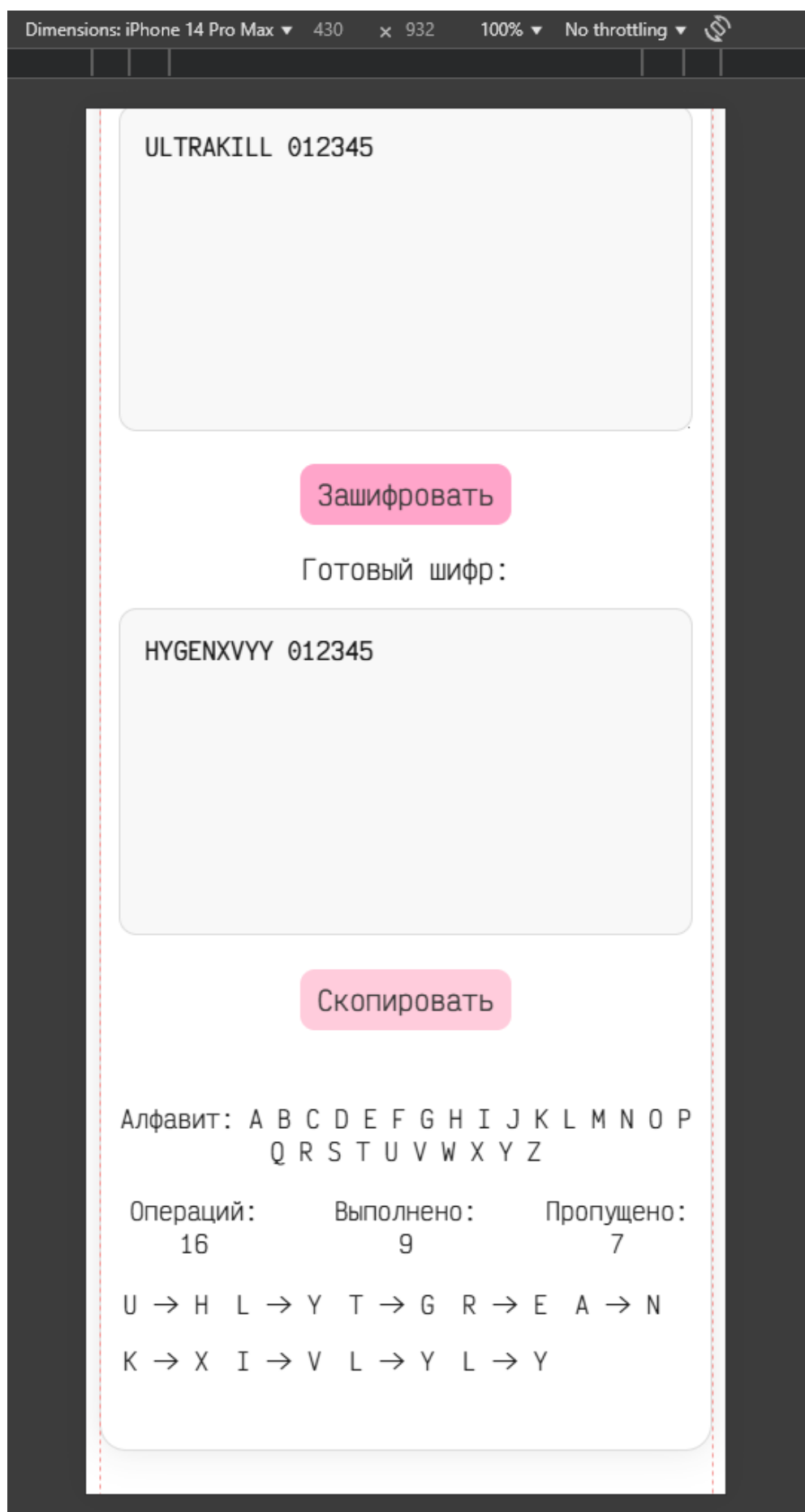


Рисунок 14 – Вид страницы с мобильного устройства (2)



Рисунок 15 – Вид страницы с мобильного устройства (3)

# ДЕ/ШИФРАТОР

Формат: ROT13

Текст для шифровки

May the powers of the Heavenly Pact bestow upon you the ultimate WAIFU status.  
  
