AKADEMIJA TEHNIČKO-UMETNIČKIH STRUKOVNIH STUDIJA BEOGRAD
ODSEK VISOKA ŠKOLA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

WEB APLIKACIJA ZA PRETRAGU KOKTELA SA EKSTERNOG API-JA

ZAVRŠNI RAD

Mentor:
Dr Nenad Kojić dipl. ing.

Kandidat:
Jovan Vučeljić 162/14

Beograd, 2021.

AKADEMIJA TEHNIČKO-UMETNIČKIH STRUKOVNIH STUDIJA BEOGRAD
ODSEK VISOKA ŠKOLA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

Internet tehnologije

Predmet: Web programiranje

Tema: Web aplikacija za pretragu koktela sa eksternog API-ja

Mentor:

Dr Nenad Kojić dipl. ing.

Kandidat:

Jovan Vučeljić 162/14

Beograd, 2021.

Sadržaj

1. Uvod

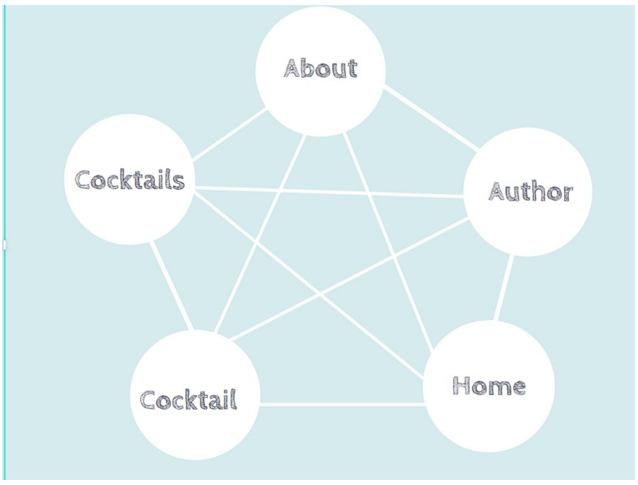
Savremeni razvoj web aplikacija se obično deli na logičke delove pod nazivom „slojevi", gde je svakom sloju dodeljena uloga. Iako su moguće mnoge varijacije, najčešća struktura je troslojna aplikacija. U svojoj najuobičajenijoj formi, tri sloja se nazivaju prezentacija, aplikacija i skladištenje. Diplomski rad predstavlja prezentacioni sloj, dok ostale slojeve koristi gotove, besplatne i preuzete sa interneta. Korišćene tehnologije su u trenutku izrade jedne od najpopularnijih i najkorištenijih na tržištu.

Aplikacija je detaljan prikaz koktela sa slikom, podacima i receptima sa pretragom i filtriranjem po imenu, sastojcima, kategoriji, vrsti čaše i sortiranje po nazivu. Aplikacija je zapravo rešenje problema za potrebe ličnog hobija.

Uzroci problem prilikom izrade su bili nekompletna i nestandardna struktura za rute na API-ju, glavni je bio taj da ne podržavaju upite sa više parametara. Pošto su potrebe aplikacije komplikovanije pretrage i filtriranje rezultata, taj problem je rešen tako sto se vrši više upita istovremeno, pa se rezltati dodatno obrađuju pre prikaza. Takvo rešenje nije optimalno, ali nije bilo izbora.

2. Organizacija

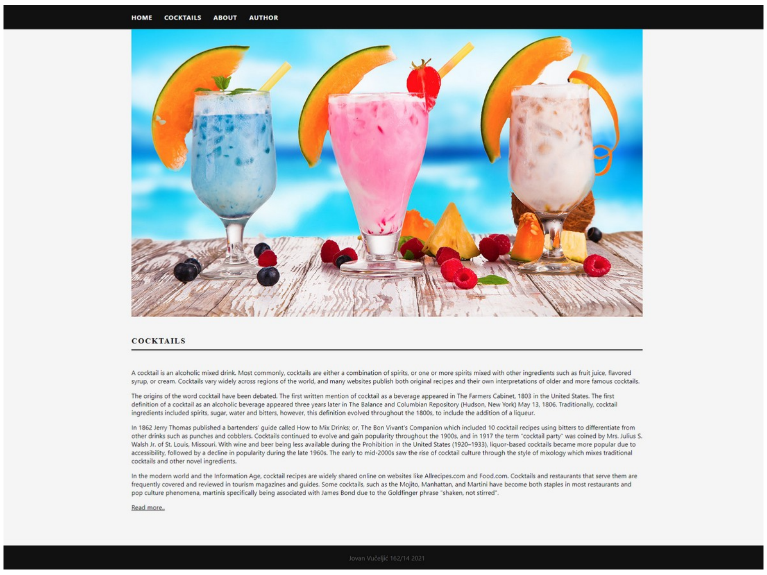
Projekat se sastoji od 5 strana koje su medjusobno povezane menijem, tako da se svakoj može pristupiti sa ostalih. (Slika 2.1)



Slika 2.1 – Stranice i njihov odnos

2.1. Početna strana

Početna strana sadrži sliku, kratak opis i istoriju koktela, kao i hiperlink za vise informacija za one koje zanima. (Slika 2.2)



COCKTAILS

A cocktail is an alcoholic mixed drink. Most commonly, cocktails are either a combination of spirits, or one or more spirits mixed with other ingredients such as fruit juice, flavored syrup, or cream. Cocktails vary widely across regions of the world, and many websites publish both original recipes and their own interpretations of older and more famous cocktails.

The origins of the word cocktail have been debated. The first written mention of cocktail as a beverage appeared in The Farmers Cabinet, 1803 in the United States. The first definition of a cocktail as an alcoholic beverage appeared three years later in The Balance and Columbian Repository (Hudson, New York) May 13, 1806. Traditionally, cocktail ingredients included spirits, sugar, water and bitters, however, this definition evolved throughout the 1800s, to include the addition of a liqueur.

In 1862 Jerry Thomas published a bartenders' guide called How to Mix Drinks; or, The Bon Vivant's Companion which included 10 cocktail recipes using bitters to differentiate from other drinks such as punches and cobblers. Cocktails continued to evolve and gain popularity throughout the 1900s, and in 1917 the term "cocktail party" was coined by Mrs. Julius S. Walsh Jr. of St. Louis, Missouri. With wine and beer being less available during the Prohibition in the United States (1920–1933), liquor-based cocktails became more popular due to accessibility, followed by a decline in popularity during the late 1960s. The early to mid-2000s saw the rise of cocktail culture through the style of mixology which mixes traditional cocktails and other novel ingredients.

