МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЕВФРОСИНИИ ПОЛОЦКОЙ»

Факультет информационных технологий

Кафедра технологий программирования

ЛАБОРАТОРНАЯ РАБОТА №7

по дисциплине «Веб - технологии»

Тема: «Разработка формы обратной связи с валидацией на JavaScript»

|  |  |
| --- | --- |
| Выполнил: | студент группы 23-ИТ-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Ю. Страпко |
| Проверил: | ассистент кафедры ТП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Макарычева |

|  |
| --- |
|  |

**Цель работы:**изучить клиентскую валидацию форм, используя HTML и JavaScript

1. **Краткая теория**

Форма – это компонент веб-страницы с элементами управления, такими как текстовые поля, кнопки, флажки, диапазон или поле выбора цвета. Формы являются неотъемлемой частью веб-разработки, позволяя пользователям взаимодействовать с веб-сайтами. Они используются для сбора данных, отправки информации на сервер и выполнения других интерактивных задач. Форма в HTML начинается с тега <form>Начало формы

, который определяет область, в которой будут находиться все элементы формы. Основные элементы формы включают:

− Тег <form> Начало формы

: контейнер для всех элементов формы.

− Тег <input> : используется для создания различных типов полей ввода.

− Тег <label> : связывает текстовую метку с элементом формы.

− Тег <textarea> : создает кнопку для отправки формы или выполнения других действий.

− Тег <select> и <option> : создают выпадающий список.

Является одним из самых универсальных элементов формы и может использоваться для создания различных типов полей ввода, таких как текстовые поля, поля для ввода пароля, электронных адресов, числовых значений и других. Тег связывает текстовую метку с элементом формы, что улучшает доступность и удобство использования формы. Тег создает кнопку для отправки формы или выполнения других действий.

1. **Практическая часть**

**Задание:**

− Создайте форму обратной связи с полями ввода: имя, email, сообщение.

− Реализуйте проверку на минимальную длину текста и корректность email.

− Выводите сообщения об ошибках под соответствующими полями.

− Добавьте возможность сброса формы с подтверждением действия через confirm(). Дополнительно: реализуйте валидацию формы на нескольких этапах (пошаговая валидация).

Листинг 2.1 – файл first.html.

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <link rel="stylesheet" href="styles.css">
5. <meta charset="UTF-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1.0">
7. <title>First</title>
8. </head>
9. <body>
10. <div class="form-group">
11. <p>Имя</p>
12. <input type="text" id="username">
13. <div id="errorname" class="divname"></div>
14. </div>
15. <div class="form-group">
16. <p>Фамилия</p>
17. <input type="text" id="secname">
18. <div id="errorsec" class="divname"></div>
19. </div>
20. <div class="form-group">
21. <p>Номер</p>
22. <input type="text" id="number">
23. <div id="errornumber" class="divname"></div>
24. </div>
25. <div class="form-group">
26. <p>E-mail</p>
27. <input type="email" id="email">
28. <div id="erroremail" class="divname"></div>
29. </div>
30. <div class="form-group">
31. <p>Сообщение</p>
32. <input type="text" id="comment">
33. <div id="errorcomment" class="divname"></div>
34. </div>
35. <button id="btnSell">Отправить</button>
36. <button id="btnDelete">Сбросить</button>
37. <script src="script.js"></script>
38. </body>
39. </html>

Листинг 2.2 – файл script.js.