Я - стилёвый заголовок!

Готовый шифр:

Znl gur cbjref bs gur Urnirayl Cnpg orfgbj hcba lbh gur hygvzngr JNVSH fgngfh.  
  
Я - стилёвый заголовок!

Зашифровать

Скопировать

Алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Операций: 103      Выполнено: 64      Пропущено: 39

M → Z a → n y → l t → g h → u e → r p → c o → b w → j e → r r → e s → f  
o → b f → s t → g h → u e → r H → U e → r a → n v → i e → r n → a l → y  
y → l P → C a → n c → p t → g b → o e → r s → f t → g o → b w → j u → h  
p → c o → b n → a y → l o → b u → h t → g h → u e → r u → h l → y t → g  
i → v m → z a → n t → g e → r W → J A → N I → V F → S U → H s → f t → g  
a → n t → g u → h s → f

Рисунок 16 – Пример работы

Таблица 1 – Всестороннее, полноценное тестирование на правильность работы

№	Входные данные	Выходные данные	Ожид. результат
1	May the powers of the Heavenly Pact bestow upon you the ultimate WAIFU status.  Я - стилёвый заголовок!	Znl gur cbjref bs gur Urnirayl Cnpg orfgbj hcba lbh gur hygvzngr JNVSH fgnghf.  Я - стилёвый заголовок!	Znl gur cbjref bs gur Urnirayl Cnpg orfgbj hcba lbh gur hygvzngr JNVSH fgnghf.  Я - стилёвый заголовок!
2	LOREM IPSUM? НЕТ СПАСИБО, У МЕНЯ ЕСТЬ  ВЕРХНИЙ ТЕКСТ НИЖНИЙ ТЕКСТ	YBERZ VCFHZ? НЕТ СПАСИБО, У МЕНЯ ЕСТЬ  ВЕРХНИЙ ТЕКСТ НИЖНИЙ ТЕКСТ	YBERZ VCFHZ? НЕТ СПАСИБО, У МЕНЯ ЕСТЬ  ВЕРХНИЙ ТЕКСТ НИЖНИЙ ТЕКСТ
3	Kept you waiting, huh?	Xrcg lbh jnvgvat, uhu?	Xrcg lbh jnvgvat, uhu?
4	Privet Mir!	Cevirg Zve!	Cevirg Zve!
5	Я воровал сырки в магните	Я воровал сырки в магните	Я воровал сырки в магните



