МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«КУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет физики, математики, информатики

Кафедра программного обеспечения и администрирования информационных систем

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине

Интерфейсы программирования приложений

на тему: *Проектирование и реализация модели поведения в веб-приложениях*

Обучающегося 2 курса очной формы обучения

направления подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль)

Прикладной искусственный интеллект

Гордеева Ярослава Сергеевича

Руководитель:

старший преподаватель кафедры ПОАИС

Ураева Елена Евгеньевна

Курск, 2025

**Индивидуальное задание**

1. Используя веб-интерфейс пользователя, спроектированный в лабораторной работе 3, реализовать моделирование действий пользователя, согласно сценариям, описанным в лабораторной 1. Допустимо использовать статичные данные для элементов ввода\вывода.
2. Реализовать систему оповещений для альтернативных веток сценария.
3. В процессе реализации модели поведения веб-приложения необходимо следовать указаниям:

– по крайней мере на одной из веб-страниц интерфейса необходимо реализовать форму, содержащую не менее 5 элементов;

– для проверки корректности заполнения данных формы необходимо реализовать и использовать по крайней мере один JavaScript;

– использовать обращение к элементам формы через методы DOM (поиск, доступ), при этом реализовать по крайней мере один случай динамического наполнения страницы (изображение, текст, элементы document) за счет вызова обработчика событий;

– реализовать по крайней мере одно событие (для мыши или клавиатуры) для моделирования действия пользователя согласно сценария;

– реализовать всплывающие информационные окна и диалоговые окна для альтернативных веток сценария (можно также использовать всплывающее окно для ввода данных);

– использовать по крайней мере один JavaScript с внешним подключением.

**Интерфейс веб-клиента**

На рисунках 1-8 представлена реализация модели поведения веб-приложения на JS.

