

## Тестовое задание

Необходимо создать упрощенную версию планировщика задач с помощью React.js. Тестовое задание состоит из нескольких обязательных задач и нескольких дополнительных. Вы можете не делать дополнительные задачи или сделать некоторые из них, однако, в сопроводительной записке укажите пути к решению оставшихся задач.

## Внешний вид приложения

Для упрощения представлены мокапы, показывающие основную структуру страниц. В случае необходимости вы можете поправить внешний вид элементов так, как сочтете нужным, однако основная логика страниц должна остаться неизменной.

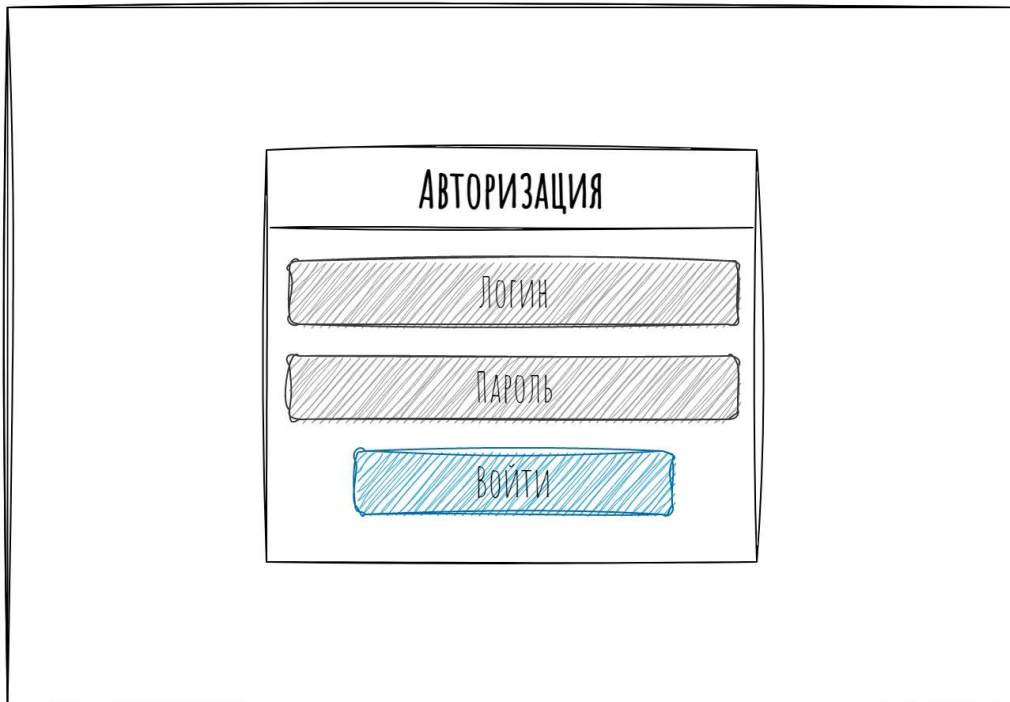
При реализации данного приложения вы можете использовать любую сетку или не использовать её вовсе.

Вы можете воспользоваться библиотеками готовых компонентов (например, Material-UI, ant.design или что-то другое).

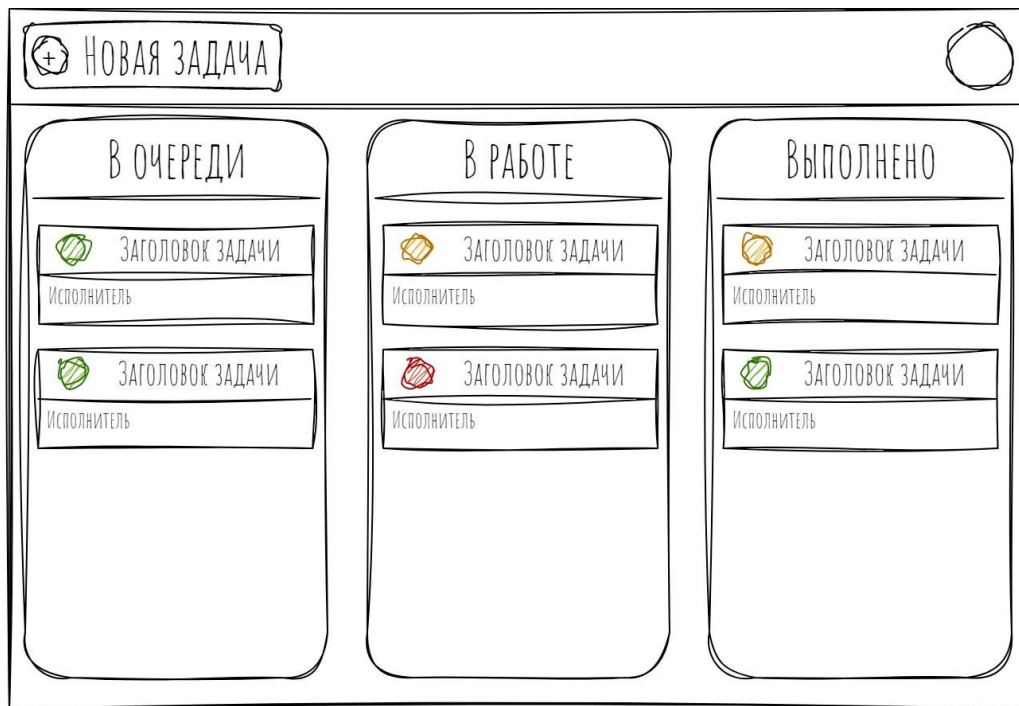
Приложение должно быть респонсив от 320px.