In the modern world and the Information Age, cocktail recipes are widely shared online on websites like Allrecipes.com and Food.com. Cocktails and restaurants that serve them are frequently covered and reviewed in tourism magazines and guides. Some cocktails, such as the Mojito, Manhattan, and Martini have become both staples in most restaurants and pop culture phenomena, martinis specifically being associated with James Bond due to the Goldfinger phrase "shaken, not stirred".

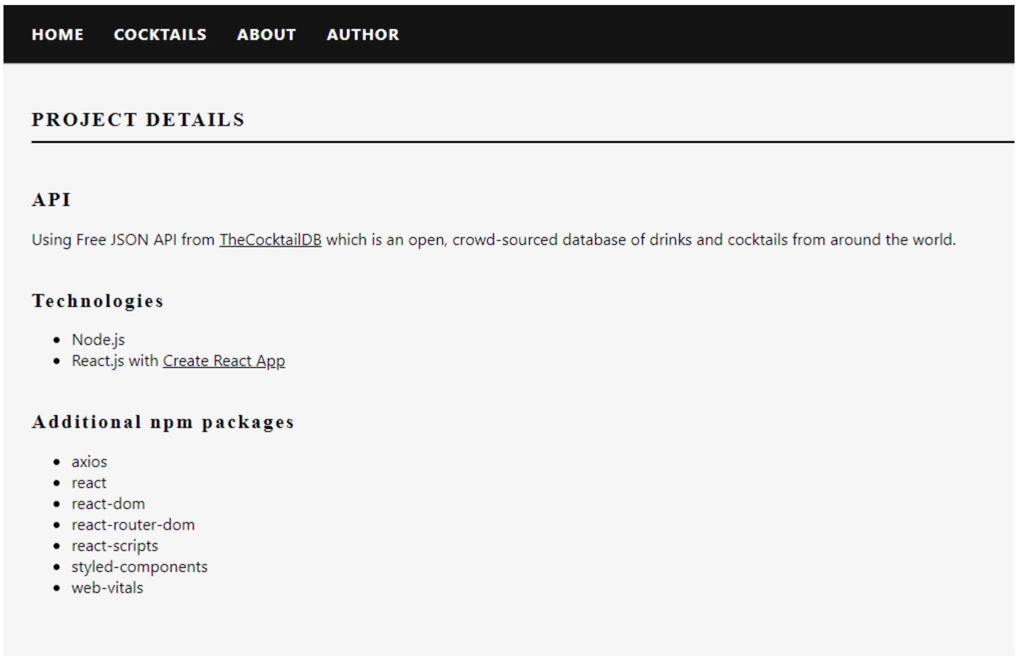
Read more..

Jovan Vučeljić 162/14 2021

Slika 2.2 – Home strana

2.2. About strana

Na „About“ strani nalazi se kratak opis korišćenih tehnologija, paketa i hiperlinkovi za vise informacija. (Slika 2.3)



PROJECT DETAILS

API

Using Free JSON API from TheCocktailDB which is an open, crowd-sourced database of drinks and cocktails from around the world.

Technologies

- Node.js
- React.js with Create React App

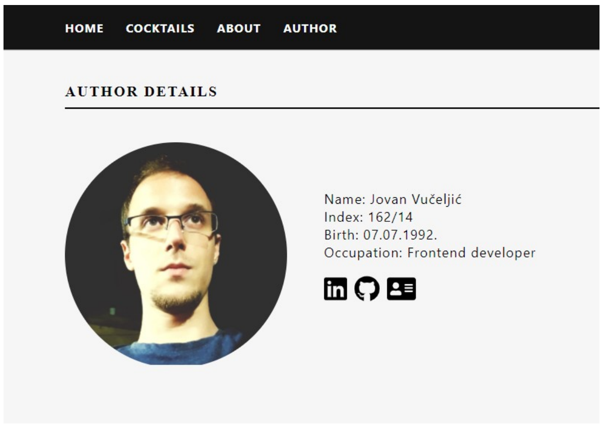
Additional npm packages

- axios
- react
- react-dom
- react-router-dom
- react-scripts
- styled-components
- web-vitals

Slika 2.3 – About stranica

2.3. Autor strana

Na „Author“ stranici nalaze se podaci o autoru kao i linkovi ka njegovim poslovnim profilima. (Slika 2.4)



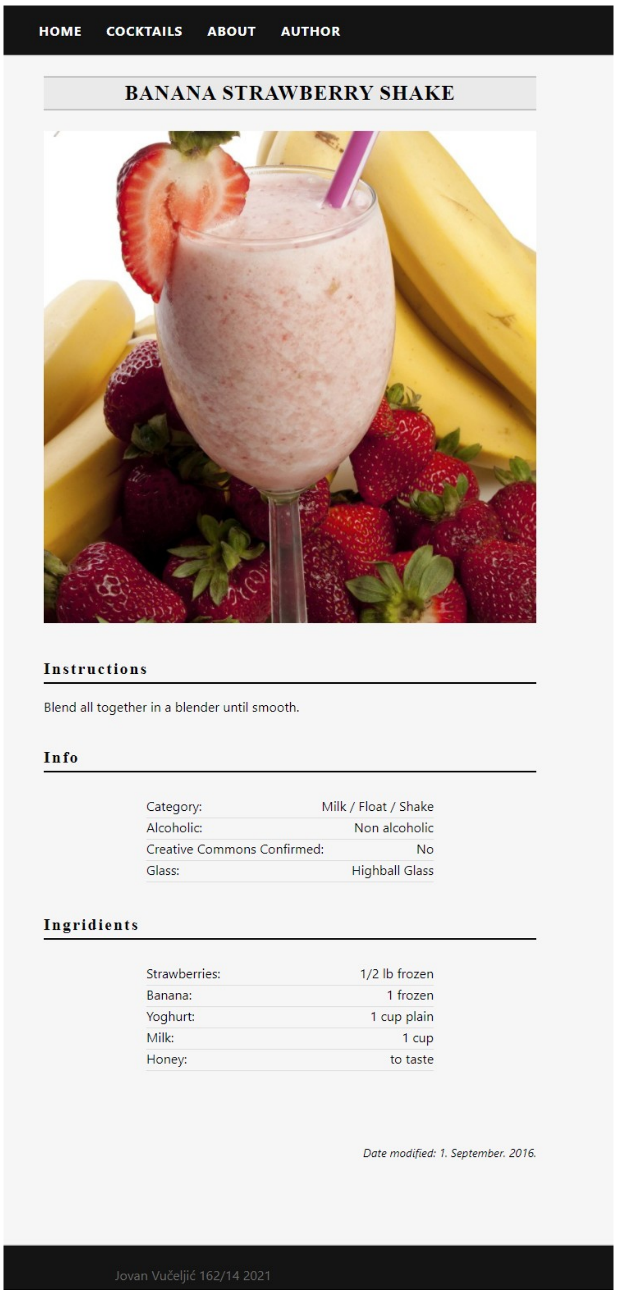
AUTHOR DETAILS

Name: Jovan Vučeljić
Index: 162/14
Birth: 07.07.1992.
Occupation: Frontend developer

Slika 2.4 – Stranica o autoru

2.4. Pregled koktela strana

Ovo je dinamicna strana koja učitava koktel odabrane po id, prikazuju se kratke instrukcije pripreme, sastojci koktela, kao i kojoj kategoriji pripada u kojoj časi se poslužuje itd. (Slika 2.5)



BANANA STRAWBERRY SHAKE

Instructions

Blend all together in a blender until smooth.