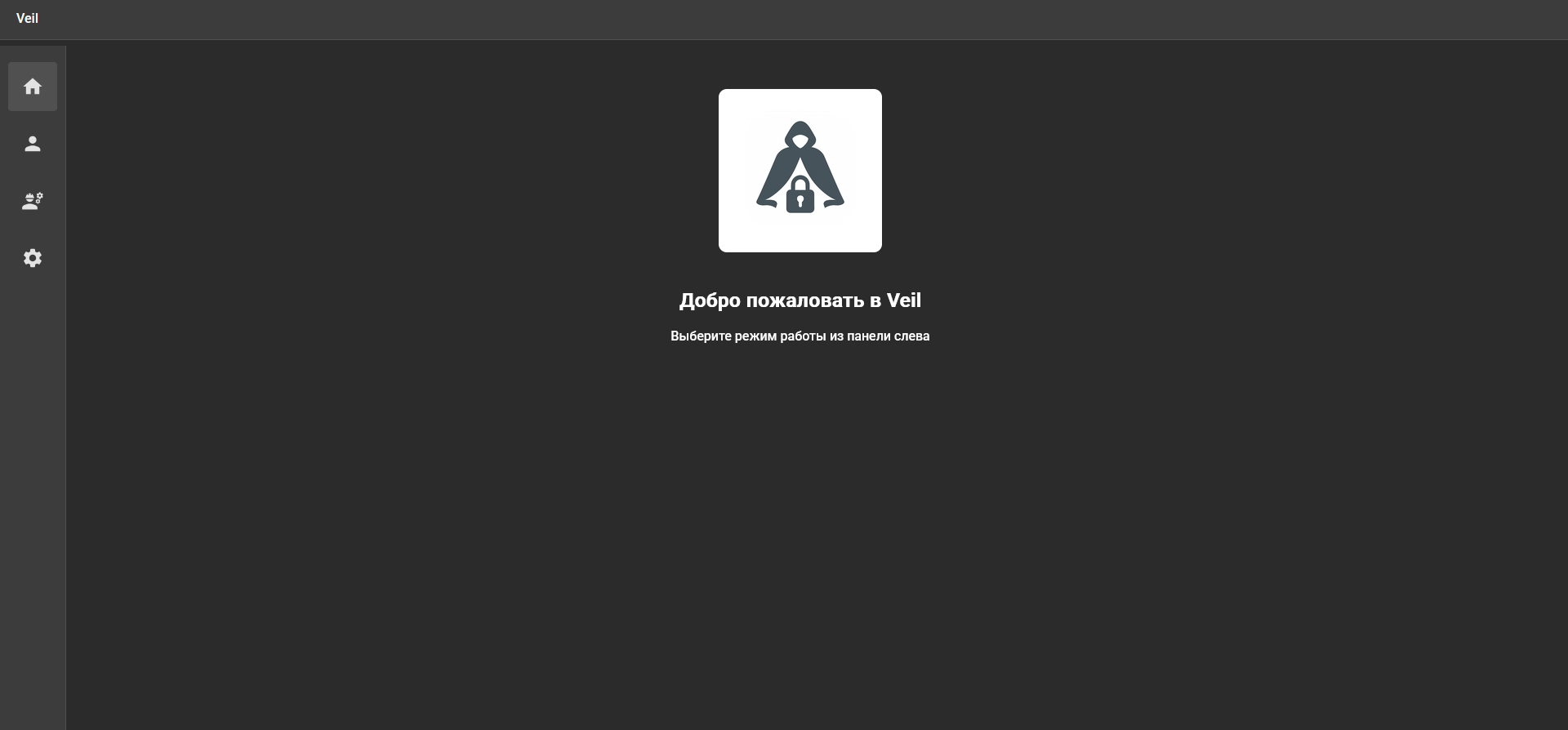


Рисунок 1 – Главная страница сайта

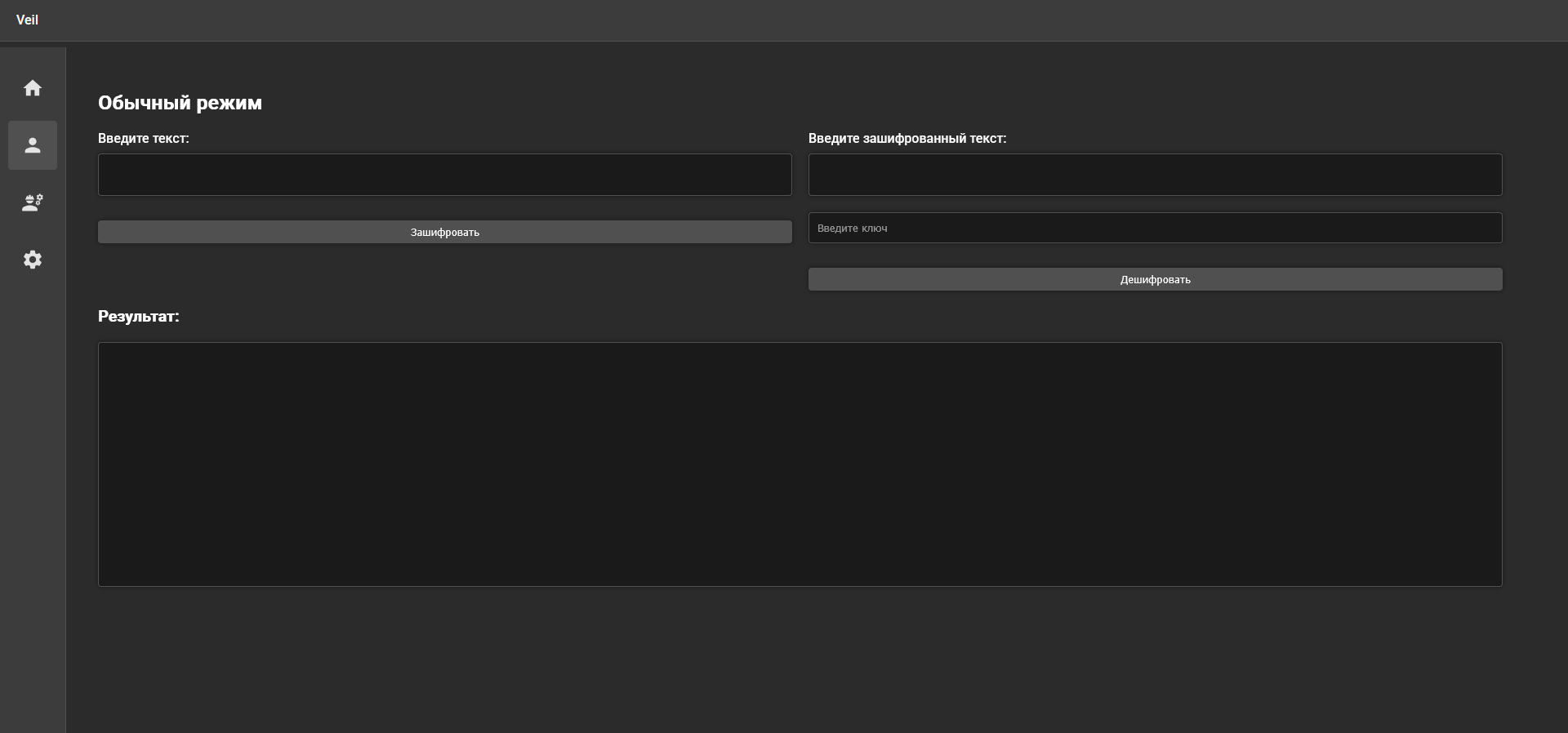


Рисунок 2 – Страница обычного режима

