

Задание на экзамен

ДАННОЕ ЗАДАНИЕ ИДЕЙНО ОСНОВАНО НА ПРЕДЫДУЩИХ РАБОТАХ. МЫ УЖЕ ИСПОЛЬЗОВАЛИ `useState` для СОХРАНЕНИЯ СОСТОЯНИЯ, `useEffect` для ЗАПРОСОВ НА СЕРВЕР, `react-router-dom` для СОЗДАНИЯ НАВИГАЦИИ, `styled-components`/ГЛОБАЛЬНЫЕ СТИЛИ/`CSS-IN-JS`/CSS-МОДУЛИ для СТИЛИЗАЦИИ

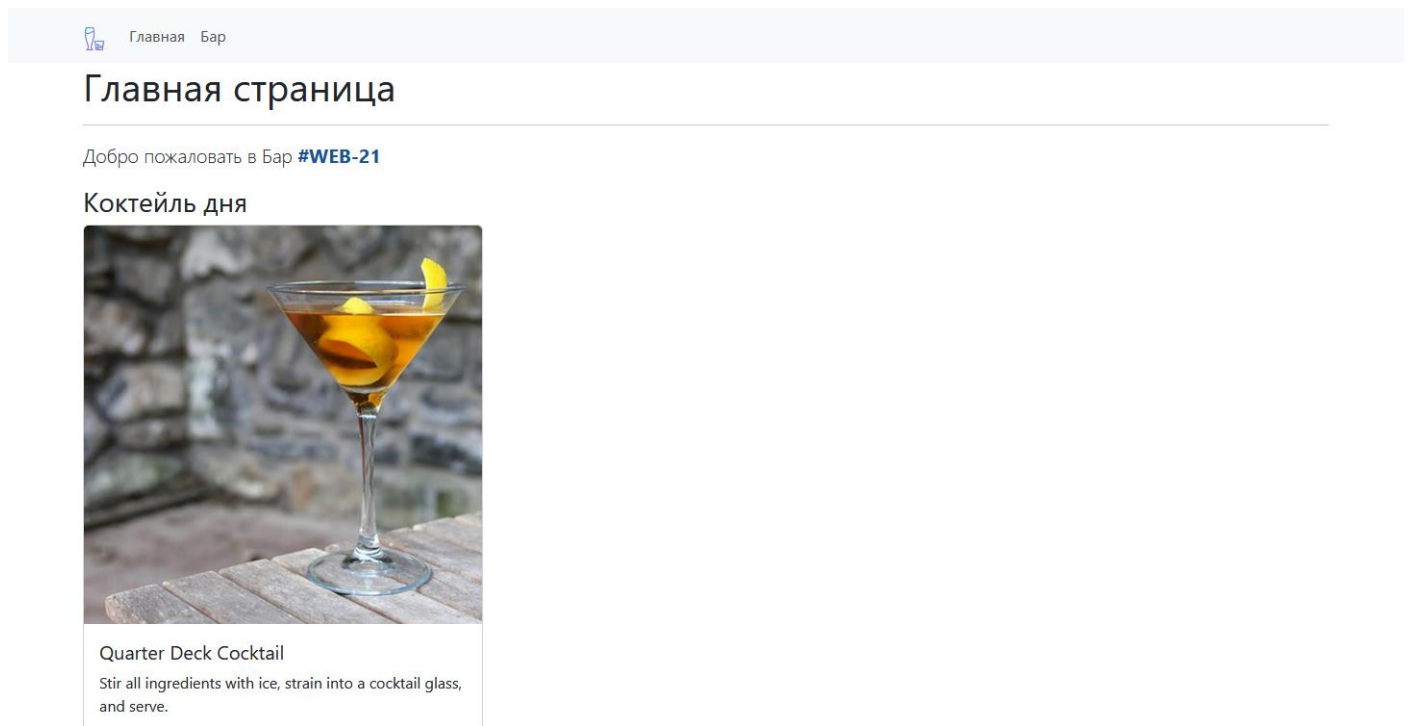
Создать приложение со следующими функциями:

1. Несколько страниц (не менее 2 страниц) с адресами “/” и “/bar”
2. На каждой страницу отобразить заголовок **h1**
3. На главной странице выводим один коктейль с адреса <https://www.thecocktaildb.com/api/json/v1/1/random.php>
4. На второй странице выводим список коктейлей <https://www.thecocktaildb.com/api/json/v1/1/search.php?f=a>
5. Использование стилизации – по желанию
6. Добавление строки меню – по желанию