Info

Category:	Milk / Float / Shake
Alcoholic:	Non alcoholic
Creative Commons Confirmed:	No
Glass:	Highball Glass

Ingridients

Strawberries:	1/2 lb frozen
Banana:	1 frozen
Yoghurt:	1 cup plain
Milk:	1 cup
Honey:	to taste

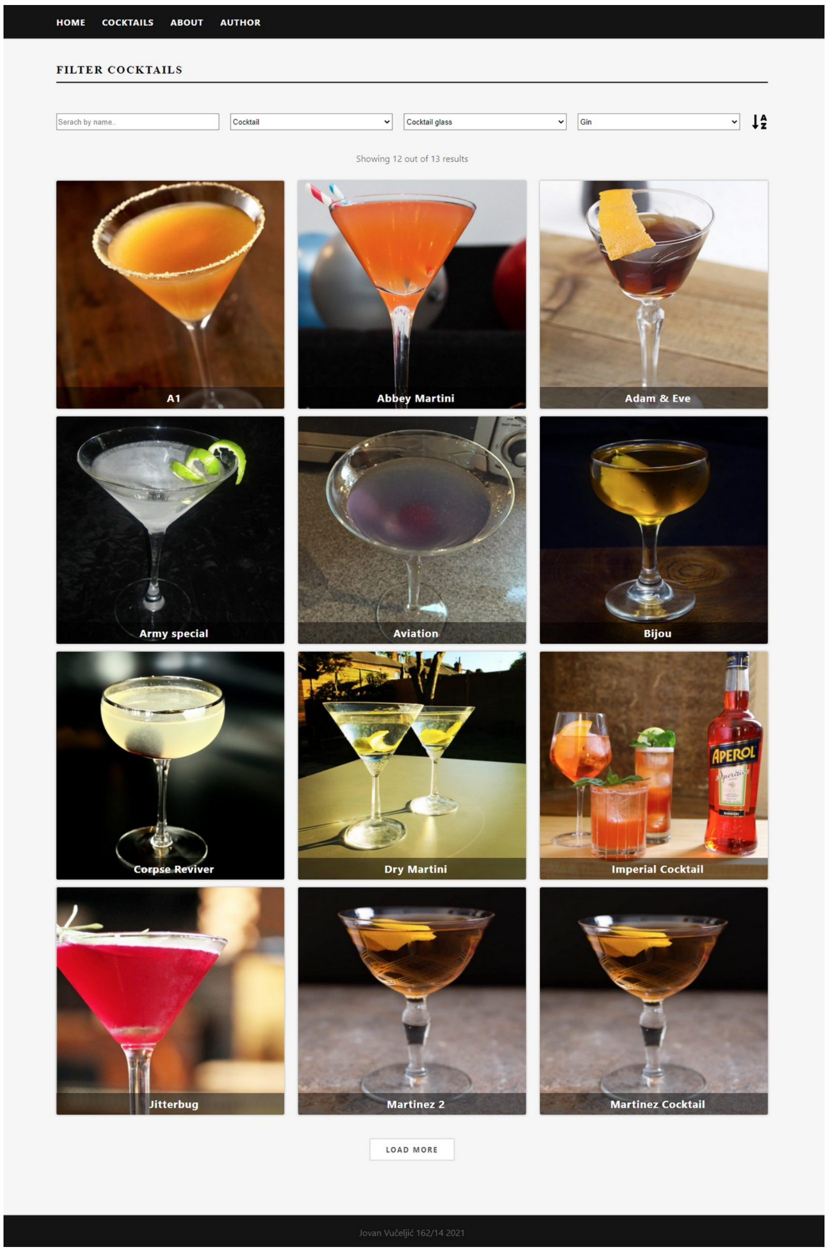
Date modified: 1. September. 2016.

Jovan Vučeljić 162/14 2021

Slika 2.5 – Stranica „Cocktail“

2.5. Pretraga i filter strana

Stranica na kojoj se vrši pretraga i filtriranje. Parametri su naziv, kategorija, sastojak, čaša i vrsta sortiranja. Podaci koji se selektivno biraju su učitani dinamično preko Api. Pri svakoj promeni nekog od parametara pretrage, podaci se reaktivno dohvataju i obradjuju. Koristi se „lazy load“ metoda prikaza podataka, tj. ako ima više od 12 rezultata, ostali se ne renderuju, već se dodatnih 8 prikazuje klikom na dugme „Load more“ i tako dok se ne učitaju svi. (Slika 2.6)



Slika 2.6 – Stranica „Cocktails“ za pretragu i filtriranje

3. Eksterni API

Podaci za potrebe aplikacije se preuzimaju sa portala TheCocktailDB[1], crowd-sourced, otvorena baza podataka pića i koktela, podaci se dostavljaju u JSON formatu preko API. Dodatne informacije se nalaze na sledećem linku *https://www.thecocktaildb.com/*.

Podržane rute nalaze se na sledem linku – *https://www.thecocktaildb.com/api.php*

Korišćene rute za svrhe projekta:

- Pretraga i filter strana:
  - Lista kategorija – *https://www.thecocktaildb.com/api/json/v1/1/list.php?c=list*
  - Lista čaša – *https://www.thecocktaildb.com/api/json/v1/1/list.php?g=list*
  - Lista sastojaka – *https://www.thecocktaildb.com/api/json/v1/1/list.php?i=list*
  - Pretraga koktela po imenu – *https://www.thecocktaildb.com/api/json/v1/1/search.php?s=margarita*
  - Filter po sastojku – *https://www.thecocktaildb.com/api/json/v1/1/filter.php?i=Gin*
  - Filter po kategoriji – *https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Ordinary_Drink*
  - Filter po čaši – *https://www.thecocktaildb.com/api/json/v1/1/filter.php?g=Cocktail_glass*

- Pregled koktela strana:
  - Pun prikaz koktela po id – *https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=11007*

4. Tehnologije

Korišćene tehnologije za potrebe projekta su Node.js[2] i React.js[3][6][7]. Razlozi za njihov izbor za projekat su to što sam već upoznat sa kombinacijom, posedujem i radno iskustvo u industriji, a statistike zastupljenosti, korišćenosti u svetu su sve bolje, svake godine.

