Проектна задача по предметот Напреден Веб Дизајн

Тема: Food Recipe App

Изработила: Мартина Георгиева со бр.Индекс 203008

Целта на овој проект е да се направи апликација на која ќе може да се пребаруваат различни рецепти и истите да можат да се зачувуваат во делот за омилени рецепти. Исто така има chatbot кој може да одговори на различни прашања поврзани со храна. API кое ќе биде користено и од кое ќе се вадат податоците е https://spoonacular.com/food-api/

Структура на проектот
Проектот е составен од три фајлови: index.html, app.js, style.css.

Index.html фајлот е составен од шест главни делови:

- Навигациски дел
- Главен дел во кој е прикажан насловот и дел за брзо пребарување
- Дел за прикажување на резултатите од пребарувањето и копче за прикажување повеќе резултати
- Дел за детален преглед на еден рецепт
- Дел за зачувување на омилените рецепти
- Дел за chatbot (комуникација и прикажување на резултати)

На почетокот се прикажани само навигацискиот дел и главниот дел, останатите делови се скриени и не се прикажуваат.
Навигацискиот дел се сосои од dropdown копче кое овозможува пребарување на рецепти според типот во кој припаѓа истиот, потоа копче за преминување на преглед на фаворити (рецептите кои корисникот ги зачувал како фаворити) и копче за chatbot кој при негово избирање се преминува на делот за комуникација со chatbot.
Овде корисникот има две можни опции за пребарување:

1. **Пребарување според тоа во кој тип спаѓа рецептот**. Можни опции овде се: доручек, стартер, главно јадење, десерт, салата, закуска.
2. **Пребарување според клучен збор.**

Ова се овозможува со тоа што при селектирање на соодветната опција се зема вредноста која е сместена во атрибутот ‘ data-dish-type ’(1) или при внесување на клучниот збор се зачувува истиот(2) и се повикува функцијата fetchAPI(). Оваа функција е асинхрона функција која прави повик кон API-то и има комплексно пребарување.Пребарување според типот на рецептот се прави со тоа што на API-то го дадаваме и параметарот type со вредност вредноста која претходно ја земавме од атрибуот data-dish-type. Пребарување според клучен збор правиме така што на API-то го дадаваме и параметарот query со вредност вредноста која претходно ја зачувавме. Правиме fetch на API-то, ги претвараме резултатите во .json формат и ја повикуваме функцијата generateHTML() со добиените резултати. Ќе имаме добиено само 10 резултати бидејќи почетните вредности на параметрите offset и number се offset=0 и number=10.
Функцијата generateHTML() прави неколку промени во изгледот апликацијата: позадината ја поставува да биде црна боја, ја трга сликата која беше позадина на

почетниот екран, делот за прикажување на резултатите го прави видлив, останатите делови (детален преглед на рецепт, дел за фаворити и chatbot делот остануват невидливи). За подобар изглед се додадени некои css стилови. Резултатите кои се влезен параметар на функцијата се мапираат така што за секој еден резултат се креира нов div елемент кој има слика, наслов, други елементи (дали рецептот е без глутен, колкав health score има рецептот, колку време е потребно да се направи рецептот, дали е вегански, дали е попуварен рецепт) и две копчиња (едно копче за детален преглед на потребните состојќи и инструкциите за подготовка на рецептот и второто копче за зачувување на рецептот во фаворити). Секој овој div елемент се додава на променливата generatedHTML и одкако ќе бидат додадени сите елементи од резултатот, на делот за прикажување на резултати му се додава HTML код кој е всушност вредноста на променливата generatedHTML. Со ова се овозможува прикажување на добиените резултати од повикот кон API-то во формат и начин на кој ние сакаме.
На крајот од резултатите има копче „Load More“ кое при кликнување на него генерире плус 10 нови рецепти на истиот начин како претходно. Ова е направено со цел да не се прикажуваат многу рецепти одеднаш но на корисникот да му се даде можност да прегледува повеќе рецепти доколку сака.

Како што спомнавме претходно, секој рецепт има две копчиња: за преглед на рецептот и додавање на истиот во фаворити.