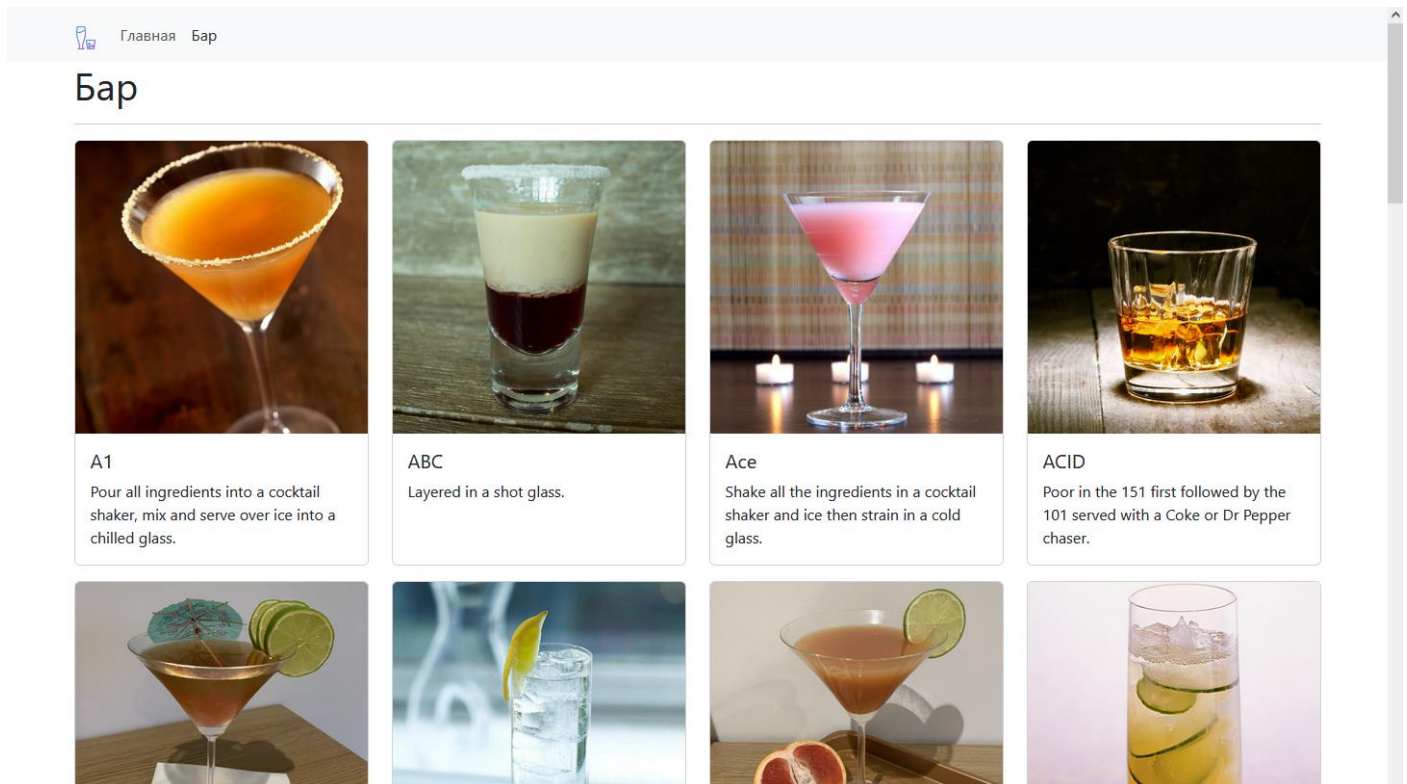
При открытии главной страницы – сразу выводим один коктейль

При открытии второй страницы – сразу выводим список коктейлей

Пример результата (страница 1):



Пример результата (страница 2):



Примерный план работы:

1. Создаем приложение либо в браузере (**codesandbox**) либо локально (**npx create-react-app exam-react**).
2. Добавляем следующие пакеты:
 - a. **react-router-dom** (для навигации),
 - b. **axios** (для запросов - опционально, можно использовать **fetch**),
 - c. **styled-components** / **sass** / **emotion** / **glamorous** / **глобальные стили** (по желанию)
 - d. Другие библиотеки по желанию
3. Создаем папку для страниц и для компонентов
4. Добавляем роутинг в файл **App.js** (**BrowserRouter**, **Routes**, **Route**)
5. Поключаем глобальные стили (если планируется их использование) в файле **index.css** или **styles.css** с помощью **@import url(адрес стилей);**
6. Создаем файлы страниц и описываем внешний вид страниц (добавляем параграфы, заголовки и т.д.)
7. Создаем файл с шаблоном вывода одного коктейля (шаблон карточки/строки/другой вариант)
8. Создаем файл с меню (если планируется использовать меню)
9. В файл главной страницы добавляем **useState** и **useEffect** для запроса и сохранения информации с сервера (см пункт 3 из постановки задачи)
10. В файл второй страницы добавляем **useState** и **useEffect** для запроса и сохранения информации с сервера (см пункт 4 из постановки задачи)

11. Копируем в браузер строку запроса для пункта 3 и/или 4 из постановки задачи и смотрим структуру возвращаемых данных
12. С помощью созданного в п.7 компонента выводим информацию на обеих страницах с учетом той структуры, которую узнали в п.11

Пояснения:

1. Картинки для оформления можно взять с сайта (чтобы не скачивать картинки, можно посмотреть адрес на интересующую картинку через инструменты разработчика)
<https://www.flaticon.com>
2. При желании можно использовать библиотеку стилей. Например
<https://getbootstrap.com/>
3. Сайт с API
<https://www.thecocktaildb.com/api.php>
4. Гайд по flexbox
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>