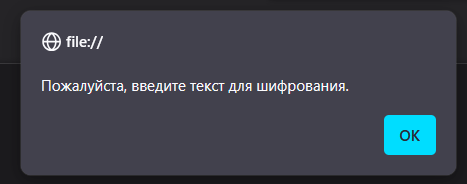


Рисунок 3 – Обработка формы на странице обычного режима

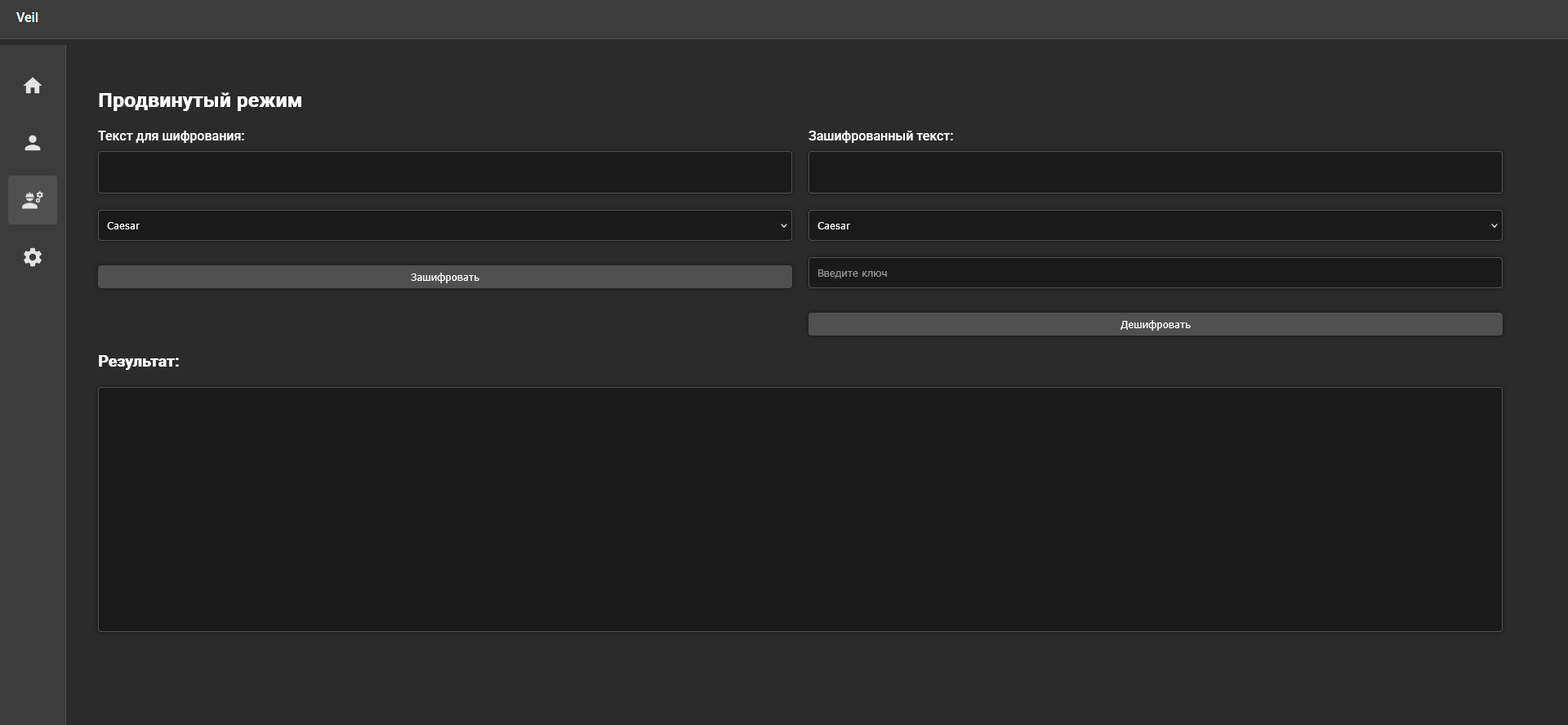


Рисунок 4 – Страница продвинутого режима

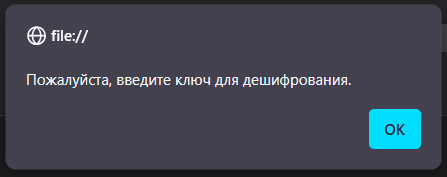
****

Рисунок 5 – Обработка формы на странице продвинутого режима



Рисунок 6 – Событие при наведении мыши на кнопку (подсветка)

