МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №11**

по дисциплине: «Веб-технологии»

на тему: «Программирование с использованием библиотеки React.»

|  |  |
| --- | --- |
| Выполнил | Студент группы 22-ИТ-1  Короткевич П.А. |
|  |  |
| Проверил | Макарычева В.А |

Полоцк 2024

**Цель:** изучить основные принципы работы библиотеки React, научиться применять полученные знания на практике.

**Задание 1**

Создайте приложение «Планировщик задач» (TO DO). Реализовать функции добавления и удаления задач. Задачи отображаются списком. Рядом с каждой задачей расположена кнопка «Выполнить». После выполнения задача подсвечивается зелёным цветом.

Листинг 1

import React, { useState } from 'react';

import './App.css';

function App() {

const [tasks, setTasks] = useState([]);

const [newTask, setNewTask] = useState('');

const addTask = () => {

if (newTask.trim() !== '') {

setTasks([...tasks, { text: newTask, completed: false }]);

setNewTask('');

}

};

const deleteTask = (index) => {

const updatedTasks = tasks.filter((\_, i) => i !== index);

setTasks(updatedTasks);

};

const completeTask = (index) => {

const updatedTasks = tasks.map((task, i) =>

i === index ? { ...task, completed: true } : task

);

setTasks(updatedTasks);

};

return (

<div className="App">

<h1>Планировщик задач</h1>

<div className="input-container">

<input

type="text"

value={newTask}

onChange={(e) => setNewTask(e.target.value)}

/>

<button className="add-button" onClick={addTask}>Добавить задачу</button>

</div>

<ul>

{tasks.map((task, index) => (

<li key={index} className={task.completed ? 'completed' : ''}>

{task.text}

<button className="complete-button" onClick={() => completeTask(index)}>Выполнить</button>

<button className="delete-button" onClick={() => deleteTask(index)}>Удалить</button>

</li>

))}

</ul>

</div>

);

}

export default App;

Этот React-компонент представляет собой простое приложение "Планировщик задач", которое позволяет пользователю добавлять задачи, помечать их как выполненные и удалять их. Приложение начинается с импорта React и хука useState из библиотеки React, а также файла стилей App.css. Внутри функции App используется хук useState для создания двух состояний: tasks и newTask. Состояние tasks представляет собой массив задач, где каждая задача является объектом с полями text и completed, а newTask хранит текст новой задачи, вводимой пользователем.

Функция addTask вызывается при нажатии на кнопку "Добавить задачу". Эта функция проверяет, что введенный текст не пустой (удаляя пробелы с помощью метода trim). Если текст не пустой, он добавляется в массив задач как новый объект с полем text, содержащим текст задачи, и свойством completed, установленным в false. После добавления задачи функция очищает поле ввода, устанавливая newTask в пустую строку.

Функция deleteTask вызывается при нажатии на кнопку "Удалить" рядом с задачей. Эта функция создает новый массив задач, исключая задачу с заданным индексом, и обновляет состояние tasks с помощью нового массива.

Функция completeTask вызывается при нажатии на кнопку "Выполнить" рядом с задачей. Эта функция создает новый массив задач, в котором задача с заданным индексом помечается как выполненная путем установки свойства completed в true. Обновленное состояние tasks затем заменяет старое состояние.

Листинг 2

.App {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

.input-container {

margin-bottom: 20px;

}

input {

padding: 10px;

margin-right: 10px;

}

button {

padding: 10px 20px;

margin: 5px;

border: none;

border-radius: 4px;

cursor: pointer;

}

ul {

list-style-type: none;

padding: 0;

margin:0 auto;

max-width: 800px;

}

li {

padding: 10px;

margin: 5px 0;

border: 1px solid #ccc;

display: flex;

justify-content: space-between;

align-items: center;

background-color: #f9f9f9;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

Продолжение листинга 2

}

li.completed {

background-color: #d4edda;

text-decoration: line-through;

}

.add-button {

background-color: #007bff;

color: white;

}

.complete-button {

background-color: #28a745;

color: white;

}

.delete-button {

background-color: #e94f29;

color: white;

}

Этот файл стилей App.css предназначен для оформления компонентов в React-приложении "Планировщик задач". Вся область приложения оформляется с использованием стиля App, который задает шрифт Arial, выравнивает текст по центру и добавляет внутренний отступ в 20 пикселей. Элемент input-container имеет отступ снизу в 20 пикселей для создания пространства между формой ввода и списком задач. Стили для элемента input включают в себя внутренний отступ в 10 пикселей и отступ справа в 10 пикселей для создания пространства между полем ввода и кнопкой.

