**4 Реализация и тестирование программного обеспечения**

**4.1 Создание базы данных**

База данных представляет собой упорядоченный набор структурированной информации или данных, организованных для эффективного доступа, управления и обновления. База данных обычно хранится в электронном виде в компьютерной системе и используется для различных целей, таких как хранение, поиск, анализ и обработка данных.

Для разрабатываемого веб-приложения «TaskMaster» была создана своя система хранения данных. Хранилище LocalStorage в браузерах представляет собой механизм хранения данных в виде пар ключ-значение на стороне клиента. Данное хранилище позволяет веб-приложениям сохранять данные локально на устройстве пользователя без автоматической передачи на сервер. [10]

Была создана структура данных представляющая собой список tasks, куда по запросу помещаются словари с ключами task и state. После каждого запроса происходит обновление созданной структуры в LocalStorage. Структура хранения данных представлена на рисунке 4.1.

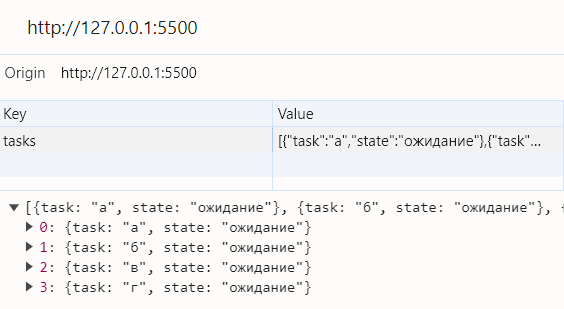


Рисунок 4.1 – Структура хранения данных

Источник: собственная разработка

Как итог, с помощью, сохраненных локально, данных после обновления или закрытия страницы в браузере данные, введенные пользователем не удаляются из системы.

**4.2 Реализация пользовательского интерфейса**

Основным требованием к разработке веб-приложения «TaskMaster» являлось создание приятного и простого интерфейса.

Пользовательский интерфейс веб-приложения был реализован с помощью языков HTML и CSS. С помощью HTML страница была структурирована, а с помощью CSS была выполнена стилизация веб-приложения. [8]

На рисунке 4.2 представлена главная страница веб-приложения «TaskMaster».



Рисунок 4.2 – Главная страница веб-приложения «TaskMaster»

Источник: собственная разработка

На главной страницы можно выделить 2 основных блока: центральный блок добавления заданий и боковая панель.

Центральный блок включает в себя строку ввода, кнопку ADD и задания, которые появляются после добавления. По нажатию кнопки ADD появляются блоки с заданиями. Была реализована анимация показа кнопок при наведении курсора мыши на блок с заданием. Анимация была реализована с помощью keyframes. Пример кода представлен на рисунке 4.3.Пример работы анимации представлен на рисунке 4.4.

Боковая панель по наведению мыши раскрывается, после чего можно посмотреть выполненные задания. Количество галочек на не раскрытой боковой панели равно количеству выполненных заданий. Код для реализации анимации раскрытия боковой панели представлен на рисунке 4.5. Пример работы анимации боковой панели представлен на рисунке 4.6.



Рисунок 4.3 – Реализация анимации блока задания

Источник: собственная разработка



Рисунок 4.4 – Анимация блока задания

Источник: собственная разработка

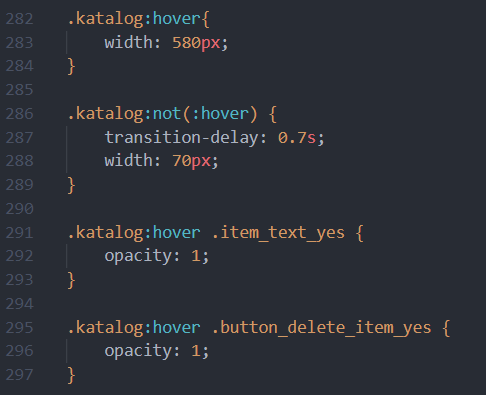


Рисунок 4.5 – Реализации анимации боковой панели

Источник: собственная разработка



Рисунок 4.6 – Анимация боковой панели

Источник: собственная разработка

Так же на главной странице присутствует кнопка CLEAR ALL, созданная для очистки блока с добавленными заданиями. [7]

Таким образом, был реализован приятный и простой интерфейс для разрабатываемого веб-приложения «TaskMaster».

**4.3 Реализация функциональных компонентов программного**

**обеспечения**

Функциональная часть веб-приложения была реализована с помощью языка JavaScript. Были реализованы следующие функции: добавления заданий в блок заданий, добавления заданий в список выполненных, удаление заданий поштучно или всех сразу, работа с хранилищем LocalStorage.