4.1. Node.js

Node.js[4] je višeplatformsko JavaScript radno okruženje otvorenog koda za izvršavanje JavaScripta na serverskoj strani. Istorijski gledano JavaScript je primarno korišćen na klijentskoj strani, gde su skripte napisane u JavaSkriptu bile ugrađene u HTML stranice, kako bi se izvršile na klijentskoj strani u web pregledaču. Node.js omogućava da se JavaSkript koristi za skripte na serverskoj strani koje omogućavaju da se sadržaj dinamičnih web stranica generiše na serveru pre nego što se pošalje do web pregledača korisnika. Zbog toga je Node.js postao jedan od osnovnih elemenata paradigme „JavaScript svuda“ jer omogućava uniformisanje razvoj web aplikacija u jednom programskom jeziku, bez potrebe da se za skripte na serverskoj strani koristi različit programski jezik.

4.1.1. NPM

Npm (engl. Node Package Manager)[4] je menadžer paketa za JavaScript programski jezik. On je takođe podrazumevani menadžer paketa za JavaScript radno okruženje Node.js. Sastoji se od konzole, koja se još naziva npm i onlajn baze podataka javnih i plaćenih privatnih paketa, koji se nazivaju npm registri. Ovim registrima se pristupa preko klijentske konzole, dok se dostupni paketi mogu pretraživati na npm websajtu. Menadžerom paketa i registrima upravlja kompanija npm (engl. npm, Inc.)[4]

Npm paketi korišćeni za potrebe projekta su:

- axios
- react
- react-dom
- react-router-dom
- react-scripts
- styled-components
- web-vitals

4.2. React.js

React.js[3][6][7] je JavaScript biblioteka otvorenog koda koja se koristi za izradu korisničkih interfejsa ili UI komponenti. React.js se koristi za izradu web, kao i mobilnih aplikacija. React obezbeđuje programeru model u kojem podkomponente ne mogu direktno da utiču na spoljašnje komponente, efikasno ažuriranje prikaza u datom trenutku pri promeni podataka i jasno razdvajanje komponenti.

4.2.1. Create React App

Create React App[5] je oficijalno podržan način za kreiranje React aplikacije, koje nudi modernu postavku bez dodatne konfiguracije. Uz jednu komandu instalira se skup prekonfigurisanih npm paketa[4] kako bi developeri mogli odmah da se fokusiraju na kod i krenu sa izradom aplikacije.

5. Kodovi

Projekat, kodovi i uputstvo za konfiguraciju i startovanje aplikacije, koji je javan i plan je da tako i ostane, se nalaze na sledećem github linku – *https://github.com/JovanVuceljic/CocktailDIY/*

Struktura projekta:

- ***/api/***
  - *functions.js*
- ***/components/***
  - *DrinksGrid.js*
  - *NoticeMessages.js*
  - *Header.js*
  - *Footer.js*
- ***/pages/***
  - *Home.js*
  - *About.js*
  - *Author.js*
  - *Cocktails.js*
  - *Drink.js*
- *App.js*
- *CocktailContext.js*
- *index.css*
- *index.js*

5.1. index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import'./index.css';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

5.2 index.css

```
body {
  margin: 0;
  padding: 0;
  background-color: #f5f5f5;
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",
    "Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, "Courier New",
    monospace;
}
a {
  color: black;
}
a:hover {
  opacity: 0.8;
}
img {
  width: 100%;
}
h1 {
  text-transform: uppercase;
  font-size: 20px;
  letter-spacing: 2px;
  margin-bottom: 45px;
  border-bottom: 2px solid black;
  padding-bottom: 10px;
  margin-top: 45px;
  font-family: "sans-serif";
}
h2 {
  font-size: 20px;
  letter-spacing: 2px;
  margin-top: 40px;
  font-family: "sans-serif";
}
div.open {
  display: block;
}
```

5.3. App.js

```
import React, { useState } from 'react';
import styled from 'styled-components';
import { BrowserRouter, Route, Switch } from "react-router-dom";
import Header from './components/Header.js';
import Footer from './components/Footer.js';
import Home from './pages/Home.js';
import About from './pages/About.js';
import Author from './pages/Author.js';
import Drink from './pages/Drink.js';
import Cocktails from './pages/Cocktails.js';
import { CocktailContext } from './CocktailContext.js';
const Content = styled.div`
  min-height: 100vh;
        max-width: 1280px;
        margin: 0 auto;
  padding: 0 20px 70px 20px;
`;
const App = () => {

  const [visitedCocktails, setVisitedCocktails] = useState([]);

  return (
    <BrowserRouter>
      <div className="App">
        <Header />
        <Content>
          <Switch>
            <Route exact path="/" component={Home} />
            <Route path="/about" component={About} />
            <Route path="/author" component={Author} />
            <CocktailContext.Provider value={{visitedCocktails,setVisitedCocktails}}>
              <Route path="/cocktails" component={Cocktails} />
              <Route path="/drink/:id" component={Drink} />
            </CocktailContext.Provider>
          </Switch>
        </Content>
        <Footer />
      </div>
    </BrowserRouter>
  );
}
export default App;
```

5.4. CocktailContext.js

```
import { createContext } from "react";
export const CocktailContext = createContext(null);
```

5.5. Api

5.5.1 functions.js

```
import axios from 'axios';
const API_URL = "https://thecocktaildb.com/api/json/v1/1/";
const config = { mode: 'no-cors' };

export const fetchDrinksByName = (name = '') => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}search.php?s=${name}`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}

export const fetchDrinksByIngridientName = (ingridient = '') => {
        console.log(ingridient);
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}search.php?i=${ingridient}`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}


export const fetchIngridientDrinks = (ingridient = '') => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}filter.php?i=${ingridient}`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}

export const fetchGlassDrinks = (glass = '') => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}filter.php?g=${glass}`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}


export const fetchCategoryDrinks = (category = '') => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}filter.php?c=${category}`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}

export const fetchGlasses = () => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}list.php?g=list`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}

export const fetchIngridients = () => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}list.php?i=list`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}

export const fetchCategories = () => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}list.php?c=list`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}


export const fetchDrink = (id) => {
        return new Promise((resolve, reject) => {
                axios
                        .get(`${API_URL}lookup.php?i=${id}`, config)
                        .then(res => {
                                resolve(res.data.drinks);
                        })
                        .catch(err => {
                                console.error(err.message);
                                reject();
                        });
        });
}
```

