

Лабораторная работа № 15

Тема: Пользовательская интерактивность.

Общие требования:

- Использовать только addEventListener для обработки событий.
- Управление состоянием через CSS-классы (completed, hidden).
- Разделять логику и оформление.
- Код размещать во внешнем файле script.js.
- Все изменения интерфейса должны давать явную обратную связь пользователю.

Вариант 1.

HTML-шаблон:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Интерактивная страница</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

<h2>Выбор тарифа</h2>
<div class="cards">
  <div class="card">Базовый</div>
  <div class="card">Стандарт</div>
  <div class="card">Премиум</div>
</div>

<h2>Комментарий</h2>
<textarea id="comment" placeholder="Введите комментарий"></textarea>
<p id="commentMessage"></p>
<button id="submitBtn" disabled>Отправить комментарий</button>

<h2>Дополнительная информация</h2>
<button id="toggleInfo">Показать</button>
<div id="extraInfo" class="hidden">
  <p>Здесь подробная информация о выбранном тарифе и
условиях.</p>
</div>
```

```
<h2>Журнал действий пользователя</h2>
<ul id="logList"></ul>
```

```
<script src="script.js"></script>
</body>
</html>
```

CSS (базовое оформление):

```
.card {
  padding: 10px;
  border: 1px solid #ccc;
  margin: 5px;
  cursor: pointer;
  display: inline-block;
}

.card.active {
  background-color: lightgreen;
  border-color: green;
}

.hidden {
  display: none;
}

textarea {
  display: block;
  margin-bottom: 5px;
  width: 300px;
  height: 60px;
}
```

Требования к интерактивности.

Выбор тарифа:

При клике на карточку тариф подсвечивается (класс active), а предыдущая активная карточка теряет подсветку.

В лог действий добавляется сообщение о выбранном тарифе.

Обратная связь при вводе комментария:

- Если поле пустое — показывать сообщение «Комментарий пуст» и блокировать кнопку «Отправить».
- Если поле заполнено — показывать сообщение «Комментарий готов к отправке» и разблокировать кнопку.

Отправка комментария:

- При нажатии кнопки «Отправить» добавлять комментарий в журнал действий и очищать поле ввода.
- Сообщение о комментарии отображается в списке (ul#logList).

Показ / скрытие дополнительной информации:

- Кнопка «Показать» раскрывает блок с дополнительной информацией.
- Кнопка меняет текст на «Скрыть» при открытом блоке.
- Действие записывается в журнал.

Журнал действий пользователя:

- Каждое действие (выбор тарифа, изменение текста, отправка комментария, показ/скрытие информации) добавляется в список (ul#logList).
- Использовать createElement и appendChild для добавления записей.

Вариант 2.

HTML-шаблон:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Интерактивная страница — Вариант 2</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

<h2>Список задач</h2>
<ul id="taskList">
  <li class="task">Купить продукты</li>
  <li class="task">Написать отчет</li>
  <li class="task">Позвонить клиенту</li>
```

```
</ul>
```

```
<h2>Добавить новую задачу</h2>
```

```
<input type="text" id="newTask" placeholder="Новая задача">
```

```
<button id="addTaskBtn">Добавить</button>
```

```
<p id="taskMessage"></p>
```

```
<h2>Фильтр задач</h2>
```

```
<button id="showAll">Показать все</button>
```

```
<button id="showCompleted">Показать выполненные</button>
```

```
<h2>Журнал действий пользователя</h2>
```

```
<ul id="logList"></ul>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

CSS (базовое оформление):

```
.task {  
  padding: 8px;  
  border: 1px solid #ccc;  
  margin: 3px;  
  cursor: pointer;  
}
```

```
.task.completed {  
  text-decoration: line-through;  
  color: gray;  
}
```

```
button {  
  margin: 3px;  
}
```

```
.hidden {  
  display: none;  
}
```

Требования к интерактивности.

Отметка задачи как выполненной:

- При клике на элемент списка (li.task) добавляется класс completed.
- Если элемент уже отмечен — класс убирается.
- Каждое изменение добавляется в журнал действий.

Добавление новой задачи:

- Ввод текста в поле и нажатие кнопки «Добавить» создаёт новый элемент списка (li.task) с текстом из поля.
- Поле очищается после добавления.
- В журнал действий добавляется сообщение о новой задаче.

Обратная связь при пустом поле:

- Если поле для новой задачи пустое, выводить сообщение «Введите текст задачи» и не добавлять элемент.
- Сообщение исчезает после успешного добавления.

Фильтр задач:

- Кнопка «Показать все» отображает все задачи.
- Кнопка «Показать выполненные» скрывает невыполненные задачи (li без класса completed).

Журнал действий:

- Каждое действие (отметка задачи, добавление новой задачи, применение фильтра) добавляется в список (ul#logList).
- Использовать createElement и appendChild.

 **Отчет должен содержать (см. образец):**

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна VSC с исходным кодом программ;
- скриншоты с результатами выполнения программ;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email:

colledge20education23@gmail.com