**Проект «Канбан-доска»**

В этом проекте вы будете программировать одностраничное приложение, которое работает в браузере — канбан-доску. Приложение позволяет добавлять карточки задач, перемещать задачи между колонками, удалять выполненные задачи.

Интерфейс состоит из четырёх колонок для размещения задач по разным статусам, кнопок добавления и удаления задач, поля ввода для добавления названия задачи. В эталонной реализации используются нативные браузерные API.

В комплект входят:

* [Техническое задание проекта](#_Техническое_задание_проекта)
* Стартовый шаблон – *находится в скачанном архиве*.
* [Инструкция по запуску](#_Инструкция_по_запуску)
* [Полезные ссылки](#_Полезные_ссылки)

Вы можете начинать работу со *стартового шаблона*. В него включена готовая вёрстка проекта. Для начала работы вам нужно установить на компьютер удобный редактор кода и открыть в нём стартовый шаблон. У проекта такая структура:

* в файле index.html находится разметка проекта, её можно разбивать на отдельные компоненты;
* в папках css, fonts и img находятся необходимые для оформления статические ресурсы;
* в папке js есть заготовка для будущих скриптов — файл script.js — начинать работу можно в нём.

# Техническое задание проекта

## Добавление задачи

* Название карточки вводится в поле ввода, поле обязательно для заполнения.
* Чтобы добавить новую карточку с задачей на доску, надо ввести текст в поле и нажать кнопку «Добавить».
* Новая карточка добавляется в конец списка «Бэклог» с текстом, который был введён в поле.
* Поле ввода после добавления карточки в бэклог должно быть очищено.

## Редактирование задачи

* При наведении на карточку появляется иконка с карандашом.
* По клику на иконку ставится фокус в название задачи, появляется возможность редактировать текст.
* При нажатии на ENTER фокус с текста убирается, режим редактирования завершается, в описании сохраняется введённый текст.

## Перемещение задач

* Любую из задач можно перемещать внутри колонки или между колонками.
* При подъёме карточки, она исчезает из колонки, остальные карточки смещаются, занимая свободное место.
* При поднесении карточки к возможному месту дропа, задачи рядом раздвигаются, освобождая место для карточки. Появляется серый блочок с контурной рамкой, который показывает возможное расположение карточки.
* После дропа цвет на карточке слева становится такой же, как и у остальных в колонке.
* Если из колонки перенесли все карточки и она стала пустой, в колонке после названия появляется серый блочок с текстом «Перетащите карточку».

## Удаление задач

* По клику на кнопку «Очистить» все карточки из колонки «Корзина» удаляются без возможности восстановления.
* Если корзина пуста, кнопка «Очистить» блокируется и появляется серый блочок с информацией о том, что корзина сейчас пуста (тот же блок, что и в пустых колонках, только с текстом «Корзина пуста».

# Инструкция по запуску

Чтобы открыть рабочий проект в браузере на вашем компьютере, необходимо запустить в папке проекта локальный сервер.

Мы рекомендуем сделать это с помощью [Node.js](https://nodejs.org/). Для установки вы можете воспользоваться [инструкцией в нашем блоге](https://htmlacademy.ru/blog/boost/backend/installing-nodejs).

Далее для работы вам потребуется консоль. В операционных системах MacOS и Linux есть хорошая встроенная консоль — Terminal. А в Windows лучше всего будет установить консоль [cmder](https://cmder.net/" \t "_blank) отдельно — в блоге есть [инструкция](https://htmlacademy.ru/blog/boost/tools/installing-the-console-on-windows), как это сделать.

После установки консоли можно переходить к запуску проекта. Для этого вам нужно перейти в консоли в папку с эталонным решением проекта и затем запустить команду npx http-server. После успешного запуска, не закрывая консоль, откройте в браузере проект по адресу http://127.0.0.1:8080.

# Полезные ссылки

## Подготовка

* [Обзор редакторов для верстальщика](https://htmlacademy.ru/blog/40-editors-for-the-coders)
* [Что на самом деле происходит, когда пользователь вбивает в браузер адрес google.com](https://htmlacademy.ru/blog/js/brauzer-google" \t "_blank)
* [Принципы работы современных веб-браузеров](https://www.html5rocks.com/ru/tutorials/internals/howbrowserswork/)
* [Софт для программиста на каждый день](https://htmlacademy.ru/blog/boost/tools/soft-for-coders)
* [Введение в Chrome DevTools. Панель Elements](https://htmlacademy.ru/blog/boost/tools/chrome-devtools-1)
* [Введение в Chrome DevTools. Console, Sources, Network](https://htmlacademy.ru/blog/boost/tools/chrome-devtools-2)
* [Полезные команды в консоли Windows](https://htmlacademy.ru/blog/boost/tools/windows-command-line)

## Работа с DOM

* [Что такое DOM-дерево?](https://learn.javascript.ru/dom-nodes)
* [Улучшаем опыт взаимодействия с формами](http://simonenko.su/38146501854/improving-ux-for-web-form)
* [Туториал. Список задач с drag & drop](https://htmlacademy.ru/blog/boost/frontend/drag-and-drop)
* [Живые и неживые коллекции в JavaScript](https://htmlacademy.ru/blog/boost/frontend/collections-js)
* [Зачем нужен метод preventDefault](https://htmlacademy.ru/blog/boost/frontend/prevent-default)

## Функции

* [Введение в колбэк-функции в JavaScript](https://htmlacademy.ru/blog/boost/frontend/callback-functions)
* [Замыкания в JavaScript](https://htmlacademy.ru/blog/boost/frontend/lets-learn-javascript-closures)

## Паттерны проектирования

* [Что такое паттерн проектирования](https://refactoring.guru/ru/design-patterns/what-is-pattern)
* [Паттерны проектирования в JavaScript (англ.)](https://addyosmani.com/resources/essentialjsdesignpatterns/book/#designpatternsjavascript)
* [Паттерн Event Emitter в JavaScript](https://medium.com/@an_parubets/pattern-event-emitter-js-9378aa082e86)
* [Одностороннее связывание в JavaScript (англ.)](https://dev.to/phoinixi/one-way-data-binding-in-vanilla-js-poc-4dj7)
* [Двустороннее связывание в JavaScript (англ.)](https://foxypanda.me/understanding-two-way-data-binding-in-js/)
* [Паттерн Event Emitter для двунаправленного общения: примеры и антипримеры](http://dmenshikov.com/2017-08-02-bidirectional-event-emitting/)
* [Паттерн наблюдатель](https://refactoring.guru/ru/design-patterns/observer)
* [Паттерн наблюдатель в JavaScript](https://monsterlessons.com/project/lessons/observer-pattern-v-javascript)
* [Паттерн Посредник (он же Controller)](https://refactoring.guru/ru/design-patterns/mediator)
* [Паттерны для новичков: MVC vs MVP vs MVVM](https://habr.com/ru/post/215605/)

## Публикация в Интернете

* [Как начать работу на GitHub](https://htmlacademy.notion.site/GitHub-fa1a2ea00a6d451fa6c29ff449f0e139)