5.6. Components

5.6.1 DrinksGrid.js

```
import React, { useState } from 'react';
import { Link } from "react-router-dom";
import styled from 'styled-components';
import NoticeMessage from './NoticeMessage';

const Grid = styled.div`
        display: grid;
        grid-template-columns: 32% 32% 32%;
        grid-column-gap: 2%;

        @media only screen and (max-width: 680px) {
                grid-template-columns: 49% 49%;
        }
`;

const DrinkWrap = styled.div`
        box-shadow: 0 0 3px 1px rgb(0 0 0 / 25%);
        border-radius: 3px;
        margin-bottom: 14px;
        overflow: hidden;
        position: relative;
        &>div {
    display: grid;
                overflow: hidden;
        }
        img {
                transition: transform 0.3s ease-in-out;
        }
        &:hover {
                img {
                        transform: scale(1.1);
                }
        }
`;

const Heading = styled.h3`
        color: white;
        background: rgba(0,0,0,0.5);
        text-align: center;
        margin: 0;
        padding: 7px;
        margin-top: -4px;
        letter-spacing: 1px;
        white-space: nowrap;
        text-overflow: ellipsis;
        overflow: hidden;
        position: absolute;
        bottom: 0;
        width: 100%;
`;


const Img = styled.img`
        width: 100%;
`;

const LoadMore = styled.div`
        box-shadow: 0 0 2px 1px rgb(0 0 0 / 20%);
        color: #555;
        padding: 10px 0;
        text-transform: uppercase;
        letter-spacing: 2px;
        font-size: 13px;
        font-weight: bold;
        max-width: 150px;
        text-align: center;
        background: white;
        margin: 30px auto;
        cursor: pointer;
        user-select: none;
        &:hover {
                opacity: 0.8;
        }
        &:active {
                box-shadow: inset 0 0 2px 1px rgb(0 0 0 / 20%);
        }
`;

const DrinksGrid = (props) => {
        const [limit, setLimit] = useState(12)
        const { elements, hideMessage } = props;
        const gridItems = elements.slice(0, limit) || [];
        return (
                <div>
                        {!hideMessage && <NoticeMessage message={`Showing ${limit <
elements.length ? limit : elements.length} out of ${elements.length} results`} />}
                        <Grid>
                                {gridItems.map((el, i) => {
                                        return (<Drink key={i} drink={el} />)
                                })}
                        </Grid>
                        {gridItems.length !== elements.length ? <LoadMore onClick={() =>
setLimit(limit + 6)}>Load More</LoadMore> : ""}
                </div>
        )
}


const Drink = ({ drink }) => {
        const { idDrink, strDrink, strDrinkThumb, strImageAttribution } = drink;

        return (
                <Link to={`/drink/${idDrink}`}>
                        <DrinkWrap>
                                <div>
                                        <Img src={strDrinkThumb} alt={strImageAttribution} />
                                </div>
                                <Heading>{strDrink}</Heading>
                        </DrinkWrap>
                </Link>
        )
}

export default DrinksGrid;
```

5.6.2 NoticeMessage.js

```
import React from 'react';
import styled from 'styled-components';

const Message = styled.div`
        width: 100%;
        text-align: center;
        color: rgba(0,0,0,0.6);
        margin-bottom: 30px;
`;

const NoticeMessage = ({ message = "No data" }) => {
        return (<Message>{message}</Message>)
}

export default NoticeMessage
```

5.6.3 Header.js

```
import React, { useState } from 'react';
import { Link } from "react-router-dom";
import styled from 'styled-components';

// styled component as object
const HeaderWrap = styled.div({
        backgroundColor: '#111',
        height: '60px',
        width: '100%',
        boxShadow: '0 0 1px 1px rgba(0,0,0,0.4)',
});
const MenuItem = styled.div`
        display: flex;
        justify-content: center;
        align-items: center;
        text-transform: uppercase;
        letter-spacing: 1px;
        font-weight: bold;
        padding: 0 15px;
        cursor: pointer;
        user-select: none;
        color: white;
        &:first-child {
                padding-left: 0;
        }
        a {
                color: white;
                text-decoration: none;
                width: 100%;
                height: 100%;
                display: flex;
                align-items: center;
                justify-content: center;
                &:hover {
                        opacity: 0.7;
                }
        }
`;
const Bars = styled.img`
        display:none;
        @media only screen and (max-width: 680px) {
```

```
    display: block;
    height: 40px;
    margin-left: auto;
    width: 60px;
    padding-top: 10px;
        }
`;
const Menu = styled.div`
        max-width: 1280px;
        margin: 0 auto;
        height: 100%;
        display: flex;
        align-items: ceneter;
        justify-conent: center;
        @media only screen and (max-width: 1280px) {
                padding: 0 50px;
        }
        @media only screen and (max-width: 680px) {
                display: none;
                flex-direction: column;
                position: absolute;
                top: 60px;
                left: 0;
                width:100%;
                height: auto;
                background: #151515;
                padding: 10px 0;
                a {
                        padding: 5px 0;
                }
        }
`;
const Header = (props) => {

        const [toggleMenu, setToggleMenu] = useState(false);


        const handleToggleMenu = () => {
                setToggleMenu(!toggleMenu)
        }

        return (
                <HeaderWrap>
                        <Bars src="/bars-solid.svg" alt="menu bars"
onClick={handleToggleMenu} />
                        <Menu className={toggleMenu && 'open'}>
                                <MenuItem><Link to="/">Home</Link></MenuItem>
                                <MenuItem><Link to="/cocktails">Cocktails</Link></MenuItem>
                                <MenuItem><Link to="/about">About</Link></MenuItem>
                                <MenuItem><Link to="/author">Author</Link></MenuItem>
                        </Menu>
                </HeaderWrap>
        )
}
export default Header;
```

5.6.4 Footer.js

```
import React from 'react';
import styled from 'styled-components';

const FooterWrap = styled.div`
  background-color: #111;
        height: 60px;
        width: 100%;
        color: #666;
        display: flex;
        align-items: center;
        justify-content: center;
        box-shadow: 0 0 1px 1px rgba(0,0,0,0.4);
        `;
const Footer = (props) => {
        return (
                <FooterWrap>
                        <p>Jovan Vučeljić 162/14 2021</p>
                </FooterWrap>
        )
}
export default Footer;
```