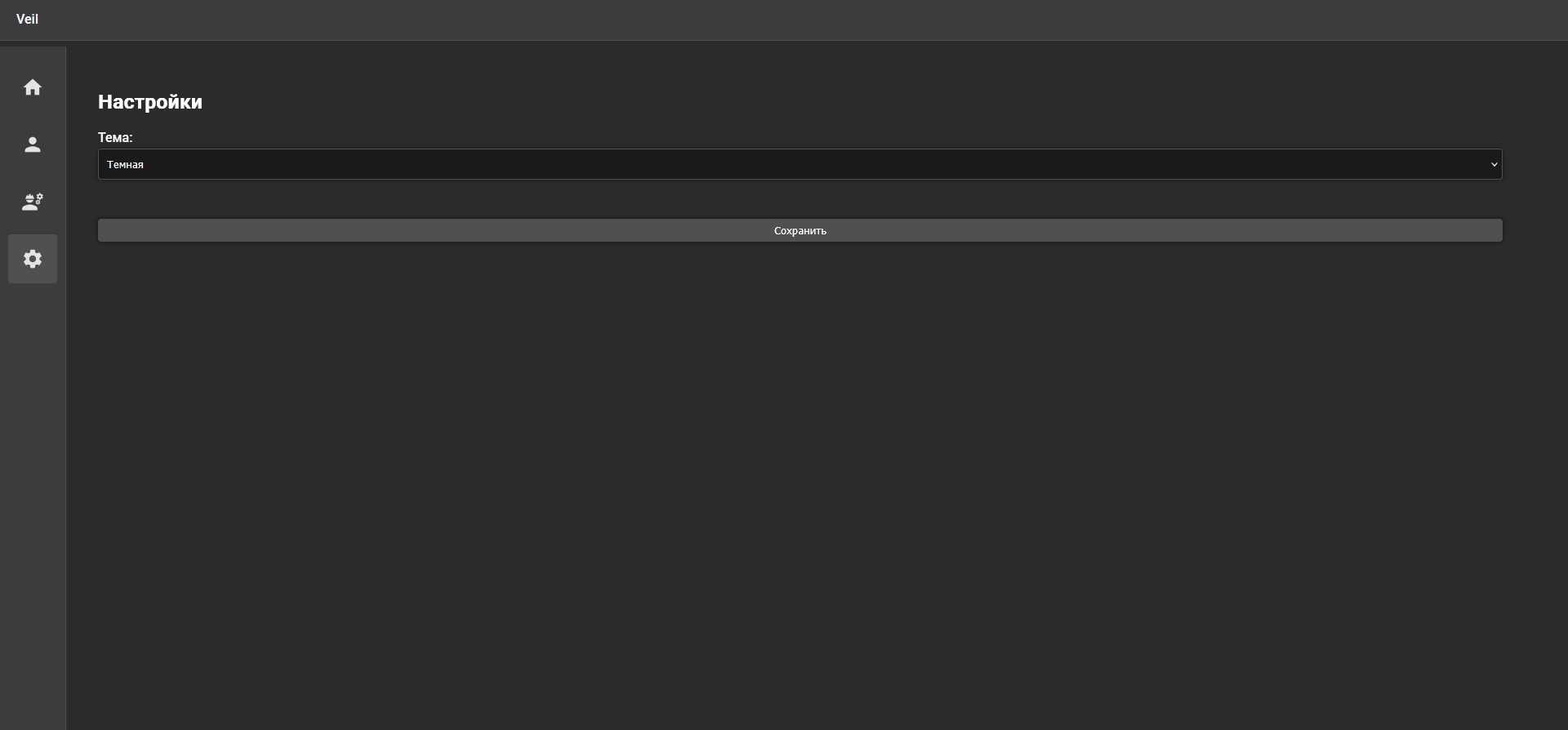


Рисунок 7 – Страница настроек

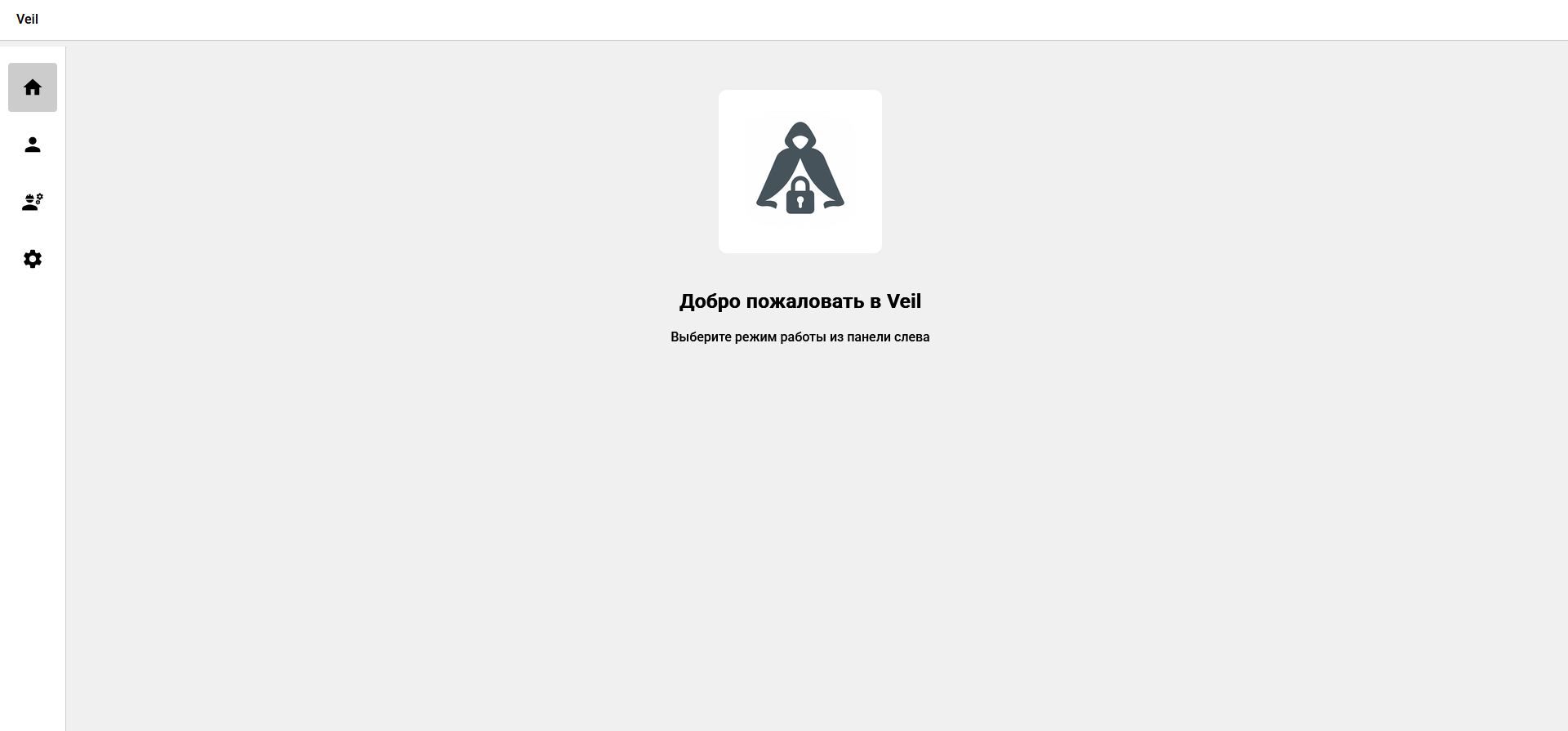


Рисунок 8 – Светлая тема сайта

**Проект в среде разработке**

Ниже представлен код веб-приложения в среде разработки HTML и его модель поведения на JS:

index.html :

<!DOCTYPE html>

<html lang="ru">

<head>

  <meta charset="UTF-8"/>

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

  <title>Veil</title>

  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@200;300;500&display=swap" rel="stylesheet">

  <link rel="stylesheet" href="styles.css" />

</head>

<body>

  <header class="header">

    <div class="logo">Veil</div>

    <div class="controls">

      <button class="btn-minimize" aria-label="Свернуть"></button>

      <button class="btn-maximize" aria-label="Развернуть"></button>

      <button class="btn-close" aria-label="Закрыть"></button>

    </div>

  </header>

  <nav class="sidebar">

    <label for="tab-home" class="sidebar-item" data-tab="home"><img src="icons/home.svg" alt="Home" class="icon"></label>

    <label for="tab-basic" class="sidebar-item" data-tab="basic"><img src="icons/user.svg" alt="Basic Mode" class="icon"></label>

    <label for="tab-advanced" class="sidebar-item" data-tab="advanced"><img src="icons/advanced.svg" alt="Advanced Mode" class="icon"></label>

    <label for="tab-settings" class="sidebar-item" data-tab="settings"><img src="icons/settings.svg" alt="Settings" class="icon"></label>

  </nav>

  <main class="content">

    <section id="home" class="page active">

      <div class="home-content">

        <img src="icons/logo.png" alt="Veil Logo">

        <h1>Добро пожаловать в Veil</h1>

        <p>Выберите режим работы из панели слева</p>

      </div>

    </section>

    <section id="basic" class="page">

      <h2>Обычный режим</h2>

      <form class="basic-form" novalidate>

        <div class="split-container">

          <div class="input-group">

            <label>Введите текст:</label>

