

УДК 004.9

ВЕБ-ПРИЛОЖЕНИЕ «КНИГА РЕЦЕПТОВ»

Лебедев Е.И. студент гр. Пиб-191, 2 курс
Спицын В.С. студент гр. Пиб-191, 2 курс
Научный руководитель: Киреева К.А., ассистент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева
г. Кемерово

Веб-приложение – это приложение, логика которого распределена между клиентом и сервером. Клиентом выступает видимая пользователю часть программы, с которой он взаимодействует, а сервером является компонент вычислительной системы, выполняющий сервисные функции по запросу клиента, предоставляя ему доступ к определённым ресурсам и услугам.

На сегодняшний день веб-приложения пользуются огромным спросом. Поскольку они хранят в себе много полезной информации, которая находится в открытом доступе для любых пользователей, кроме того веб-приложения позволяют максимально быстро донести информацию широкому кругу лиц, организовать обратную связь с клиентами, а также за счет рекламы привлечь покупателей и клиентов [1].

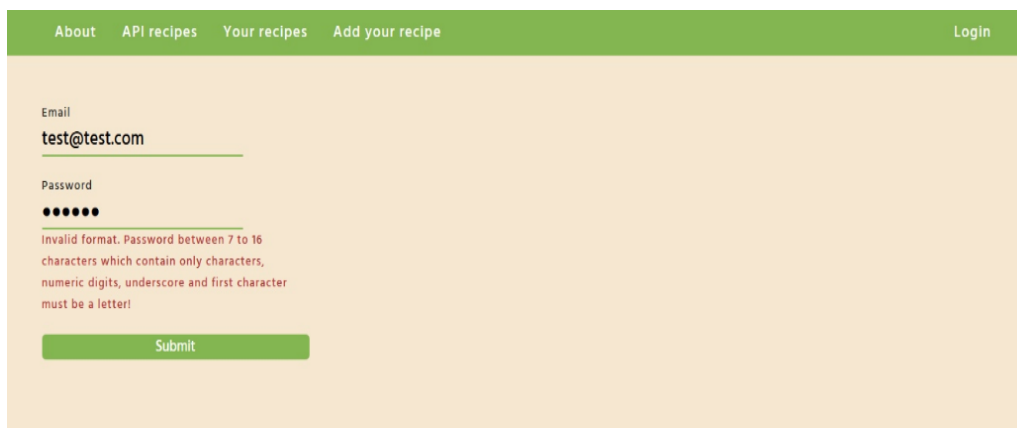
Данное веб-приложение создано для того, чтобы люди могли легко и комфортно находить рецепты блюд, а также создавать собственные и делиться ими с другими пользователями сайта.

При входе на сайт пользователю будет предложено окно авторизации (рис.1).



Рисунок 1 – Окно авторизации

Если пользователь первый раз заходит на сайт ему необходимо зарегистрироваться. Для регистрации пользователю нужно ввести свой email и пароль в соответствующие поля. Для каждого поля реализованы проверки со стороны клиентской части приложения. При вводе некорректных данных пользователю выводится сообщение с информацией об ошибке (рис. 2)



The screenshot shows a login form with a green header containing navigation links: 'About', 'API recipes', 'Your recipes', 'Add your recipe', and a 'Login' button. The form fields are: 'Email' with the value 'test@test.com', and 'Password' with masked characters. A red error message is displayed below the password field: 'Invalid format. Password between 7 to 16 characters which contain only characters, numeric digits, underscore and first character must be a letter!'. A green 'Submit' button is at the bottom.

Рисунок 2 – Пример проверки

После авторизованного входа в приложение будет доступен весь функционал данного сайта, а именно поиск готовых рецептов (рис. 3) и добавление собственных рецептов (рис. 4).

