Учебный проект «Менеджер задач»

Техническое задание

О проекте

«Менеджер задач» помогает пользователю организовывать и контролировать выполнение задач. Минималистичный интерфейс приложения не позволит пользователю отвлекаться по пустякам и сфокусирует внимание на главном — задачах.

1. Описание функциональности

Приложение состоит из двух экранов: «Задачи» и «Статистика».

1.1 Задачи

- После загрузки приложения в списке задач отображается не более 8 карточек задач.

- Показ оставшихся карточек задач осуществляется с помощью кнопки «Load more». При нажатии в список добавляются очередные 8 задач. Либо оставшиеся задачи, если их количество меньше 8.

- Любое изменение фильтра или сортировки, а так же переключение на экран статистики и обратно, сбрасывает счётчик показанных задач и отсчёт начинается заново.

- Если все задачи показаны, кнопка «Load more» скрывается.

- Если все задачи выполнены, то вместо задач отображается текст: `Click «ADD NEW TASK» in menu to create your first task`.

- Нажатие на кнопку «Tasks» скрывает статистику и отображает список задач.

- Задачи в списке доступны только для просмотра. Для редактирования задачи пользователь нажимает на кнопку «Edit».

- В карточке пользователю доступна кнопка «Favorites». При нажатии на кнопку, задача добавляется в избранное. Повторное нажатие приводит к удалению из избранного. При этом сразу же происходит пересчёт количества задач в избранном (см. «Фильтры»).

- В карточке пользователю доступна кнопка «Archive». При нажатии на кнопку, задача помечается выполненной. При этом к ней сразу же применяется действие фильтров и сразу же происходит пересчёт количества задач в избранном (см. «Фильтры»).

- Повторное нажатие на кнопку «Archive» помечает задачу, как невыполненную. Это действие возможно только при активации фильтра «Archive» (см. «Фильтры»).

- Кнопки «Favorites» и «Archive» отражает состояние задачи с помощью установки/удаления css-класса `card__btn--disabled`.

- Просроченным задачам добавляется класс `card--deadline`. Под просроченной задачей подразумевается задача с установленной датой исполнения меньше текущей даты.

1.2 Создание задачи

- Новая задача создаётся нажатием на кнопку «Add new task». После нажатия пользователь видит форму создания новой задачи, которую ему необходимо заполнить.

- Форма создания новой задачи всегда появляется первой в списке задач. Форма создания «сдвигает» остальные задачи, а не занимает место одной из них. То есть в списке может быть 8 задач плюс форма создания и так далее.

- Если в момент нажатия на кнопку «Add new task» был выбран фильтр или применена сортировка, то они сбрасываются на состояния «All» и «SORT BY DEFAULT» соответственно.

- Нажатие на кнопку `Esc` приводит к закрытию формы создания без сохранения изменений.

- Менеджер задач позволяет создавать задачи двух видов:

- задачи без повторения и регулярные задачи — повторяемые по определённым дням недели;
- для регулярных задач обязательно должен быть выбран день (дни) недели для повторения и скрыто поле «Дата исполнения»;
- в случае несоблюдения условий задача не может быть сохранена — кнопка «Save» заблокирована;

- В форме создания пользователь заполняет поля:
  - Описание задачи:
    - минимальная длина 1 символ;
    - максимальная длина 140 символов;
  - Дата исполнения:
    - поле может быть не заполнено;
    - дата: число и месяц полностью и время в 24-часовом формате (например «23 September 16:15»);
    - скрыто при активированном флаге повторения;
    - выбор времени и даты осуществляется с помощью библиотеки `flatpickr.js`;
  - Флаг повторения:
    - определяет, является ли задача регулярной;
    - значение по умолчанию `NO`;
    - изменение флага повторения сразу отображается в шапке карточки задачи (до сохранения) — цветная полоска становится волнистой, а не прямой;
  - Дни недели повтора задачи:

        - по умолчанию ни один день не выбран;
        - отображено, если установлен флаг повторения. В противном случае скрыто;
    - Цветовая категория:
        - выбор цветовой категории меняет цвет в шапке задачи путём добавления одного из css-классов: `card--black`, `card--green`, `card--yellow`, `card--pink`, `card--blue`;
        - обновление цветовой категории в шапке происходит после сохранения изменений;
        - для задачи может быть установлена только одна цветовая категория;
        - значение по умолчанию: `card--black`;
- Скрытые поля интерпретируются как пустые при сохранении задачи.

- Введённые пользователем данные экранируются.