5.7. Pages

5.7.1 Home.js

```
import React from 'react';
import styled from 'styled-components';
const ImageWrap = styled.div`
        width: 100%;
`;
const Home = () => {
        return (<div>
                        <ImageWrap>
                                <img src="/cocktail.jpg" alt="Cocktail header"/>
                        </ImageWrap>
                        <h1>Cocktails</h1>
                        <p>A cocktail is an alcoholic mixed drink. Most commonly, cocktails
are either a combination of spirits, or one or more spirits mixed with other ingredients such
as fruit juice, flavored syrup, or cream. Cocktails vary widely across regions of the world,
and many websites publish both original recipes and their own interpretations of older and
more famous cocktails.</p>
                        <p>The origins of the word cocktail have been debated. The first
written mention of cocktail as a beverage appeared in The Farmers Cabinet, 1803 in the United
States. The first definition of a cocktail as an alcoholic beverage appeared three years
later in The Balance and Columbian Repository (Hudson, New York) May 13, 1806. Traditionally,
cocktail ingredients included spirits, sugar, water and bitters, however, this definition
evolved throughout the 1800s, to include the addition of a liqueur.</p>
                        <p>In 1862 Jerry Thomas published a bartenders' guide called How to
Mix Drinks; or, The Bon Vivant's Companion which included 10 cocktail recipes using bitters
to differentiate from other drinks such as punches and cobblers. Cocktails continued to
evolve and gain popularity throughout the 1900s, and in 1917 the term "cocktail party" was
coined by Mrs. Julius S. Walsh Jr. of St. Louis, Missouri. With wine and beer being less
available during the Prohibition in the United States (1920–1933), liquor-based cocktails
became more popular due to accessibility, followed by a decline in popularity during the late
1960s. The early to mid-2000s saw the rise of cocktail culture through the style of mixology
which mixes traditional cocktails and other novel ingredients.</p>
                        <p>In the modern world and the Information Age, cocktail recipes are
widely shared online on websites like Allrecipes.com and Food.com. Cocktails and restaurants
that serve them are frequently covered and reviewed in tourism magazines and guides. Some
cocktails, such as the Mojito, Manhattan, and Martini have become both staples in most
restaurants and pop culture phenomena, martinis specifically being associated with James Bond
due to the Goldfinger phrase "shaken, not stirred".</p>
                        <p><a target="_blank" href="https://en.wikipedia.org/wiki/Cocktail"
rel="noreferrer">Read more..</a></p>
                </div>)}
export default Home
```

5.7.2 About.js

```
import React from 'react';

const About = () => {
        return (
                <div>
                        <h1>Project details</h1>

                        <h2>API</h2>
                        <p>Using Free JSON API from <a target="_blank"
href="https://www.thecocktaildb.com/" rel="noreferrer">TheCocktailDB</a> which is an open,
crowd-sourced database of drinks and cocktails from around the world. </p>

                        <h2>Technologies</h2>
                        <ul>
                                <li>Node.js</li>
                                <li>React.js with <a target="_blank" href="https://create-
react-app.dev/" rel="noreferrer">Create React App</a></li>
                        </ul>

                        <h2>Additional npm packages</h2>
                        <ul>
                                <li>axios</li>
                                <li>react</li>
                                <li>react-dom</li>
                                <li>react-router-dom</li>
                                <li>react-scripts</li>
                                <li>styled-components</li>
                                <li>web-vitals</li>
                        </ul>
                </div>
        )
}
export default About
```

5.7.3 Author.js

```
import React from 'react';
import styled from 'styled-components';
const AuthorImage = styled.div`
        max-width: 300px;
        div {
                border-radius: 50%;
                overflow: hidden;
        }
`;
const Content = styled.div`
        display: flex;
        @media only screen and (max-width: 680px) {
                flex-direction: column;
    align-items: center;
        }
`;
const Info = styled.div`
        display: flex;
        align-items: left;
        flex-direction: column;
        justify-content: center;
        margin-left: 50px;
        font-size: 18px;
        letter-spacing: 1px;
`;
const SocialLinks = styled.div`
        display: flex;
        align-items: left;
        list-style-type: none;
        padding: 20px 0;
        a {
                display: block;
                margin-right: 10px;
        }
        img {
    height: 35px;
                &:hover {
                        opacity: 0.7;
                }
        }
`;
const Author = () => {
  return (
         <div>
            <h1>Author details</h1>
               <Content>
                 <AuthorImage>
                     <div><img src="jovan-vuceljic.jpg" alt="author" /></div>
                 </AuthorImage>
                <Info>
                   <div>Name: Jovan Vučeljić</div>
                   <div>Index: 162/14</div>
                   <div>Birth: 07.07.1992.</div>
                   <div>Occupation: Frontend developer</div>
                   <SocialLinks>
                     <a target="_blank" href="https://rs.linkedin.com/in/jovan-vuceljic"
rel="noreferrer"><img src="linkedin-brands.svg" alt="linkedin" /></a>
                     <a target="_blank" href="https://github.com/JovanVuceljic/"
rel="noreferrer"><img src="github-brands.svg" alt="github" /></a>
                     <a target="_blank" href="https://jovanvuceljic.github.io/portfolio/"
rel="noreferrer"><img src="address-card-solid.svg" alt="portfolio" /></a>
                     </SocialLinks>
                   </Info>
                 </Content>
                </div>)}
export default Author
```

