

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Керівник роботи

_____ Максим ГОЛОВЧЕНКО

“ ” _____ 2024 р.

ІНТЕЛЕКТУАЛЬНИЙ ІНТЕРНЕТ-МАГАЗИН

Текст програми

КПІ.ПІ-1522.045440.03.12

“ПОГОДЖЕНО”

Керівник роботи:

_____ Максим ГОЛОВЧЕНКО

Консультант:

_____ Максим ГОЛОВЧЕНКО

Виконавець:

_____  Андрій МЄШКОВ

Київ – 2024

```
// ./client/src/reportWebVitals.js
```

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;
```

```
// ./client/src/index.js
```

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import { AppProvider } from "./context/productContext";
import { FilterProvider } from "./context/filterContext";
import { CartProvider } from "./context/cartContext";
import { CheckoutProvider } from "./context/checkoutContext";
import { AuthContextProvider } from "./config/firebase";
```

```
const root = ReactDOM.createRoot(document.getElementById("root"));
```

```
root.render(
```

```
<AppProvider>
  <AuthProvider>
    <FilterProvider>
      <CartProvider>
        <CheckoutProvider></CheckoutProvider>
      <App />
    </CartProvider>
  </FilterProvider>
</AuthProvider>
</AppProvider>
```

```
);
```

```
reportWebVitals();
```

```
// ./client/src/GlobalStyle.js
```

```
import { createGlobalStyle } from "styled-components";
```

```
export const GlobalStyle = createGlobalStyle`
```

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
  font-family: "Work Sans", sans-serif;  
  
}
```

```
html {  
  font-size: 62.5%;  
  /* scroll-behavior: smooth; */  
  /* 1rem = 10px */  
  overflow-x: hidden;  
}
```

```
body {  
  overflow-x: hidden;  
  background: ${({ theme }) => theme.colors.bgS};  
  scrollbar-color: rgb(98 84 243);  
  scrollbar-width: thin;  
  margin: 0;  
  padding: 0;  
}
```

```
body::-webkit-scrollbar {  
  width: 1.5rem;  
}
```

```
body::-webkit-scrollbar-track {  
  background-color: rgb(24 24 29);  
}
```

```

body::-webkit-scrollbar-thumb {

    background: #fff;
    border: 5px solid transparent;
    border-radius: 9px;
    background-clip: content-box;
}

${" /* section{
    background: ${(({ theme }) => theme.colors.bgS});
} */}

h1,
h2,
h3,
h4 {
    font-family: "Work Sans", sans-serif;

}

h1 {
    color: ${(({ theme }) => theme.colors.heading)};
    font-size: 6rem;
    font-weight: 900;
}

h2 {
    color: ${(({ theme }) => theme.colors.heading)};
    font-size: 4.4rem;
    font-weight: 300;
    white-space: normal;

}

h3 {
    font-size: 1.8rem;
    font-weight: 400;
}

p, button {
    color: ${(({ theme }) => theme.colors.text)};
    font-size: 1.65rem;
    line-height: 1.5;
    font-weight: 400;
}

```

```
}
```

```
a {  
  text-decoration: none;  
}
```

```
li {  
  list-style: none;  
}
```

```
`${"" /* reusable code section */}
```

```
.container {  
  max-width: 120rem;  
  margin: 0 auto;  
}
```

```
.grid {  
  display: grid;  
  gap: 9rem;  
}
```

```
.grid-two-column {  
  grid-template-columns: repeat(2, 1fr);  
}
```

```
.grid-three-column {  
  grid-template-columns: repeat(3, 1fr);  
}
```

```
.grid-four-column {  
  grid-template-columns: 1fr 1.2fr .5fr .8fr ;  
}
```

```
.grid-five-column {  
  grid-template-columns: repeat(5, 1fr);  
}
```

```
.common-heading {  
  font-size: 3.8rem;  
  font-weight: 600;  
  margin-bottom: 6rem;
```

```

    text-transform: capitalize;
}

.intro-data {
    margin-bottom: 0;
    text-transform: uppercase;
    color: #fff;
    ${" /* color: #c49952; */}

}

.caption {
    position: absolute;
    top: 15%;
    right: 10%;
    text-transform: uppercase;
    background-color: ${{ theme }} => theme.colors.bg};
    color: ${{ theme }} => theme.colors.helper};
    padding: 0.8rem 2rem;
    font-size: 1.2rem;
    border-radius: 2rem;
}

input, textarea{
    max-width: 50rem;
    color: ${{ theme }} => theme.colors.black};
    padding: 1.6rem 2.4rem;
    border: 1px solid ${{ theme }} => theme.colors.border};
    ${" /* text-transform: uppercase; */}
    box-shadow: ${{ theme }} => theme.colors.shadowSupport};
}

input[type="submit"]{
    max-width: 16rem;
    margin-top: 2rem;
    background-color: ${{ theme }} => theme.colors.btn};
    color: ${{ theme }} => theme.colors.white};
    padding: 1.4rem 2.2rem;
    border-style: solid;
    border-width: .1rem;
    text-transform: uppercase;
    font-size: 1.8rem;
    cursor: pointer;
}