1. var username = document.getElementById('username');
2. var secname = document.getElementById('secname');
3. var number = document.getElementById('number');
4. var email = document.getElementById('email');
5. var comment = document.getElementById('comment');
6. var btnSell = document.getElementById('btnSell');
7. var btnDelete = document.getElementById('btnDelete');
8. const regexNumber = /^\+375(29|25|33|44)\d{7}$/;
9. const regexEmail = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
10. btnSell.addEventListener('click',function(){
11. var errorMessage = "";
12. if(username.value.trim()=='')
13. errorMessage = "Это поле не может быть пустым";
14. errorname.textContent = errorMessage;
15. });
16. btnSell.addEventListener('click',function(){
17. var errorMessage = "";
18. if(secname.value.trim()=='')
19. errorMessage = "Это поле не может быть пустым";
20. errorsec.textContent = errorMessage;
21. });
22. btnSell.addEventListener('click',function(){
23. var errorMessage = "";
24. if(!regexNumber.test(number.value))
25. errorMessage = "Введите корректный номер(Например: +375291234567)";
26. errornumber.textContent = errorMessage;
27. });
28. btnSell.addEventListener('click',function(){
29. var errorMessage = "";
30. if(!regexEmail.test(email.value))
31. errorMessage = "Введите корректный E-mail";
32. erroremail.textContent = errorMessage;
33. });
34. btnSell.addEventListener('click',function(){
35. var errorMessage = "";
36. if(comment.value.trim()=='')
37. errorMessage = "Добавьте, пожалуйста, комментарий";
38. errorcomment.textContent = errorMessage;
39. confirm("Вы уверены в отправке?");
40. });
41. btnDelete.addEventListener('click',function(){
42. confirm("Вы уверены, что хотите сбросить данные?");
43. var errorMessage = "";
44. username.value = "";
45. secname.value = "";
46. number.value = "";
47. email.value = "";
48. comment.value = "";
49. });

В данном листинге реализовано следующее:

Этот код представляет собой простую форму валидации на стороне клиента с использованием JavaScript. Он проверяет введенные пользователем данные в различных полях формы и выводит сообщения об ошибках, если данные не соответствуют ожидаемым критериям. Происходит получение ссылок на элементы HTML-формы по их идентификаторам. Код добавляет обработчики событий на кнопку btnSell (предположительно, кнопку отправки формы). Каждый обработчик проверяет соответствующее поле формы и выводит сообщение об ошибке, если данные не соответствуют ожидаемым критериям.

**3.Контрольные вопросы**

**1. Что такое формы? Для чего они нужны?**

Формы — это элементы веб-страниц, предназначенные для сбора информации от пользователей. Они позволяют отправлять данные на сервер для обработки, например, для регистрации, обратной связи или покупок.

**2. Перечислите основные элементы формы?**

Основные элементы формы включают:

* <input> — для ввода данных.
* <textarea> — для многострочного текста.
* <select> — для выпадающих списков.
* <button> — для кнопок.
* <label> — для меток полей.

**3. Перечислите типы полей, которые могут быть указаны в теге <input>.**

Типы полей для тега <input>:

* text — текстовое поле.
* password — поле для ввода пароля.
* email — поле для ввода адреса электронной почты.
* number — числовое поле.
* date — поле для выбора даты.
* checkbox — флажок.
* radio — радиокнопка.
* file — для загрузки файлов.
* url — для ввода URL-адреса.

**4. Что такое валидация?**

Валидация — это процесс проверки введенных данных на соответствие заданным критериям или правилам перед их отправкой на сервер. Это помогает предотвратить ошибки и недействительные данные.

**5. Какое преимущество использования валидации в HTML?**

Преимущества валидации в HTML включают:

* Улучшение качества данных.
* Снижение нагрузки на сервер, так как ошибки могут быть обработаны на стороне клиента.
* Повышение удобства для пользователей за счет мгновенных уведомлений о неверно введенных данных.

**6. Опишите атрибут required для создания валидации.**

Атрибут required указывает, что поле обязательно для заполнения. Если пользователь попытается отправить форму, не заполнив это поле, браузер предупредит его об этом.

**7. Опишите атрибуты max и min для создания валидации.**

Атрибуты max и min используются для определения пределов значений в числовых и датах полях. Например, для поля типа number:

* min="1" — минимально допустимое значение.
* max="10" — максимально допустимое значение.

**8. Опишите атрибут pattern для создания валидации.**

Атрибут pattern задает регулярное выражение, которому должно соответствовать значение поля. Если введенные данные не соответствуют шаблону, форма не будет отправлена.

**9. Что такое регулярное выражение?**

Регулярное выражение — это последовательность символов, определяющая шаблон для поиска и обработки строк. Они используются для проверки формата данных, таких как электронные адреса или номера телефонов.