5.7.4 Cocktails.js

```
import React, { useState, useEffect, Fragment, useContext } from 'react';
import styled from 'styled-components';
import DrinksGrid from '../components/DrinksGrid.js';
import NoticeMessage from '../components/NoticeMessage.js';
import {
        fetchGlassDrinks,
        fetchGlasses,
        fetchIngridientDrinks,
        fetchIngridients,
        fetchCategories,
        fetchCategoryDrinks,
        fetchDrinksByName
} from '../api/functions.js';
import { CocktailContext } from '../CocktailContext.js';

const Filters = styled.div`
        display: flex;
        margin: 30px -10px;
        @media only screen and (max-width: 680px) {
          flex-direction: column;
          div {
               width: unset;
            }
        }
`;

const InputWrap = styled.div`
        width: 100%;
        padding: 10px;
        display: flex;
        align-items: center;
        justify-content: center;
        input, select {
                border:  1px solid rgba(0,0,0,0.5);
                width: 100%;
                height: 30px;
                outline-color: rgba(0,0,0,0.5);
        }
`;

const SortSelect = styled.div`
        display: flex;
        height: 30px;
        padding: 10px;
        max-width: 30px;
        width: 100%;
        margin-left: auto;
`;

const LastVisitedCocktails = styled.div`
        display: flex;
        flex-direction: row;
        margin-top: 20px;
        height: 100px;
        overflow: hidden;
        box-shadow: inset 0 0 6px 1px rgb(0 0 0 / 10%);
        background: rgba(128,128,128,0.1);
        padding: 10px 25px;
span {
    letter-spacing: 2px;
    font-weight: bold;
    text-transform: uppercase;
    font-size: 11px;
    align-items: center;
    display: flex;
    margin-right: 20px;
    min-width: 100px;
}
div {
    width: 100%;

  div{
      grid-template-columns: 100px 100px 100px 100px 100px 100px 100px 100px 100px 100px;
      grid-column-gap: 10px;
      opacity: 0.8;
   h3 {
      font-size: 10px;
      letter-spacing: 0;
   }
 }
}
`;
const FilterCocktails = () => {
        const [keyword, setKeyword] = useState("");
        const [glass, setGlass] = useState(null);
        const [ingridient, setIngridient] = useState(null);
        const [category, setCategory] = useState(null);
        const [sortType, setSortType] = useState(1);
        const [glassTypeList, setGlassTypeList] = useState([]);
        const [ingridientList, setIngridientList] = useState([]);
        const [categoryList, setCategoryList] = useState([]);
        const [drinks, setDrinks] = useState(null);
        const [drinksByName, setDrinksByName] = useState(null);
        const [drinksByCategory, setDrinksByCategory] = useState(null);
        const [drinksByGlass, setDrinksByGlass] = useState(null);
        const [drinksByIngridient, setDrinksByIngridient] = useState(null);

        const initialFetch = () => {
                fetchGlasses().then(res => {
                        setGlassTypeList(res)
                }).catch(err => {
                        console.error(err);
                });
                fetchIngridients().then(res => {
                        setIngridientList(res)
                }).catch(err => {
                        console.error(err);
                });
                fetchCategories().then(res => {
                        setCategoryList(res)
                }).catch(err => {
                        console.error(err);
                });
        }

        const fetchByName = () => {
                fetchDrinksByName(keyword).then(res => {
                        setDrinksByName(res);
                }).catch(err => {
                        setDrinks(null)
                        console.error(err);
                });
        }

        const fetchByCategories = () => {
                fetchCategoryDrinks(category).then(res => {
                        setDrinksByCategory(res);
                }).catch(err => {
                        console.error(err);
                });
        }
        const fetchByGlass = () => {
                fetchGlassDrinks(glass).then(res => {
                        setDrinksByGlass(res);
                }).catch(err => {
                        console.error(err);
                });
        }
        const fetchByIngridient = () => {
                fetchIngridientDrinks(ingridient).then(res => {
                        setDrinksByIngridient(res);
                }).catch(err => {
                        console.error(err);
                });
        }
        const intersectionLogic = (firstSet, ...sets) => {
                const intersect = (a, b) => new Set([...a].filter(item => [...b].some(x =>
x.idDrink === item.idDrink)));
                sets.forEach(sItem => firstSet = intersect(firstSet, sItem));
                return firstSet;
        }
        const filterData = () => {
                const sets = [];

                drinksByName && sets.push(new Set(drinksByName))
                drinksByCategory && sets.push(new Set(drinksByCategory))
                drinksByIngridient && sets.push(new Set(drinksByIngridient))
                drinksByGlass && sets.push(new Set(drinksByGlass))

                const filtered = intersectionLogic(...sets);

                return filtered ? [...filtered] : [];
        }
        useEffect(() => {
                initialFetch()
                // eslint-disable-next-line react-hooks/exhaustive-deps
        }, []);

        useEffect(() => {
                glass && fetchByGlass()
                ingridient && fetchByIngridient();
                category && fetchByCategories();
                keyword && fetchByName();
                setSortType(1)
                // eslint-disable-next-line react-hooks/exhaustive-deps
        }, [glass, ingridient, category, keyword]);


        useEffect(() => {
                setDrinks(filterData)
                setSortType(1)
                // eslint-disable-next-line react-hooks/exhaustive-deps
        }, [drinksByIngridient, drinksByName, drinksByGlass, drinksByCategory]);


        useEffect(() => {
                drinks && setDrinks(drinks.sort((a, b) => sortType * a.strDrink >
b.strDrink ? 1 : -1));
        }, [sortType, drinks])

        const handleGlassSelect = (e) => {
                setGlass(e.target.value)
        }

        const handleIngridientSelect = (e) => {
                setIngridient(e.target.value)
        }

        const handleCategorySelect = (e) => {
                setCategory(e.target.value)
        }

        const handleSearch = (e) => {
                setKeyword(e.target.value)
        }
        const handleSortType = () => {
                setSortType(sortType * -1)
        }
        const { visitedCocktails } = useContext(CocktailContext);
        return (
                <div>
```

```
                        {visitedCocktails.length > 0 && (
                                <LastVisitedCocktails>
                                        <span>Last viewed</span>
                                        <DrinksGrid elements={visitedCocktails}
hideMessage={true} />
                                </LastVisitedCocktails>
                        )}
                        <h1>Filter cocktails</h1>
                        <Filters>
                                <InputWrap>
                                        <input type="search" onChange={handleSearch}
placeholder={`Serach by name..`} />
                                </InputWrap>
                                <InputWrap>
                                        <select onChange={handleCategorySelect}>
                                                <option> Select category.. </option>
                                                {categoryList && categoryList.map((el, i) =>
{
                                                        return (
                                                                <option key={i}>
{el.strCategory} </option>
                                                        )
                                                })}
                                        </select>
                                </InputWrap>
                                <InputWrap>
                                        <select onChange={handleGlassSelect}>
                                                <option> Select glass type.. </option>
                                                {glassTypeList && glassTypeList.map((el, i)
=> {
                                                        return (
                                                                <option key={i}>
{el.strGlass} </option>
                                                        )
                                                })}
                                        </select>
                                </InputWrap>
                                <InputWrap>
                                        <select onChange={handleIngridientSelect}>
                                                <option> Select ingridient.. </option>
                                                {ingridientList && ingridientList.map((el, i)
=> {
                                                        return (
                                                                <option key={i}>
{el.strIngredient1} </option>
                                                        )
                                                })}
                                        </select>
                                </InputWrap>
                                <SortSelect onClick={handleSortType}>
                                        {sortType === 1 ? <img src="/sort-alpha-down-
solid.svg" alt="sort asc icon" /> :
                                                <img src="/sort-alpha-up-solid.svg" alt="sort
desc icon" />}
                                </SortSelect>
                        </Filters>
                        {drinks == null ? <NoticeMessage message="Select option" /> :
                                drinks.length ? (
                                        <Fragment>
                                                <DrinksGrid elements={drinks} />
                                        </Fragment>) :
                                        <NoticeMessage message="No data for given filters" />
                        }
                </div>
        )

}
export default FilterCocktails
```