@media (max-width: ${{ theme }} => theme.media.tab)} {

```

```

.container {
  max-width: 130rem;
  padding: 0 3.2rem;
}
}

@media (max-width: ${({ theme }) => theme.media.mobile}) {
  html {
    font-size: 50%;
  }
}

.grid{
  gap: 3.2rem;
}
.grid-two-column , .grid-three-column, .grid-four-column{
  grid-template-columns: 1fr;
}
}

`;

```

```
// ./client/src/App.test.js
```

```
import { render, screen } from '@testing-library/react';
import App from './App';
```

```
test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

```
// ./client/src/setupTests.js
```

```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

```
// ./client/src/App.js
```

```

import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Home from "./pages/Home";
import About from "./pages/About";
import Products from "./pages/Products";
import Contact from "./pages/Contact";
import SingleProduct from "./pages/SingleProduct";
import Cart from "./pages/Cart";
import ErrorPage from "./pages/ErrorPage";
import { GlobalStyle } from "./GlobalStyle";
import { ThemeProvider } from "styled-components";
import Header from "./components/Header";
import Footer from "./components/Footer";
import Checkout from "./pages/Checkout";
import Callback from "./pages/Callback";
import Login from "./pages/Sign/Login.jsx";
import Register from "./pages/Sign/Register.jsx";

const App = () => {

  const theme = {
    colors: {

      text: "rgb(255,255,255)",
      heading: "rgba(196,153,82, 1)",
      white: "#fff",
      black: "#212529",
      helper: "#c49952",

      bg: "rgb(72, 54, 54)",
      bgS: "rgb(106, 80, 80)",
      footer_bg: "#0a1435",
      btn: "rgba(98, 84, 243, 0.5)",
      border: "rgba(98,84,243,0.5)",
      hr: "#fff",
      gradient: "linear-gradient(0deg, rgb(132 144 255) 0%, rgb(98 189 252) 100%)",
      shadow: "rgba(0, 0, 0, 0.02) 0px 1px 3px 0px, rgba(27, 31, 35, 0.15) 0px 0px 0px 1px;",
      shadowSupport: "rgba(0, 0, 0, 0.3) 0px 1px 4px",
    },
    media: {
      mobile: "768px",
      tab: "998px",
    }
  };

```



```

    }
  };

```

```

return (
  <ThemeProvider theme={theme} >
    <Router>
      <GlobalStyle />
      <Header/>
      <Routes>
        <Route path="/" element={<Home/>} />
        <Route path="/about" element={<About/>} />
        <Route path="/products" element={<Products/>} />
        <Route path="/contact" element={<Contact/>} />
        <Route path="/singleproduct/:id" element={<SingleProduct/>} />
        <Route path="/cart" element={<Cart/>} />
        <Route path="/checkout" element={<Checkout/>} />
        <Route path="/callback" element={<Callback/>} />
        <Route path="/login" element={<Login/>} />
        <Route path="/register" element={<Register/>} />
        <Route path="*" element={<ErrorPage/>} />
      </Routes>
      <Footer/>
    </Router>
  </ThemeProvider>
)
};

```

```
export default App;
```

```
// ./client/src/context/cartContext.js
```

```

import { createContext, useContext, useEffect, useReducer } from "react";
import reducer from "../reducer/cartReducer";
import { postAction } from "../api/actions";
import { useAuthState } from "../config/firebase";

```

```
const CartContext = createContext();
```

```

const getLocaleCartData = () => {
  let newCartData = localStorage.getItem("coffeeCart");
  if(!newCartData){
    return []
  }

```

```

    } else {
      return JSON.parse(newCartData);
    }
  }
}

const initialState = {
  cart: getLocaleCartData(),
  totalItem: "10",
  totalAmount: "",
  shippingFee: 5,
};

const CartProvider = ({ children }) => {
  const [state, dispatch] = useReducer(reducer, initialState);
  const { uid, isAuthenticated } = useAuthState();

  const addToCart = async(id, amount, product) => {
    try {
      if(isAuthenticated){
        await postAction({
          user_id: uid,
          product_id: id,
          action_type: `cart`,
        });
      }

      dispatch({ type: "ADD_TO_CART", payload: { id, amount, product } });
    } catch (error) {
      console.error('Error posting action and updating filters:', error);
    }
  };

  const setDecrease = (id) => {
    dispatch({ type: "SET_DECREMENT", payload: id });
  };

  const setIncrease = (id) => {
    dispatch({ type: "SET_INCREMENT", payload: id });
  };

  const removeItem = (id) => {
    dispatch({ type: "REMOVE_ITEM", payload: id });
  };

```

