

Практическое задание №1: Дизайн-ориентированный To-Do List

Цель: Разработать интерактивный и стильный To-Do List, применяя знания HTML, CSS, JavaScript и принципов UI/UX.

Описание: Вам предстоит создать веб-приложение To-Do List, которое позволит пользователям добавлять, удалять, редактировать и отмечать задачи как выполненные. Особое внимание уделите визуальному оформлению и удобству использования.

Функциональные требования:

1. **Добавление задачи:** Пользователь должен иметь возможность добавлять новые задачи через текстовое поле и кнопку "Добавить" или клавишу Enter.
2. **Отображение задач:** Задачи должны отображаться в виде списка. Каждая задача должна содержать текст задачи и элементы управления (редактировать, удалить, отметить как выполненную).
3. **Удаление задачи:** Пользователь должен иметь возможность удалить задачу из списка.
4. **Редактирование задачи:** Пользователь должен иметь возможность редактировать текст существующей задачи.
5. **Отметка как выполненная:** Пользователь должен иметь возможность отметить задачу как выполненную. Визуально это должно быть четко отображено (например, зачеркиванием текста, изменением цвета, добавлением иконки).
6. **Локальное хранение (опционально, + к оценке):** Реализуйте сохранение задач в localStorage браузера, чтобы при перезагрузке страницы задачи не терялись.

Требования к дизайну и UI/UX:

1. **Современный дизайн:** Приложение должно выглядеть стильно и современно. Вы можете использовать современные тренды в дизайне (например, Neumorphism, Glassmorphism, Material Design), но не обязательно строго им следовать. Главное - чистый и приятный глазу интерфейс.
2. **Анимации:** Добавьте плавные анимации при добавлении, удалении, редактировании и отметке задач как выполненных. Анимации должны улучшать пользовательский опыт и делать взаимодействие с приложением более приятным.
3. **Адаптивность:** Приложение должно корректно отображаться на разных устройствах (десктоп, планшет, мобильный).
4. **Удобство использования:** Интерфейс должен быть интуитивно понятным и удобным для использования.

Темы для повторения:

- **HTML:** Структура HTML документа, семантические теги, формы.
- **CSS:** Селекторы, стилизация элементов, Flexbox/Grid (для позиционирования элементов), анимации и переходы, медиа-запросы (для адаптивности).
- **JavaScript:** Работа с DOM (Document Object Model), обработка событий, манипуляция элементами на странице, (опционально: работа с localStorage).

Инструменты (рекомендации):

- **Форматирование кода:** Prettier
- **Проверка кода:** ESLint
- **Сборка проекта:** Webpack/Parcel (опционально)
- **Контроль версий:** Git
- **UI-фреймворки/библиотеки:** Tailwind CSS, Bootstrap (**опционально, с акцентом на кастомизацию, а не просто использование готовых компонентов**)

Критерии оценки:

- **Функциональность (1):** Все функции работают корректно и соответствуют требованиям.
- **Дизайн (1):** Визуальное оформление приложения, соответствие современным трендам, гармоничность цветовой палитры, типографика.
- **Качество кода (1):** Чистота и структурированность кода, использование комментариев, отсутствие ошибок.
- **Адаптивность (1):** Корректное отображение на разных устройствах.
- **Анимации (1):** Качество и плавность анимаций, соответствие общему стилю приложения.
- **Дополнительно (+1):** Использование localStorage для сохранения задач.

Что будет проверяться:

- Работоспособность всех функций.
- Соответствие дизайна требованиям.
- Качество и чистота кода.
- Адаптивность верстки.
- Наличие и качество анимаций.
- (Опционально) Сохранение данных в localStorage.

Примечания:

- Поощряется использование современных подходов и технологий.
- Креативность и оригинальность приветствуются.
- Код должен быть понятным и хорошо документированным.

Полезные ресурсы и примеры кода для задания "Дизайн-ориентированный To-Do List":

HTML & CSS:

- **MDN Web Docs (Mozilla Developer Network):** Лучший ресурс для изучения веб-технологий. Здесь вы найдете подробную документацию по HTML, CSS и JavaScript.
 - [HTML](#)
 - [CSS](#)
- **CSS-Tricks:** Сайт с множеством статей, туториалов и примеров по CSS. Полезен для изучения продвинутых техник CSS, включая анимации, Flexbox и Grid.
 - [CSS-Tricks](#)
- **Flexbox Froggy:** Интерактивная игра для изучения Flexbox.
 - [Flexbox Froggy](#)
- **Grid Garden:** Интерактивная игра для изучения CSS Grid.

- [Grid Garden](#)
- **Coolors:** Генератор цветовых палитр.
 - [Coolors](#)

JavaScript:

- **MDN Web Docs (Mozilla Developer Network):**
 - [JavaScript](#)
- **JavaScript.info:** Хороший ресурс для изучения JavaScript с нуля.
 - [JavaScript.info](#)
- **Примеры работы с DOM:**
 - [DOM Manipulation](#) (W3Schools)
- **Работа с localStorage:**
 - [Using the Web Storage API](#) (MDN)

Анимации:

- **Animate.css:** Библиотека готовых CSS анимаций.
 - [Animate.css](#)
- **Примеры CSS анимаций:**
 - [CSS Animations](#) (MDN)

UI/UX:

- **Laws of UX:** Основные принципы UX дизайна.
 - [Laws of UX](#)
- **Material Design:** Руководство по дизайну от Google.
 - [Material Design](#)
- **Dribbble/Behance:** Платформы для поиска вдохновения и примеров дизайна.
 - [Dribbble](#)
 - [Behance](#)

Примеры To-Do List:

Поищите в Google "to do list codepen" или "to do list github" для нахождения различных реализаций To-Do List. Обратите внимание, что эти примеры могут служить только для вдохновения и изучения отдельных техник. **Не копируйте код целиком.** Ваша задача — создать свой уникальный проект.

Сдача работы:

Работу необходимо сдать двумя способами:

1. **Ссылка на репозиторий Git:** Создайте репозиторий на GitHub и загрузите туда весь код вашего проекта. Убедитесь, что репозиторий публичный (или доступен преподавателю). Загрузите ссылку на файлообменник (например, Яндекс.Диск).
2. **Анимация работы приложения (GIF или видео):** Запишите короткий GIF или видео (например, в формате MP4), демонстрирующий работу вашего приложения. Покажите на видео все основные функции: добавление, удаление, редактирование, отметку задач как выполненных, а также анимации и адаптивность. Загрузите GIF/видео на файлообменник (например, Яндекс.Диск).

Соблюдение этих инструкций поможет оценить вашу работу быстро и эффективно.
Убедитесь, что все ссылки рабочие и доступны.

Это задание поможет вам закрепить знания по веб-разработке и развить навыки создания красивых и удобных пользовательских интерфейсов. Удачи!