Все кнопки в приложении имеют внутренний отступ 10 пикселей по вертикали и 20 пикселей по горизонтали, отступ в 5 пикселей, закругленные углы с радиусом в 4 пикселя и указатель курсора при наведении.

Элемент ul, который содержит список задач, имеет убранные отступы и маркеры списка, а также максимальную ширину в 800 пикселей и центрирование по горизонтали.

На следующих рисунках представлена реализация:

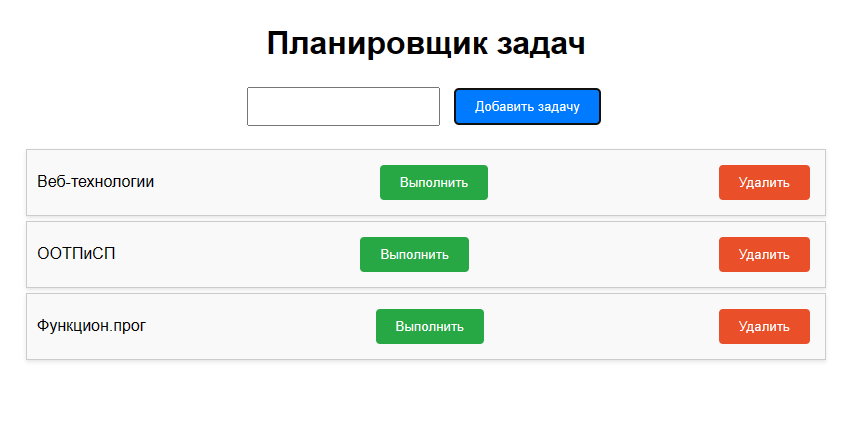


Рисунок 1 – Добавление задач

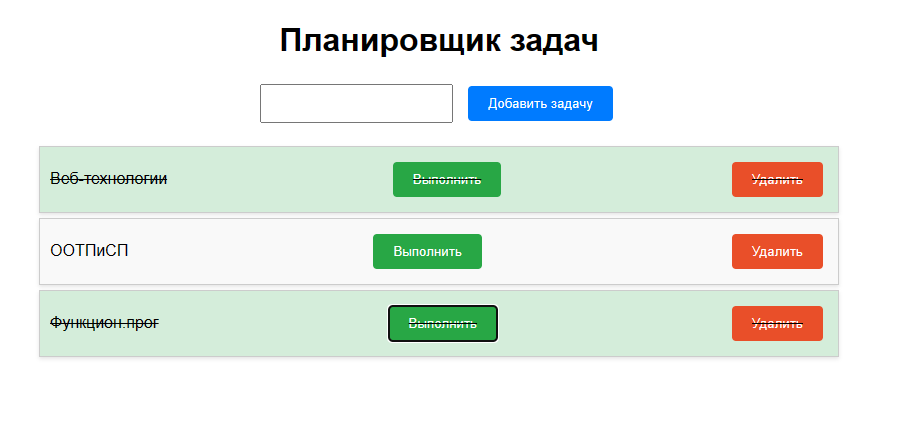


Рисунок 2 – Выполнение задач

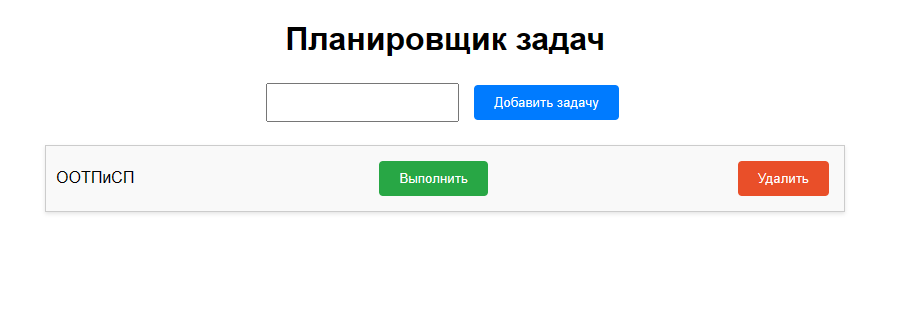


Рисунок 3 – Удаление задач

**Вывод:** Изучение основных принципов работы с библиотекой React и их практическое применение позволяет разработчикам создавать эффективные и динамичные пользовательские интерфейсы для веб-приложений. React предлагает мощные возможности благодаря своим основным концепциям, таким как компоненты, состояние и свойства (props). Компоненты, как строительные блоки React-приложений, обеспечивают модульность и повторное использование кода. Управление состоянием с использованием хуков, таких как useState, позволяет легко обновлять интерфейс в ответ на действия пользователя или изменения данных.