При селектирање на копчето за преглед на рецептот „View Recipe“ се повикува функцијата View Recipe() која како влезен аргумент го прима id-то на селектираниот рецепт. Оваа функција извршува го извршува следниот API повик: https://api.spoonacular.com/recipes/{id}/analyzedInstructions кој ни враќа детални инструкции за рецептот со соодветниот id и потребните состојки. Потоа се повикува функцијата FetchAPIInfo() која како влезни аргументи прима api и број i кој може да биде 1 или 2. Во овој случај за да ни прикаже преглед за рецептот і ќе прими вредност 1 и ќе се повика функцијата generateHTML2() која како влезен аргумент ги прима резултатите вратени од API повикот во .json формат.
Функцијата generateHTML2() најпрво го прави видлив делот за детален преглед на рецептот, а сите други останати делови ги прави невидливи. Овде имаме три променливи ingred, desc и moreHtml кои на почетокот се празни стрингови. Во променливата moreHtml се креира посакуваниот изглед односно има наслов, листа со потребните состојки која е поделена во две колони, наслов, инструкции (чекори) за изработка на рецептот. Притоа секоја состојка се доава во променливата ingred и секој чекор за изработка се додава во променливата desc. Доколку на крај некоја е празна, се прикажува текст дека нема пронајдени резултати.
Во горниот лев агол на оваа форма има копче со вредност X со кое се излегува од овој поглед односно се враќаме на погледод каде се прикажани резултатите.

При селектирање на копчето “Add to Favourite„ се повикува функцијата AddToFavourite() која како влезен аргумент го прима id-то на рецептот кој сакаме да го додадеме во омилени. Овде ќе ја искористиме променливата arrayID во која ќе бидат зачувани сите id-ја на рецепти кои се додадени во омилени. Доколку низата е празна тоа значи дека сеуште нема зачувано ниту еден рецепт во фаворити. Во тој случај го додаваме id-то на рецептот во низата arrayID и го користиме следниот API повик за резултати: GET https://api.spoonacular.com/recipes/{id}/information кој враќа целосни информации за рецептот со соодветното id, како што се состојки, нутритивни вредности, исхрана и информации за алергени итн. И потоа се повикува функцијата FetchAPIInfo() со параметри api и i со вредност 2 (со цел да се повика

функцијата generateHTML3().) Во случај да не е празна низата arrayID, односно веќе претходно има зачувано некои рецепти во омилени, најпрвин проверуваме дали id-то на елементот кој сакаме да го додадеме веќе се наоѓа во низата arrayID. Доколку го пронајдеме не се извршува ништо, односно тој елемент се наоѓа во омилени. Доколку не се пронајде id-то во низата arrayID тоа значи дека тој елемент не е претходно зачуван во омилени и во тој случај го додаваме id-то во arrayID и повторно ја повикуваме функцијата FetchAPIInfo() со аргументи api= https://api.spoonacular.com/recipes/{id}/information и променлива i=2. Ова го правиме со цел да не е можно еден ист рецепт повеќе пати да биде зачуван во делот за фаворити.

Функцијата generateHTML3() прима влезен аргумент податоци за елементот кој сакаме да го додадеме во фаворити. Потоа се креира div елемент со слика и наслов од елементот кој сакаме да го додадеме и две копчиња: за преглед на рецептот („View Recipe“) и бришење на рецептот од фаворити („Remove“). Креираниот елемент се додава на веќепостоечкиот HTML за фаворити. На овој начин се додава секој елемент кој сакаме да биде зачуван во омилени.
Копчето за преглед на рецептот веќе претходно е објаснет.
Копчето „Remove“ има за цел бришење на рецептот од фаворити. Тоа се постигнува со тоа што кога ќе се кликне на копчето „Remove“ се повикува функцијата „Remove()“ која како влезен аргумент го прима id бројот на елементот кој сакаме да го отстраниме од фаворити. Оваа функција најпрвин го пронаоѓа индексот на кој се наоѓа id бројот (кој е влезен аргумент на оваа функција) во низата arrayID се сместува во променливата index. Со помош на функцијата splice го бришеме id бројот од низата arrayID со помош на следната наредба: `arrayID.splice(index,1)`. Откако id бројот веќе е избришан, го пребришуваме претходно напишаниот код за погледот на фаворити и во еден for циклус за секој id број кој се наоѓа во низата arrayID се прави нов API Повик и се повикува функцијата FetchAPIInfo() со аргументки api и број i=2. Со ова ја постигнуваме целта корисникот да може да ги отстрани елементите кои претходно ги додал во фаворити.