1.3 Редактирование задачи

- Для редактирования задачи пользователь нажимает кнопку «Edit». В этот момент карточка задачи заменяется на форму редактирования задачи.

- В форме редактирования представлены все поля, которые пользователь заполняет при создании новой задачи (см. раздел «Создание задачи»). Правила их поведения сохраняются.

- В качестве значений по умолчанию используются фактические данные редактируемой задачи.

- Если пользователь не сохранил изменения, то все изменения, включая цветовую категорию, флаг избранного, флаг выполнения сбрасываются после закрытия формы редактирования.

- Одновременно в режиме редактирования или создания может находиться только одна задача.

- Если у пользователя открыта задача для редактирования/создания и пользователь открывает другую задачу для редактирования/создания, то первая задача переходит в режим просмотра (пропадает в случае создания). Несохраненные данные при этом теряются.

- Для сохранения внесённых изменений во время редактирования пользователь нажимает кнопку «Save». Форма редактирования закрывается, пользователь видит задачу в режиме просмотра.

- Нажатие на кнопку `Esc` приводит к закрытию формы редактирования без сохранения изменений. Пользователь видит задачу в режиме просмотра.

1.4 Фильтры

- Для быстрого поиска задач в приложении предусмотрено несколько фильтров:
  - All — все невыполненные задачи.

    - Overdue — просроченные задачи.
    - Today — задачи на сегодня.
    - Favorites — избранные задачи.
    - Repeating — повторяющиеся задачи.
    - Archive — выполненные задачи.
- При активации фильтра отображаются только те задачи, которые ему удовлетворяют.

- Рядом с названием фильтра отображается число — количество удовлетворяющих этому фильтру задач. Оно обновляется при создании, редактировании, удалении, архивации или добавлении задачи в избранное.

- Если задачи удовлетворяющие фильтру отсутствуют, то кнопке установки этого фильтра добавляется атрибут `disabled`.

- Одновременно может быть активирован только один фильтр.

- Выполненные (архивные) задачи отображаются только при активации фильтра «Archive».

1.5 Сортировка

- Пользователю доступна возможность сортировки задач по дате исполнения в обоих направлениях: от меньшей к большему и наоборот. Для этого пользователю нужно кликнуть по ссылкам «SORT BY DATE up» или

«SORT BY DATE down» соответственно.

- Для отмены сортировки и возвращению к исходному порядку нужно кликнуть по ссылке «SORT BY DEFAULT».

- При смене фильтра или переключении с экрана задач на экран статистики и обратно сортировка сбрасывается на состояние «SORT BY DEFAULT».

1.7 Статистика

- Нажатие на кнопку «Statistic» скрывает список задач и отображает статистику.

- На экране «Статистика» представлен график выполненных задач за выбранный период (количество выполненных задач за день периода) и круговая диаграмма «Выполнено задач в разрезе цветовой категории» (Done by: Colors).

- При открытии статистики в поле ввода периода установлен временной интервал равный прошлой неделе, начиная с сегодняшнего дня. Например, если сегодня 15 марта, то интервал должен быть `09.03–15.03`.

- При открытии экрана «Статистика» автоматически отрисовывается график выполненных задач и круговые диаграммы с помощью библиотеки `chart.js`.

- При изменении периода происходит обновление графика и диаграмм.

- Для упрощения ввода дат допускается применение компонента `flatpickr.js`.

- Фильтры на графики и статистику никак не влияют.

1.8 Взаимодействие с сервером

- Сервер расположен по адресу `https://11.ecmascript.pages.academy/task-manager/`.
- Каждый запрос к серверу должен содержать заголовок `Authorization` со значением `Basic ${случайная строка}`. Например, `Basic er883jdzbdw`. Случайная строка формируется однократно при старте приложения.

1.8.1 Ресурсы

Структуры данных:

- LocalTask:

```
{
  "color": "yellow",
  "description": "find money for travel",
  "due_date": "2019-07-19T09:52:05.173Z",
  "is_archived": false,
  "is_favorite": false,
  "repeating_days": {
    "mo": false,
    "tu": false,
    "we": false,
```

```
    "th": false,
    "fr": false,
    "sa": false,
    "su": false
  }
}
```

- Task:

```
{
  "id": "0",
  "color": "yellow",
  "description": "find money for travel",
  "due_date": "2019-07-19T09:52:05.173Z",
  "is_archived": false,
  "is_favorite": false,
  "repeating_days": {
    "mo": false,
    "tu": false,
    "we": false,
    "th": false,
    "fr": false,
    "sa": false,
    "su": false
  }
}
```

- Task.due_date — строка new `Date().toISOString()` или `null`, если срок выполнения не установлен;

- AuthorizationError:

```
{
  "error": 401,
  "message": "Header Authorization is not correct"
```

```
}
```

- NotFoundError:

```
{
  "error": 404,
  "message": "Not found"
}
```

Список задач `/tasks`

GET /tasks

Получение всех созданных задач.