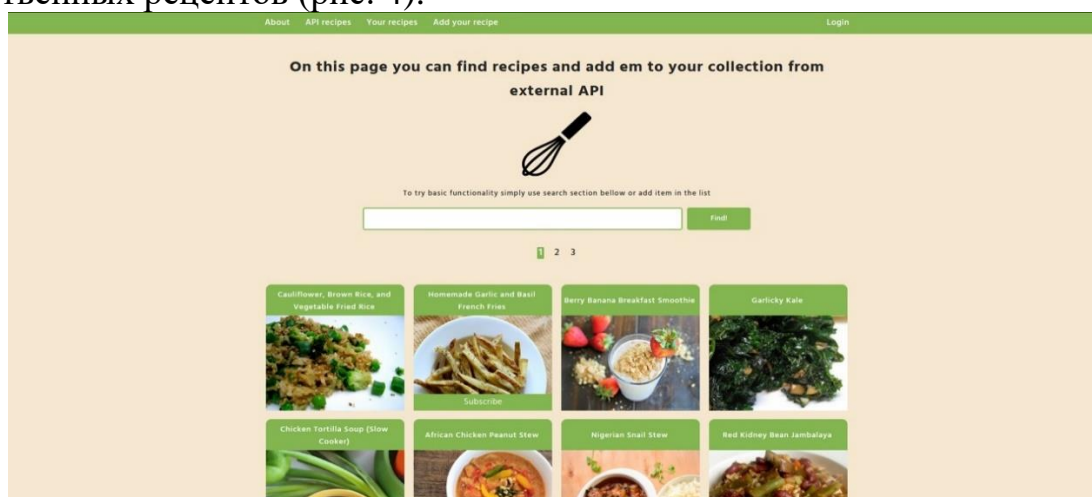
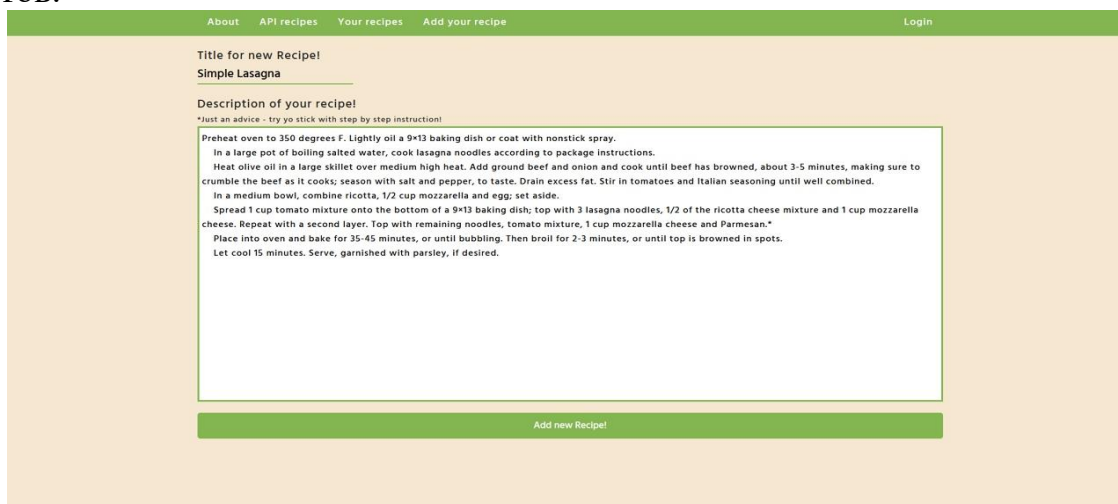


Рисунок 3 – Поиск рецептов

На странице поиска пользователь может найти интересующий его рецепт. Для его поиска в строку поиска необходимо написать название рецепта и нажать на кнопку «Find!». После нахождения нужного рецепта пользователь может подписаться на него, после чего он будет добавлен в его коллекцию рецептов.



The screenshot shows a form for adding a new recipe. The header is green with navigation links. The form has a title field 'Title for new Recipe!' with the value 'Simple Lasagna'. Below it is a 'Description of your recipe!' field with a text area containing detailed instructions for making lasagna. A green 'Add new Recipe!' button is at the bottom.