Поради тоа што во последно време се поактуелна е вештачката интелигенција и chatbot-овите, затоа одлучив во апликацијата да додадам и дел во кој корисникот би можел да комуницира со chatbot.
Корисникот може да се навигира кон chatbot-от при селектирање на копчето „Chatbot“ од делот за навигација. Притоа се менуваат следните работи: позадинската слика која е веќепоставена се трга и за позадина се поставува црна боја. Погледот за брзо пребарување, погледот за прикажување на резултатите од пребарувањето, копчето за прикажување на повеќе резилтати, погледот за фаворити и погледот за детален преглед на рецептот се кријат, однодно им се додава својството display = ‘none’, а се прикажува делот за chatbot-от.
Овој поглед е составен од три дела:

- Прв дел во кој се наоѓаат насловот, поднаслов (опис) и форма за пребарување која има input поле во кое корисникот може да напише барање или прашање кое е поврзано со храна.
- Втор дел кој всушност е dropdown копче кое при негово кликање се прикажуваат можни теми, прашања, барања кои можат да бидат поставени. Ова е направено со цел бидејќи chatbot-от не може да одговори на секави прашања и да не дојде до забуна кај корисникот при негово креирање. Можни прашања, односно барања, кои можат да бидат поставени на chatbot-от се:

1. Побарајте рецепти како „пилешки рецепти“ или „шпагети со ракчиња“

2. Побарајте содржина на хранливи материи како „витамин а во 2 моркови“ или „калории се 1 чаша путер“

3. Претворете од една мерна единица во друга со помош на барањето: „2 чаши путер во грамови“

4. Проверете ги подароците за храна велејќи „покажи ми некои подароци за храна“

5. Најдете замени за храна велејќи „што е замена за брашното“

6. Дали сте жедни? Побарајте парови на вино како „кое вино оди добро со шпагети карбонара“

7. Ако сакате повеќе резултати, само кажете „повеќе“

8. За повеќе слични резултати кажете „повеќе како првото/второто/трето...“

-Трет дел во кој се прикажуваат резултатите кои ги генерире chatbot-от.

Кога корисникот ќе го постави прашањето, односно барањето, се повикува функцијата fetchAPIChat(). Оваа функција е асинхрона функција која прави повик до следното api: https://api.spoonacular.com/food/converse?text= vitamin +a +in +2 +carrots
Откако ќе го добиеме резултатот од api- то се повикува функцијата generateChatBotHTML() која како влезен елемент ги прима податоците кои се генерирани при повикот од api-то
Функцијата generateChatBotHTML() има за цел прикажување на добиените резултати со помош на HTML. Податоците се прикажуваат во третиот дел од chatbot погледот кој го спомнавме претходно. Резултатот кој ќе ни го врато chatbot-от може да биде само текст (кога корисникот има му поставува некое прашање) или може да биде текст и производи (елементи) (доколку корисникот бара да му се прикаже / излиста нешто). За таа цел овде користиме една променлива за зачувување на производите (елементите) кои се вратени во резултатот. Таа променлива е променливата ‘chatMedia’ која како што кажав ја користиме за сместување на елементите кои се вратени како резултат. Бидејќи секогаш во вратениот резултат имаме текст најпрво го сместуваме текстот во делот за прикажување на тестот chatResult1. Тоа го правиме со следната наредба : chatResult1.innerHTML= data.answerText
Потоа за да ги прикажеме елементите (доколку ги има секако) проверуваме дали делот media од добиениот резултет (таму се сместени елементите кои ги враќа chatbot-от) има должина поголема од 0. Доколку да, тоа значи дека имаме елементи кои треба да ги прикажеме, па за таа цел ја користиме .map функцијата и за секој резултат креираме div елемент со слика, наслов и копче. Овде копчето е всушност линк кој води до друга страна на која се наоѓа тој елемент / произввод и таму се прикажани детални информации за истиот. Секој креиран div го сместуваме во променливата chatMedia, а потоа нејзината содржина ја презаписуваме на делот каде треба да бидат сместени елементите.
Елементите во секој поглед (овде, погледот за фаворити, погледот за прикажување на резултати од пребарување) се прикажани во грид систем со тоа што секој елемент има фиксна широчина од 360px, а бројот на елементи во ред зависи од тоа колку место има во редот.

За крај остана уште функциите кои се користени за dropdown копчињата. Тоа се функциите *myFunction u myFunction2* кои наизменично ја додаваат односно отстрануваат класата ‘show’ на елементот кој е dropdown копче. Доколку корисникот кликне на друго место освен на dropdown копчето и доколку тоа е отворени (ја има класата ‘show’) тогаш ја тргаме класата ‘show’, однодно го затвораме копчето.