- Коды ответов:
  - 200 OK;
  - 401 Unauthorized.

Пример:

- Request:
  - URL: GET /tasks;
  - Headers: Authorization: Basic kTy9gIdsz2317rD.
- Response:
  - Status: 200 OK;
  - Body: массив, содержащий элементы типа `Task`.
- Request:
  - URL: GET /tasks.
- Response:
  - Status: 401 Unauthorized;
  - Body: структура `AuthorizationError`.

POST /tasks

Создание новой задачи.

- Коды ответов:
    - 200 OK;
    - 401 Unauthorized.

Пример:

- Request
    - URL: POST /tasks;
    - Headers: Authorization: Basic kTy9gIdsz2317rD;
    - Body: структура `LocalTask`.
- Response:
    - Status: 200 OK;
    - Body: структура `Task`.

PUT /tasks/: id

Обновление существующей задачи.

- Коды ответов:
    - 200 OK;
    - 401 Unauthorized;
    - 404 Not found.

Пример:

- Request:
    - URL: POST /tasks/7;
    - Headers: Authorization: Basic kTy9gIdsz2317rD;
    - Body: структура данных вида `Task`.
- Response:
    - Status: 200 OK;

- Body: структура данных вида `Task`.

DELETE /tasks/: id

Удаление существующей задачи.

- Коды ответов:
  - 200 OK;
  - 401 Unauthorized;
  - 404 Not found.

Пример:

- Request:
  - URL: DELETE /tasks/7;
  - Headers: Authorization: Basic kTy9gIdsz2317rD;
- Response:
  - Status: 200 OK.

Синхронизация с сервером `/tasks/sync`

Этот метод потребуется для реализации дополнительного задания лекции 9.

Пример:

- Request:
  - URL: POST /tasks/sync;
  - Headers: Authorization: Basic kTy9gIdsz2317rD;
  - Body: `[$Task$, $Task$]`;
- Response:
  - Status: 200 OK;
  - Body:

```
{
  "created": [$Task$, $Task$],
  "updated": [$Task$, $Task$],
  "deleted": [$Task$, $Task$]
}
```

1.9 Обратная связь в интерфейсе

- На время загрузки данных вместо списка задач нужно вывести сообщение `Loading...`.

- В момент отправки запроса на создание/удаление/изменение задачи форма блокируется от внесения изменений. Разблокировка формы происходит после завершения выполнения запроса (неважно, успешно выполнен запрос или нет).

- При создании и редактировании задачи нажатие на кнопки «Save», «Delete» формирует запрос к серверу на добавление/изменение/удаление данных. На время выполнения запроса поля формы и кнопки блокируются, текст заголовка кнопки изменяется:
  - `Save` -> `Saving...`
  - `Delete` -> `Deleting...`
- В режиме просмотра задачи нажатие на кнопки «Favorites», «Archive» формирует запрос к серверу на изменение данных. На время выполнения запроса кнопки блокируются, текст заголовка кнопки изменяется:
  - `Favorites` -> `Favoriting...`
  - `Archive` -> `Archiving...`
- Заголовки кнопок возвращаются в исходный вид после выполнения запроса (неважно, успешно выполнен

запрос или нет).

- Обновление элементов в DOM (удаление задачи, обновление задачи и так далее) происходит после успешного выполнения запроса к серверу.

- Если запрос не удалось выполнить (сервер недоступен, произошла ошибка), форма редактирования/создания остаётся открытой, к ней применяется эффект «покачивание головой».

2. Дополнительно

2.1 Offline режим

- Реализуйте в приложении поддержку оффлайн режима.

- Приложение должно кэшировать статические ресурсы (разметку, стили, изображения) с помощью `ServiceWorker` и должно уметь их отображать без использования сети интернет.

- Приложение должно сохранять весь пользовательский функционал без доступа к сети интернет: создание/удаление/редактирование задач, фильтрация, сортировка и так далее.

- При появлении доступа к сети интернет все изменения должны синхронизироваться с сервером (см.

«Взаимодействие с сервером»).

3. Разное

- В зависимости от состояния, некоторым элементам управления применяются соответствующие классы оформления. Например, активный фильтр, активный экран, кнопки «Favorites», «Archive» и так далее. Примеры доступны в директории с вёрсткой (`markup`).