            <textarea class="input-text"></textarea>

            <button type="submit" name="action" value="encrypt" class="btn encrypt">Зашифровать</button>

          </div>

          <div class="input-group">

            <label>Введите зашифрованный текст:</label>

            <textarea class="input-text encrypted"></textarea>

            <input type="text" class="input-key" placeholder="Введите ключ">

            <button type="submit" name="action" value="decrypt" class="btn decrypt">Дешифровать</button>

          </div>

        </div>

        <div class="result-section">

          <h3>Результат:</h3>

          <textarea class="result-text" readonly></textarea>

        </div>

      </form>

    </section>

    <section id="advanced" class="page">

      <h2>Продвинутый режим</h2>

      <form class="advanced-form" novalidate>

        <div class="split-container">

          <div class="input-group">

            <label>Текст для шифрования:</label>

            <textarea class="input-text"></textarea>

            <select class="algorithm"><option>Caesar</option><option>AES</option><option>RSA</option></select>

            <button type="submit" name="action" value="encrypt" class="btn encrypt">Зашифровать</button>

          </div>

          <div class="input-group">

            <label>Зашифрованный текст:</label>

            <textarea class="input-text encrypted"></textarea>

            <select class="algorithm"><option>Caesar</option><option>AES</option><option>RSA</option></select>

            <input type="text" class="input-key" placeholder="Введите ключ">

            <button type="submit" name="action" value="decrypt" class="btn decrypt">Дешифровать</button>

          </div>

        </div>

        <div class="result-section">

          <h3>Результат:</h3>

          <textarea class="result-text" readonly></textarea>

        </div>

      </form>

    </section>

    <section id="settings" class="page">

      <h2>Настройки</h2>

      <form class="settings-form">

        <label>Тема:

          <select class="theme-select">

            <option value="dark">Темная</option>

            <option value="light">Светлая</option>

          </select>

        </label>

        <button class="btn save" type="button">Сохранить</button>

      </form>

    </section>

  </main>

  <script src="script.js"></script>

</body>

</html>

style.css :

:root {

    --dark-bg: #2B2B2B;

    --card-bg: #3C3C3C;

    --border: #505050;

    --text-light: #FFFFFF;

    --input-bg: #1A1A1A;

  }

  body {

    margin: 0;

    font-family: 'Roboto', sans-serif;

    min-width: 1200px;

    background: var(--dark-bg);

    color: var(--text-light);

  }

  .home-content {

    text-align: center;

    padding: 20px;

  }

  .home-content img {

    max-width: 100%;

    height: auto;

    margin-bottom: 20px;

    border-radius: 10px;

    background-color: #fff;

    padding: 10px;

    max-height: 20vh;

  }

  .home-content p {

    margin: 5px 0;

  }

  textarea::-webkit-scrollbar { width: 8px; }

  textarea::-webkit-scrollbar-track { background: var(--card-bg); }

  textarea::-webkit-scrollbar-thumb { background: var(--border); border-radius: 4px; }

  textarea { scrollbar-width: thin; scrollbar-color: var(--border) var(--card-bg); }

  .header {

    display: flex;

    justify-content: space-between;

    align-items: center;

    padding: 10px 20px;

    background-color: var(--card-bg);

    border-bottom: 1px solid var(--border);

  }

  .controls button {

    width: 32px;

    height: 32px;

    background-color: transparent;

    border: none;

    cursor: pointer;

    margin-left: 8px;

    border-radius: 4px;

    background-position: center;

    background-repeat: no-repeat;

    background-size: 16px;

  }

  .btn-minimize { background-image: url("icons/minimize.svg"); }

  .btn-maximize { background-image: url("icons/maximize.svg"); }

  .btn-close    { background-image: url("icons/close.svg"); }

  .controls button:hover { background-color: var(--border); }

  .sidebar {

    position: fixed;

    left: 0;

    top: 60px;

    bottom: 0;

    width: 80px;

    height: 100vh;

    background-color: var(--card-bg);

    border-right: 1px solid var(--border);

    display: flex;

    flex-direction: column;

    align-items: center;

    padding-top: 20px;

  }

  .sidebar-item {

    width: 60px;

    height: 60px;

    margin-bottom: 10px;

    display: flex;

    align-items: center;

    justify-content: center;

    text-decoration: none;

    border-radius: 4px;

    transition: background-color 0.2s;

  }

  .sidebar-item:hover { background-color: var(--border); }

  .sidebar-item .icon { width: 28px; height: 28px; }

  .content {

    margin-left: 80px;

    padding: 20px;

    position: relative;

  }

  .split-container {

    display: flex;

    gap: 20px;

    margin-bottom: 20px;

  }

  .input-group { flex: 1; }

  .input-text,

  .input-key,

  select,

  .result-text {

    background-color: var(--input-bg);

    border: 1px solid var(--border);

    box-shadow: 0 0 5px rgba(0, 0, 0, 0.5);

    color: var(--text-light);

    padding: 10px;

    width: 100%;

    margin-bottom: 20px;

    box-sizing: border-box;

    resize: none;

    border-radius: 4px;

  }

  .result-text {

    height: 300px;

  }

  .input-group label {

    display: block;

    margin-bottom: 8px;

  }

  .settings-form {

    display: flex;

    flex-direction: column;

    gap: 10px;

  }

  .btn {

    width: 100%;

    padding: 6px 10px;

    box-shadow: 0 0 5px rgba(0, 0, 0, 0.5);

    background-color: var(--border);

    color: var(--text-light);

    border: none;

    border-radius: 4px;

    cursor: pointer;

    margin-top: 10px;

    transition: background-color 0.2s;

  }

  .btn:hover { background-color: #606060; }

  .settings-form {

    display: flex;

    flex-direction: column;

    gap: 10px;

  }

  .settings-form label {

    margin-bottom: 8px;

    display: inline-block;

  }

  .settings-form select {

    display: block;

    margin-top: 4px;

  }

  body.light-theme {

    --dark-bg: #f0f0f0;

    --card-bg: #ffffff;