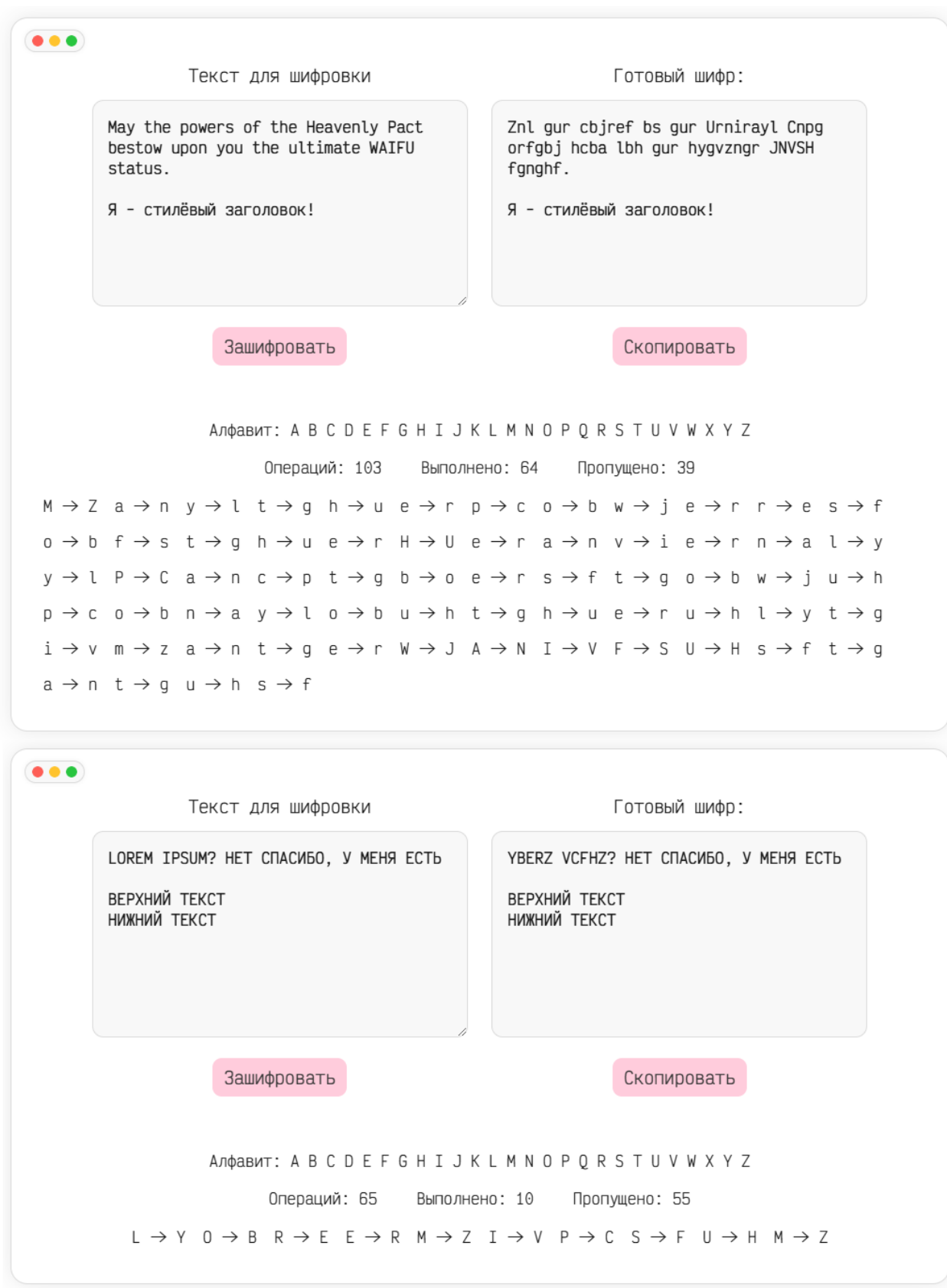


Рисунок 17 – Тесты 1 и 2

Текст для шифровки

Kept you waiting, huh?

Зашифровать

Готовый шифр:

Xrcg lbh jnvgvat, uhu?

Скопировано, Босс!

Алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Операций: 22    Выполнено: 17    Пропущено: 5

K → X   e → r   p → c   t → g   y → l   o → b   u → h   w → j   a → n   i → v   t → g   i → v  
n → a   g → t   h → u   u → h   h → u

Текст для шифровки

Privet Mir!

Зашифровать

Готовый шифр:

Cevirg Zve!

Скопировать

Алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Операций: 11    Выполнено: 9    Пропущено: 2

P → C   r → e   i → v   v → i   e → r   t → g   M → Z   i → v   r → e

Текст для шифровки

Я воровал сырки в магните

Зашифровать

Готовый шифр:

Я воровал сырки в магните

Скопировано, Босс!

Алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Операций: 25    Выполнено: 0    Пропущено: 25

Рисунок 18 – Тесты 3, 4 и 5