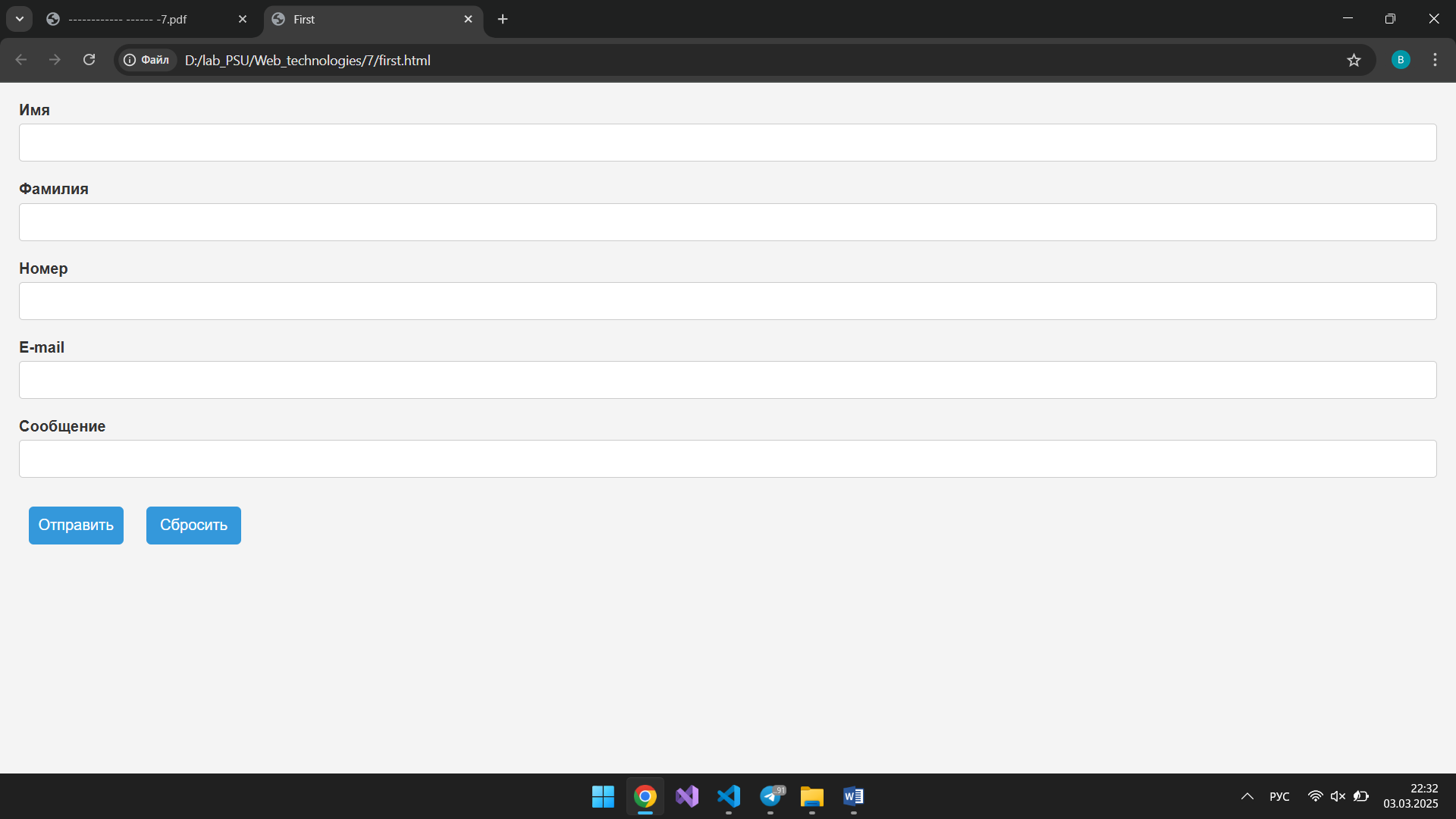
**10. Как отключить валидацию?**

Валидацию можно отключить, используя атрибут novalidate в теге <form>.