Первой была реализована функция добавления заданий в блок заданий. Пользователь вводит задание в строку ввода и нажимает на кнопку ADD, по которой срабатывает функция. Если строка оказывается пустой, будет выведено сообщение об ошибке. Данная функция создает блок для задания, добавляет в него введенный текст и добавляет в блок с заданиями, работает с другой функцией для настройки высоты основного блока, а так же сразу добавляет задание со статусом «ожидание» в LocalStorage. Пример кода для добавления блока задания представлен на рисунке 4.7. [6]



Рисунок 4.7 – Функция для добавления блока задания

Источник: собственная разработка

Второй из главных функций является функция для добавления задания в список выполненных. Пользователь нажимает на кнопку отметки задания на блоке с заданием, после чего задание удаляется из центрального блока и появляется в боковой панели. Функция ищет задание в списке tasks в словарях по ключу task, после чего статус задания в LocalStorage изменяется на «выполнено». Пример кода для добавления задания в список выполненных представлен на рисунке 4.8.



Рисунок 4.8 – Добавление задания в список выполненных

Источник: собственная разработка

Функции удаления задания или заданий реализованы почти по одному принципу: цикл foreach перебирает задания в созданной структуре данных, после чего удаляет задания из LocalStorage и из блоков, в которых находятся. Пример кода для удаления всех заданий из списка выполненных представлен на рисунке 4.9. Функция clear\_yes\_item() отвечающая за удаление из LocalStorage представлена на рисунке 4.10.

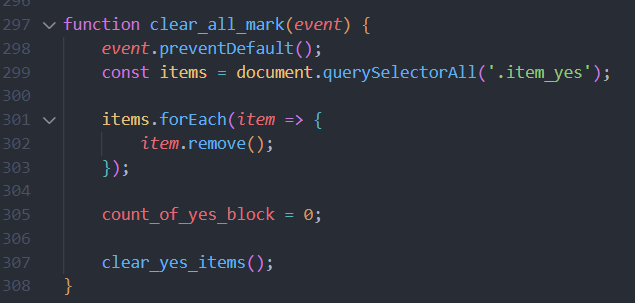


Рисунок 4.9 – Удаление всех выполненных заданий

Источник: собственная разработка

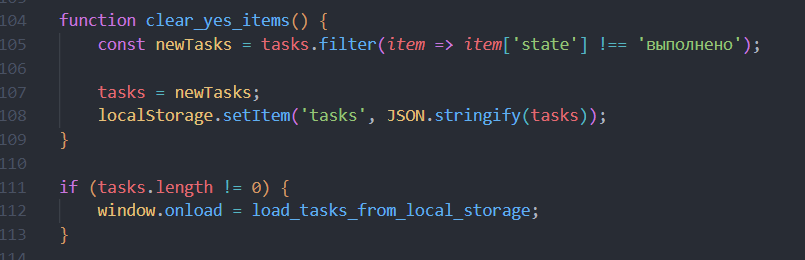


Рисунок 4.10 – Удаление выполненных заданий из LocalStorage

Источник: собственная разработка

При загрузке страницы с помощью LocalStorage данные, введенные пользователем, сохраняются и остаются неизменными. Принцип выгрузки данных из LocalStorage при загрузке страницы браузера реализован по принципам функций представленных на рисунках 4.7 и 4.8.

**4.4 Тестирование программного обеспечения**

**4.4.1 Модульное тестирование**

Модульное тестирование– это тип тестирования программного обеспечения, при котором тестируются отдельные модули или компоненты программного обеспечения. Целью данного вида тестирования является проверка каждой единицы программного кода на корректную работу. Данный вид тестирование выполняется разработчиками на этапе кодирования приложения. Модульные тесты изолируют часть кода и проверяют его работоспособность. Единицей для измерения может служить отдельная функция, метод, процедура, модуль или объект.

Для веб-приложения «TaskMaster» было произведено модульное

тестирование. Тесты представлены в ПРИЛОЖЕНИИ Г.

В процессе разработки тесты модульного тестирования успешно пройдены, ошибок не выявлено.

Готовое программное средство прошло функциональное тестирование на основе документа «Программа и методика испытаний» (ПРИЛОЖЕНИЕ Е).

Результаты функционального тестирования представлены в таблице 4.1.