```

const clearCart = () => {
  dispatch({ type: "CLEAR_CART" });
};

useEffect(() => {
  dispatch({ type: "CART_ITEM_PRICE_TOTAL" });

  localStorage.setItem("coffeeCart", JSON.stringify(state.cart))
}, [state.cart])

return (
  <CartContext.Provider value={{ ...state, addToCart, removeItem, clearCart,
setDecrease, setIncrease }}>
    {children}
  </CartContext.Provider>
);
};

const useCartContext = () => {
  return useContext(CartContext);
};

export { CartProvider, useCartContext };

// ./client/src/context/productContext.js

import { createContext, useContext, useEffect, useReducer } from "react";
import { commerce } from "../lib/commerce";
import reducer from "../reducer/productReducer"
// import { postRecommendations } from "../api/actions";
// import { useAuthState } from "../config/firebase";

const AppContext = createContext();

const initialState = {
  isLoading: false,
  isError: false,
  products: [],
  isSingleLoading: false,
  singleProduct: {},
  recommendations: []
}

const AppProvider = ({children}) => {
  const [state, dispatch] = useReducer(reducer, initialState);

```

```

// const { uid, isAuthenticated } = useAuthState();

const fetchProducts = async() => {
  dispatch({type:"SET_LOADING"})
  try {
    const {data} = await commerce.products.list();

    dispatch({type:"SET_API_DATA", payload: data})
    // await fetchRecs(data);
  } catch {
    dispatch({type:"API_ERROR"})
  }
};

const getSingleProduct = async(id) => {
  dispatch({type:"SET_SINGLE_LOADING"})
  try {
    const {data} = await commerce.products.list();
    const singleProduct = data.find(product => product.id === id);
    dispatch({type:"SET_SINGLE_PRODUCT", payload: singleProduct});
  } catch (error) {
    dispatch({type:"SET_SINGLE_ERROR"})
  }
}

useEffect(() => {
  fetchProducts();

  // eslint-disable-next-line react-hooks/exhaustive-deps
},[])

return(
  <AppContext.Provider value={{...state, getSingleProduct}} >
    {children}
  </AppContext.Provider>
)
);

const useProductContext = () => {
  return useContext(AppContext)
}

```

```

export {AppProvider, AppContext, useProductContext};

// ./client/src/context/checkoutContext.js

import { createContext, useContext, useReducer } from "react";
import checkoutReducer from "../reducer/checkoutReducer";

const CheckoutContext = createContext();

const initialState = {
  shippingAddress: "",
  paymentMethod: "",
};

const CheckoutProvider = ({ children }) => {
  const [state, dispatch] = useReducer(checkoutReducer, initialState);

  const setShippingAddress = (address) => {
    dispatch({ type: "SET_SHIPPING_ADDRESS", payload: address });
  };

  const setPaymentMethod = (method) => {
    dispatch({ type: "SET_PAYMENT_METHOD", payload: method });
  };

  return (
    <CheckoutContext.Provider value={{ ...state, setShippingAddress,
    setPaymentMethod }}>
      {children}
    </CheckoutContext.Provider>
  );
};

const useCheckoutContext = () => {
  return useContext(CheckoutContext);
};

export { CheckoutProvider, useCheckoutContext };

// ./client/src/context/filterContext.js

```

```
import { createContext, useContext, useEffect, useReducer } from "react";
import { useProductContext } from "../productContext";
import reducer from "../reducer/filterReducer"
```

```
const FilterContext = createContext();
```

```
const initialState = {
  filterProducts: [],
  allProducts: [],
  gridView: true,
  sortingValue: "lowest",
  filters: {
    text: "",
    category: "bce",
    maxPrice: 0,
    price: 0,
    minPrice: 0,
  },
}
```

```
export const FilterProvider = ({children}) => {
```

```
  const {products} = useProductContext();
  const [state, dispatch] = useReducer(reducer, initialState);
```

```
  const setGridView = () => {
    return dispatch({type:"SET_GRID_VIEW"})
  }
```

```
  const setListView = () => {
    return dispatch({type:"SET_LIST_VIEW"})
  }
```

```
  const sorting = (event) => {
    let userValue = event.target.value;
    dispatch({ type: "GET_SORT_VALUE", payload: userValue });
  };
```