    --border: #cccccc;

    --text-light: #000000;

    --input-bg: #f9f9f9;

  }

  body.dark-theme {

    --dark-bg: #2B2B2B;

    --card-bg: #3C3C3C;

    --border: #505050;

    --text-light: #FFFFFF;

    --input-bg: #1A1A1A;

  }

  .sidebar-item.active {

    background-color: #707070;

    background-color: var(--border);

  }

  .page {

    position: absolute;

    top: 0;

    left: 0;

    width: calc(100% - 80px);

    padding: 20px;

    opacity: 0;

    visibility: hidden;

    transform: translateY(10px);

    transition: opacity 0.4s ease, transform 0.4s ease;

    z-index: 0;

  }

  .page.active {

    position: relative;

    opacity: 1;

    visibility: visible;

    transform: translateY(0);

    z-index: 1;

  }

.result-text.loading {

  background: linear-gradient(90deg, #444 25%, #555 37%, #444 63%);

  background-size: 400% 100%;

  animation: loadingShimmer 1.2s ease infinite;

  color: transparent;

  caret-color: transparent;

}

@keyframes loadingShimmer {

  0% { background-position: 100% 0; }

  100% { background-position: -100% 0; }

}

body.dark-theme .sidebar-item .icon,

body.dark-theme .controls button {

  filter: none;

}

body.light-theme .sidebar-item .icon,

body.light-theme .controls button {

  filter: invert(1) brightness(0) saturate(100%);

}

.controls button {

  cursor: pointer;

}

.sidebar-item,

.sidebar-item .icon {

  cursor: pointer;

}

script.js :

document.addEventListener("DOMContentLoaded", () => {

    const pages = document.querySelectorAll(".page");

    const navButtons = document.querySelectorAll(".sidebar-item");

    const themeSelect = document.querySelector(".theme-select");

    const saveBtn = document.querySelector(".btn.save");

    const resultTextAreas = document.querySelectorAll(".result-text");

    const applyTheme = (theme) => {

      document.body.classList.toggle("light-theme", theme === "light");

      document.body.classList.toggle("dark-theme", theme === "dark");

      themeSelect.value = theme;

    };

    const savedTheme = localStorage.getItem("theme") || "dark";

    applyTheme(savedTheme);

    saveBtn.addEventListener("click", () => {

      const selected = themeSelect.value;

      localStorage.setItem("theme", selected);

      applyTheme(selected);

    });

    navButtons.forEach((btn) => {

      btn.addEventListener("click", () => {

        const targetId = btn.dataset.tab;

        pages.forEach((page) => {

          page.classList.remove("active");

          if (page.id === targetId) page.classList.add("active");

        });

        navButtons.forEach((b) => b.classList.remove("active"));

        btn.classList.add("active");

      });

    });

    function showSkeleton(form) {

      form.querySelectorAll(".result-text").forEach(area => {

        area.value = "";

        area.classList.add("loading");

      });

    }

    function hideSkeletonWithResult(form, result) {

      form.querySelectorAll(".result-text").forEach(area => {

        area.classList.remove("loading");

        area.value = result;

      });

    }

    document.querySelectorAll("form.basic-form, form.advanced-form").forEach(form => {

      form.addEventListener("submit", e => {

        e.preventDefault();

        const action = e.submitter.value;

        const container = e.submitter.closest(".split-container");

        const textArea = action === 'encrypt'

          ? container.querySelector("textarea:not(.encrypted)")

          : container.querySelector("textarea.encrypted");

        const keyInput = container.querySelector(".input-key");

        if (!textArea.value.trim()) {

          alert(action === 'encrypt'

            ? "Пожалуйста, введите текст для шифрования."

            : "Пожалуйста, введите зашифрованный текст для дешифрования.");

          return;

        }

        if (action === 'decrypt' && keyInput && !keyInput.value.trim()) {

          alert("Пожалуйста, введите ключ для дешифрования.");

          return;

        }

        showSkeleton(form);

        setTimeout(() => {

          const dummy = action === 'encrypt'

            ? "Зашифрованный текст"

            : "Дешифрованный текст";

          hideSkeletonWithResult(form, dummy);

        }, 1500);

      });

    });

    navButtons[0].classList.add("active");

  });