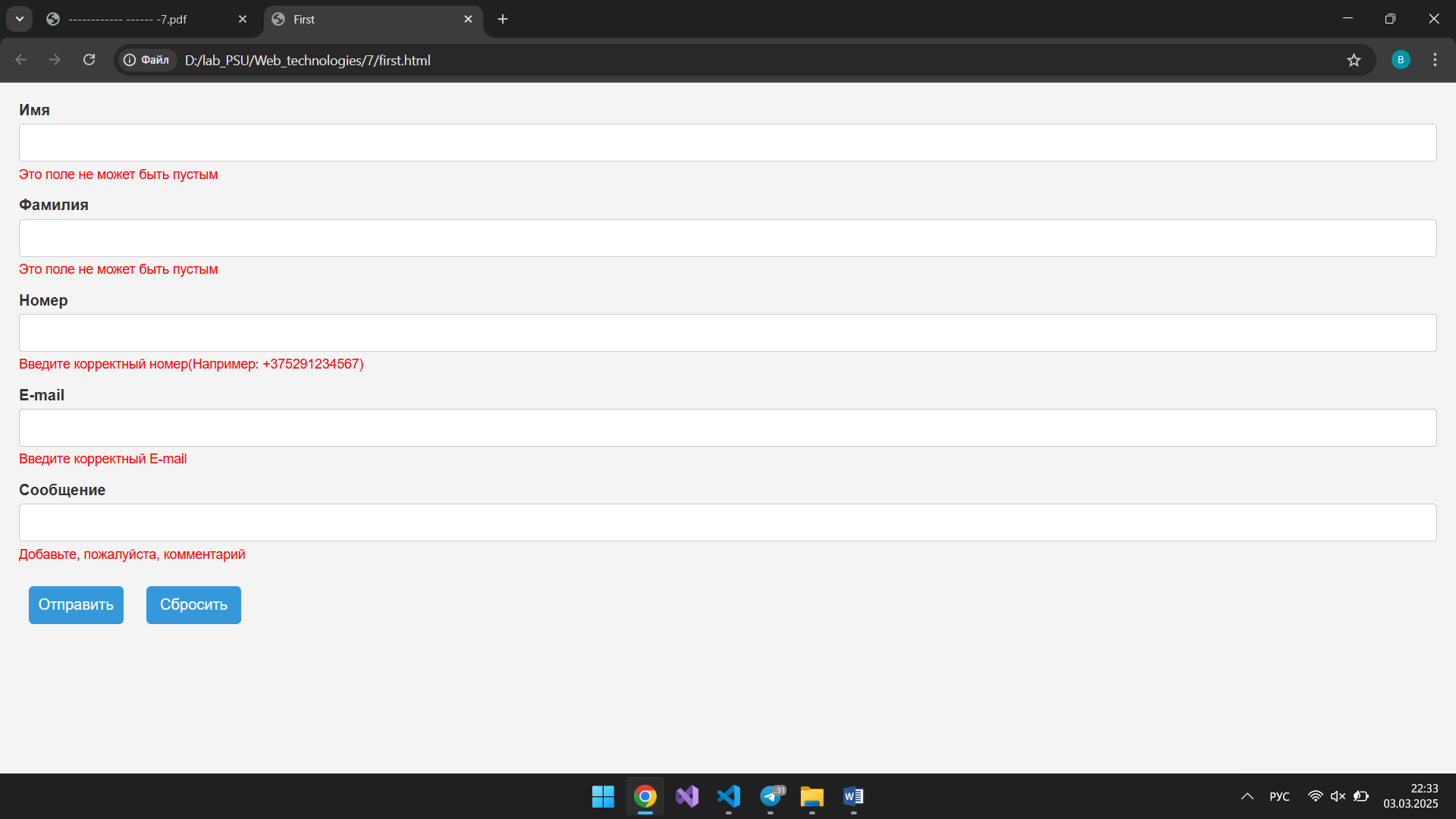
**11. Что необходимо сделать, чтобы запросить подтверждение перед отправкой формы?**

Чтобы запросить подтверждение перед отправкой формы, можно использовать JavaScript. Например, добавив обработчик события onsubmit. Этот код вызовет диалоговое окно с подтверждением, и только при положительном ответе форма будет отправлена.

**4.Скриншоты результата**



**Рисунок 4.1 – Форма**



**Рисунок 4.2 – Вывод ошибок**

**Вывод:** в ходе данной лабораторной работы были изучены принципы работы клиентской валидации форм, используя HTML и JavaScript.