Таблица 4.1 – Результаты функционального тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тестовый случай | Входные данные | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Нажатие на кнопку ADD | Корректные данные | Добавление задания в блок с заданиями | Добавление задания в блок с заданиями | Тест пройден |
| 2 | Нажатие на кнопку ADD | Некорректные данные | Показ уведомления об ошибке | Показ уведомления об ошибке | Тест пройден |
| 3 | Нажатие на кнопку отметки задания | Корректные данные | Удаление задания из блока с заданиями, появления задания в списке выполненных в боковой панели | Удаление задания из блока с заданиями, появления задания в списке выполненных в боковой панели | Тест пройден |
| 4 | Нажатие на кнопку удаления задания (блок заданий) | Корректные данные | Удаление задания из блока с заданиями | Удаление задания из блока с заданиями | Тест пройден |
| 5 | Нажатие на кнопку CLEAR ALL | Корректные данные | Удаление всех заданий из блока с заданиями | Удаление всех заданий из блока с заданиями | Тест пройден |

Продолжение таблицы 4.1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | Нажатие на кнопку удаления задания  (выполненные задания) | Корректные задания | Удаление задания из блока выполненных заданий | Удаление задания из блока выполненных заданий | Тест пройден |
| 7 | Нажатие на кнопку удаления всех заданий  (боковая панель) | Корректные данные | Удаление всех заданий из боковой панели | Удаление всех заданий из боковой панели | Тест пройден |

Источник: собственная разработка

В ходе функционального тестирования все тесты были успешно пройдены. Следовательно, веб-приложение «TaskMaster» разработано в соответствии с требованиями и работает корректно.

**4.5 Проект размещения программных компонентов**

**4.5.1 Диаграмма компонентов**

Диаграммы компонентов являются эффективным инструментом визуализации организации компонентов в системе и взаимосвязей. Данные модели позволяют создать высокоуровневое представление об архитектуре системы, выделить ключевые компоненты и продемонстрировать зависимости.

Основные цели диаграмм компонентов включают визуализацию структуры системы, выделение функциональности и облегчение понимания системы. Ключевые элементы таких диаграмм включают компоненты, интерфейсы, зависимости и артефакты. Преимущества использования диаграмм компонентов включают ясное представление структуры системы, облегчение коммуникации и помощь в проектировании и анализе.

Диаграмма компонентов для веб-приложения «TaskMaster» представлена на рисунке 4.11.

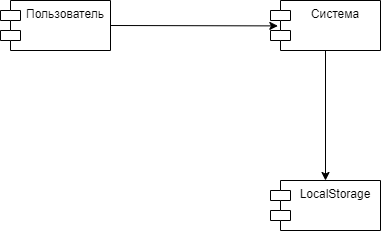


Рисунок 4.11 – Диаграмма компонентов

Источник: собственная разработка

На данной диаграмме компонентами выступают «Пользователь», «Система» и «LocalStorage».

Диаграммы компонентов играют важную роль в проектировании и разработке программного обеспечения, обеспечивая понимание структуры системы и помогая улучшить ее архитектуру.

**4.5.2 Диаграмма развертывания**

Диаграмма развертывания является типом UML-диаграммы, которая предназначена для визуализации архитектуры исполнения системы, включая различные узлы, такие как аппаратные или программные среды исполнения, а также промежуточное программное обеспечение, которое соединяет эти узлы. Модель позволяет представить физическое размещение компонентов системы и взаимосвязи на уровне исполнения.

Диаграммы развертывания широко используются для иллюстрации физической аппаратной и программной конфигурации системы. Через них можно понять, как система будет развернута на конкретном аппаратном обеспечении, какие узлы будут вовлечены и взаимодействовать друг с другом.

Основная цель диаграмм развертывания заключается в моделировании аппаратной топологии системы, что является отличием от других типов UML-диаграмм, которые скорее описывают логические компоненты и отношения. Поэтому диаграммы развертывания являются важным инструментом для понимания физической структуры и размещения компонентов в рамках системы.

Диаграмма развертывания для веб-приложения «TaskMaster» представлена на рисунке 4.12.

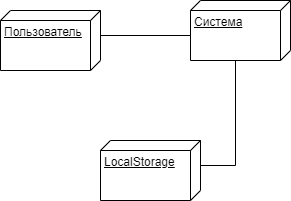


Рисунок 4.12 – Диаграмма развертывания

Источник: собственная разработка

Как итог, использование диаграмм развертывания облегчает коммуникацию между участниками проекта, помогает выявить потенциальные проблемы в распределении компонентов и оптимизировать аппаратные ресурсы.