5.7.5 Drink.js

```
import React, { useState, useEffect, useContext } from 'react';
import styled from 'styled-components';
import { fetchDrink } from '../api/functions.js';
import { CocktailContext } from '../CocktailContext.js';
import NoticeMessage from '../components/NoticeMessage.js';
const DrinkWrap = styled.div`
        max-width: 600px;
        margin: 30px auto;
`;
const Headline = styled.h1`
        letter-spacing: 1px;
        text-align: center;
        text-transform: uppercase;
        font-family: "sans-serif";
        font-size: 25px;
        margin: 25px 0;
        background: rgba(128,128,128,0.1);
        padding: 5px 0;
        border-top: 2px solid silver;
        border-bottom: 2px solid silver;
`;
const H2 = styled.h2`
        border-bottom: 2px solid black;
        padding-bottom: 5px;
`;
const Img = styled.img`
        width: 100%;
`;
const Ul = styled.ul`
        list-style-type: none;
        padding: 0;
        max-width: 350px;
        margin: 30px auto;
`;
const Li = styled.li`
        display: flex;
        width: 100%;
        justify-content: space-between;
        width: 100%;
        border-bottom: 1px solid rgba(0,0,0,0.1);
        padding-bottom: 4px;
`;
const DateModified = styled.div`
        display: flex;
        width: 100%;
        justify-content: flex-end;
        margin-top: 90px;
        font-size: 14px;
        font-style: italic;
`;
const Drink = ({ match }) => {
        const { visitedCocktails, setVisitedCocktails } = useContext(CocktailContext);
        const { id } = match.params;
        const [drink, setDrink] = useState(null);
        const fetchData = () => {
                fetchDrink(id).then(res => {
                        setDrink(res[0] || {})
                        const arr = [res[0], ...visitedCocktails];
                        var flags = {};
                        var unique = arr.filter((el) => {
                                if (flags[el.idDrink]) {
                                        return false;
                                }
                                flags[el.idDrink] = true;
                                return true;
                        });
                        setVisitedCocktails(unique.slice(0, 9))
                }).catch(err => {
                        setDrink(null)
                        console.error(err);
                });
        }
        useEffect(() => {
                fetchData()
                // eslint-disable-next-line react-hooks/exhaustive-deps
        }, []);
        const { strAlcoholic, strCategory, strCreativeCommonsConfirmed,
                strDrink, strDrinkAlternate, strDrinkThumb,
                strGlass, strIBA, strImageAttribution, strInstructions, dateModified,
                strIngredient1, strIngredient2, strIngredient3,
                strIngredient4, strIngredient5, strIngredient6,
                strIngredient7, strIngredient8, strIngredient9,
                strMeasure1, strMeasure2, strMeasure3,
                strMeasure4, strMeasure5, strMeasure6,
                strMeasure7, strMeasure8, strMeasure9,
        } = drink || {};
        const monthNames = ["January", "February", "March", "April", "May", "June",
                "July", "August", "September", "October", "November", "December"];
        const date = new Date(dateModified)
        const dateModifiedFormat = `${date.getDate()}. ${monthNames[date.getMonth()]}. $
{date.getFullYear()}.`;
        return (
                drink ? (
                        <DrinkWrap>
                                <Headline>{strDrink}</Headline>
                                <div>
                                        <Img src={strDrinkThumb} alt={strImageAttribution} />
                                </div>
                                <H2>Instructions</H2>
                                <p>{strInstructions}</p>
                                <H2>Info</H2>
                                <Ul>
                                        {strCategory &&
<Li><div>Category:</div><div>{strCategory}</div></Li>}
                                        {strAlcoholic &&
<Li><div>Alcoholic:</div><div>{strAlcoholic}</div></Li>}
                                        {strDrinkAlternate && <Li><div>Drink
Alternate:</div><div>{strDrinkAlternate}</div></Li>}
                                        {strCreativeCommonsConfirmed && <Li><div>Creative
Commons Confirmed: </div><div>{strCreativeCommonsConfirmed}</div></Li>}
                                        {strGlass &&
<Li><div>Glass:</div><div>{strGlass}</div></Li>}
                                        {strIBA &&
<Li><div>IBA:</div><div>{strIBA}</div></Li>}
                                </Ul>
                                <H2>Ingridients</H2>
                                <Ul>
                                        {strIngredient1 &&
(<Li><div>{strIngredient1}:</div><div>{strMeasure1}</div></Li>)}
                                        {strIngredient2 &&
(<Li><div>{strIngredient2}:</div><div>{strMeasure2}</div></Li>)}
                                        {strIngredient3 &&
(<Li><div>{strIngredient3}:</div><div>{strMeasure3}</div></Li>)}
                                        {strIngredient4 &&
(<Li><div>{strIngredient4}:</div><div>{strMeasure4}</div></Li>)}
                                        {strIngredient5 &&
(<Li><div>{strIngredient5}:</div><div>{strMeasure5}</div></Li>)}
                                        {strIngredient6 &&
(<Li><div>{strIngredient6}:</div><div>{strMeasure6}</div></Li>)}
                                        {strIngredient7 &&
(<Li><div>{strIngredient7}:</div><div>{strMeasure7}</div></Li>)}
                                        {strIngredient8 &&
(<Li><div>{strIngredient8}:</div><div>{strMeasure8}</div></Li>)}
                                        {strIngredient9 &&
(<Li><div>{strIngredient9}:</div><div>{strMeasure9}</div></Li>)}
                                </Ul>
                          <DateModified>Date modified: {dateModifiedFormat} </DateModified>
                        </DrinkWrap>) : <NoticeMessage message="That cocktail doesn't
exist" />        )}
export default Drink;
```

6. Zaključak

U izradu aplikacije sam ušao tako sto nisam dovoljno istražio i testirao API, očekivao sam da rute prihvataju više parametara, kao i da su greške obrađene kako je propisano standardom, što je zahtevalo dodatnu obradu kasnije, koja je namenjena serverskoj, a ne klijentskoj strani.

Pošto je korišćeni API besplatan, najverovatnije će i aplikacija ubrzo naći svoje mesto na forumima tog API-ja, a možda i šire. Projekat će ostati otvorenog koda[6], ako neko želi da učestvuje u daljem u razvoju ili da koristi kao pomoć za svoj hobi.

Posle dužeg vremena u industriji, upoznavajući se sa vlasničkim softverom, kao i sa softverom otvorenog koda, odlučio sam da želim većinu svoje karijere provesti u modelu otvorenog koda, što je i razlog zašto sam se za projekat odlučio za ove tehnologije. Gledajući na rast njihove popularnosti, zajedno sa ostalim otvorenim softverom, tokom poslednjih par godina, dobijam utisak da će u toj borbi preovladati softver otvorenog koda i da će društvo imati veći benefit nego nekolicina vlasnika, kako je to do sada bilo, drugim rečima može se reći da se nalazimo u srednjem periodu te revolucije.

7. Literatura

[1] *https://www.thecocktaildb.com/*

[2] *https://nodejs.org/en/*

[3] *https://reactjs.org/*

[4] *https://www.npmjs.com/*

[5] *https://create-react-app.dev/*

[6] *Learning React: Functional Web Development with React and Redux*
by Alex Banks and Eve Porcello

[7] *The Road to learn React: Your journey to master plain yet pragmatic React.js*
by Robin Wieruch