Рисунок 4 – Добавление собственного рецепта

На странице добавления пользователь должен вписать название рецепта в поле «Title of the new Recipe!», затем в поле «Description of your recipe!» он вводит произвольное описание для своего блюда (рис. 4).

После добавления можно посмотреть рецепт на вкладке «Your recipes» (рис.5). На этой же вкладке будут рецепты, на которые подписался пользователь.

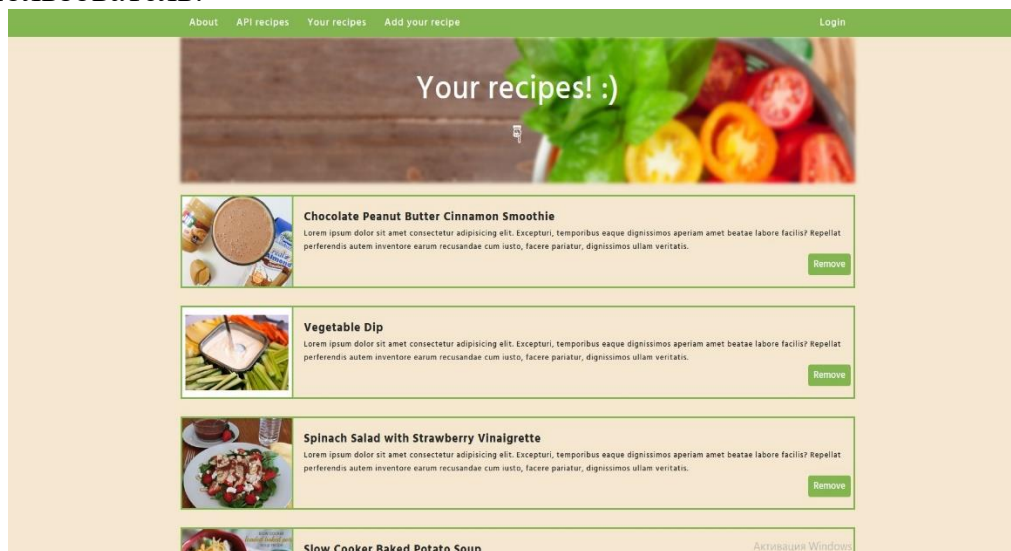


Рисунок 5 – Просмотр добавленных рецептов

Итак, благодаря данному веб-приложению пользователи могут легко и удобно находить интересующие рецепты, а также добавлять свои собственные.

Для создания веб-приложения использовался язык программирования JavaScript, на клиентской части использовались React и Redux, серверная часть работает с базой данных MongoDB на платформе Node.js, с использованием СУБД Mongoose, также применяется внешний API - spoonacular API, для загрузки рецептов. Для реализации backend части приложения используется Express - надстройка над Node.js, обладающая всеми необходимыми инструментами для создания современного серверного кода. Для хранения API ключа используется *.env файл, доступ к которому осуществляется через переменные среды. Для осуществления http/https запросов применяется axios. Для оформления стилей приложения используется библиотека Styled Components, благодаря которой легче интегрировать CSS стили в React проекты, с их компонентной архитектурой. Для валидации значений в формах используется Formik - библиотека, обладающая удобным и достаточно мощным набором функционала. Для упрощения разработки используется Nodemon - npm-пакет, позволяющий в реальном времени перезапускать сервер, если в самом коде возникли изменения (используется исключительно в режиме разработчика, в основное решение приложения этот пакет не идёт).

Список литературы:

1. Статья на тему «Основные возможности web-приложений» [Электронный ресурс] – Режим доступа: <https://infourok.ru/statya-na-temu-osnovnye-vozmozhnosti-sovremennyh-web-prilozhenij-4079046.html>, свободный (дата обращения: 20.03.2021).