Разрабатываемое приложение состоит из трех страниц:

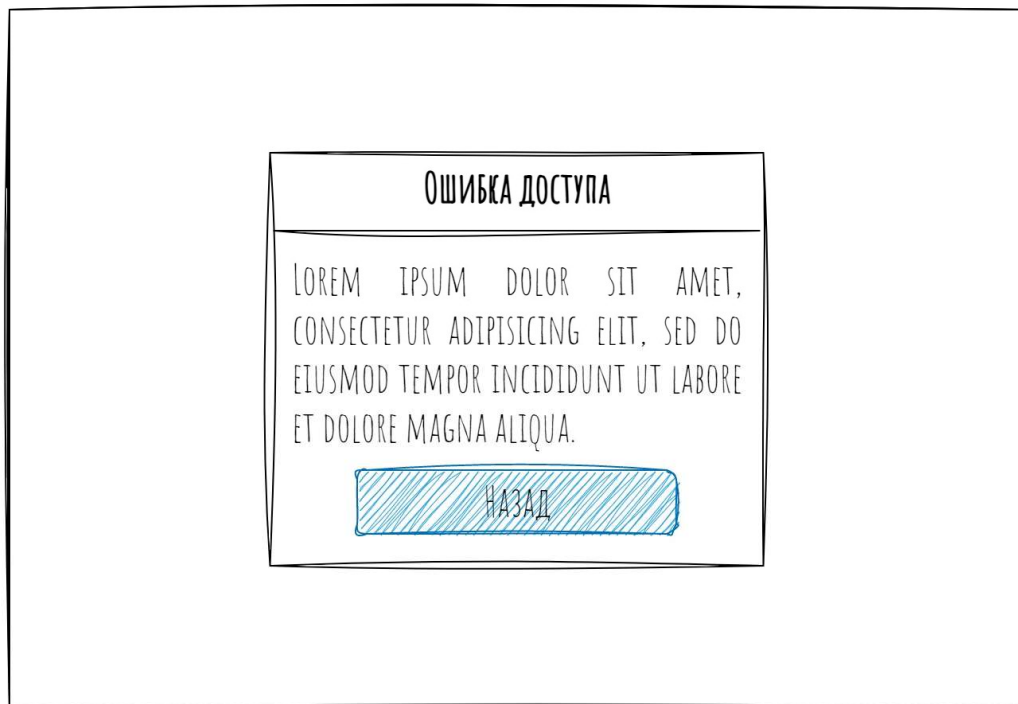
1. Экран логина (/)



2. Рабочий экран (/tasks)



### 3. Экран ошибки доступа (/error)



## Авторизация

В разрабатываемом приложении должна быть реализована система авторизации. Использовать БД для хранения данных не нужно.

При авторизации пользователю должен передаваться случайно генерируемый ключ, сохраняемый в localStorage, его валидность проверять не нужно, только его наличие.

Необходимо обеспечить доступ из-под учетной записи admin admin.

В случае попытке доступа к странице /tasks без ключа, пользователя должно редиректить на страницу /error

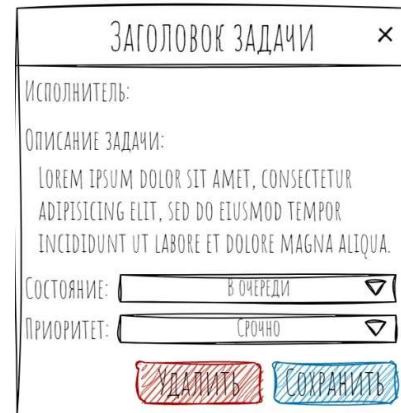
## Логика работы приложения

На основном экране должны быть отображены три списка задач по их статусу. Данные для заполнения можно взять из приложенного файла data.txt



Каждая задача должна представлять из себя карточку с названием задачи, её исполнителем и с приоритетом выполнения в виде графического элемента.

При клике на карточку конкретной задачи должно появляться модальное окно с подробной информацией по задаче возможностями для её удаления и смены статуса и приоритета. Все элементы должны быть активны и выполнять свои функции. При нажатии на кнопках «сохранить» или «удалить» советующие изменения должны происходить на основном экране



При нажатии на кнопку «новая задача» должно появляться модальное окно с необходимыми полями для создания новой задачи. При нажатии на кнопку «сохранить» соответствующие изменения должны отображаться на основной странице.

Взаимосвязь между компонентами организуйте на свое усмотрение.

### (Дополнительно) Фильтрация списков

Необходимо организовать фильтры для списков задач. Фильтровать нужно по дате создания задачи, по убыванию или возрастанию.

### (Дополнительно) Вывод приложения в один файл

Выводить приложение в один файл при билде. В итоге должен получаться один файл (например, index.html) который будет включать в себя все скрипты и css.

### (Дополнительно) Обеспечить поддержку IE10

Предусмотреть поддержку по мимо современных браузеров и поддержку IE10+

## Результат работы

В итоге результатом должно стать:

- проект, загруженный на github, или архив с git'ом, в том числе build
- сопроводительное письмо, может быть выполнено в виде README. Не должно быть большим.