```
  const updateFilterValue = async (event) => {
```

```
    try {
      let name = event.target.name;
      let value = event.target.value;

      if (value) value.toLowerCase();
```

```

        return dispatch({ type: "UPDATE_FILTERS_VALUE", payload: { name,
value } });
      } catch (error) {
        console.error('Error posting action and updating filters:', error);
      }
    };
    const clearFilters = () => {
      dispatch({ type: "CLEAR_FILTERS" });
    };

    useEffect(() => {
      dispatch({ type: "FILTER_PRODUCTS" });
      dispatch({ type: "SORTING_PRODUCTS" });
    }, [products, state.sortingValue, state.filters]);

    useEffect(() => {
      dispatch({ type: "LOAD_FILTER_PRODUCTS", payload: products });
    }, [products])

    return(
      <FilterContext.Provider value={{
        ...state,
        setGridView,
        setListView,
        sorting,
        updateFilterValue,
        clearFilters
      }}>
        {children}
      </FilterContext.Provider>
    );
  };

  export const useFilterContext = () => {
    return useContext(FilterContext);
  };

  // ./client/src/config/firebase.js

  import { initializeApp } from "firebase/app";
  import { getAuth, onAuthStateChanged } from "firebase/auth";
  import { useContext } from "react";
  import { createContext } from "react";
  import { useEffect } from "react";

```

```

import { useState } from "react";

const firebaseConfig = {
  apiKey: process.env.REACT_APP_FIREBASE_API_KEY,
  authDomain: process.env.REACT_APP_FIREBASE_AUTH_DOMAIN,
  projectId: process.env.REACT_APP_FIREBASE_PROJECT_ID,
  storageBucket: process.env.REACT_APP_FIREBASE_STORAGE_BUCKET,
  messagingSenderId:
process.env.REACT_APP_FIREBASE_MESSAGING_SENDER_ID,
  appId: process.env.REACT_APP_FIREBASE_APP_ID,
};

const app = initializeApp(firebaseConfig);

export const auth = getAuth(app);

export const AuthContext = createContext();

export const AuthContextProvider = props => {
  const [user, setUser] = useState("");
  const [error, setError] = useState("");

  useEffect(() => {
    const unsubscribe = onAuthStateChanged(getAuth(app), setUser, setError)
    return () => unsubscribe()
  }, [])
  return <AuthContext.Provider value={{ user, error }} {...props} />
}

export const useAuthState = () => {
  const auth = useContext(AuthContext);
  let isAuthenticated = false;
  let uid = "";
  if (auth && auth.user) {
    isAuthenticated = (auth?.user !== null);
    if(auth.user.uid) uid = auth.user.uid;
  };
  return { ...auth, isAuthenticated, uid}
}

// ./client/src/payment/dsf.js

import axios from 'axios';

```



```

const lig = async(data) => {
  try{
    const value = await axios.get('/data');
    const newvalue = await axios.post ('/data', data);

  }catch{}}
}

const lig2 = async() => {
  try{
    const data = await fetch('http://data')
  }catch{}}
}

// ./client/src/payment/liqpay.js

import axios from 'axios';
import crypto from 'crypto';

export class LiqPay {
  constructor(public_key, private_key) {
    this.host = "https://www.liqpay.ua/api/";

    this.api = async function (path, params, callback, callbackerr) {
      if (!params.version) {
        throw new Error('version is null');
      }

      params.public_key = public_key;
      const data = Buffer.from(JSON.stringify(params)).toString('base64');
      const signature = this.str_to_sign(private_key + data + private_key);

      try {
        const response = await axios.post(this.host + path, { data, signature });
        if (response.status === 200) {
          callback(response.data);
        } else {
          callbackerr(response.statusText, response);
        }
      } catch (error) {
        callbackerr(error.message, null);
      }
    };
  }
};

```

```

this.cnb_form = function (params) {
    let language = "en";

    if (params.language) {
        language = params.language;
    }

    params = this.cnb_params(params);
    const data = Buffer.from(JSON.stringify(params)).toString('base64');
    const signature = this.str_to_sign(private_key + data + private_key);

    return `<form method="POST"
action="https://www.liqpay.ua/api/3/checkout" accept-charset="utf-8">
    <input type="hidden" name="data" value="\${data}" />
    <input type="hidden" name="signature" value="\${signature}" />
    <input type="image" src="//static.liqpay.ua/buttons/p1\${language}.radius.png"
name="btn_text" />
</form>`;
};

this.cnb_signature = function (params) {
    params = this.cnb_params(params);
    const data = Buffer.from(JSON.stringify(params)).toString('base64');
    return this.str_to_sign(private_key + data + private_key);
};

this.cnb_params = function (params) {
    params.public_key = public_key;

    if (!params.version || !params.amount || !params.currency ||
!params.description) {
        throw new Error('Some required parameters are missing');
    }

    return params;
};

this.str_to_sign = function (str) {
    const sha1 = crypto.createHash('sha1');
    sha1.update(str);
    return sha1.digest('base64');
};

this.cnb_object = function (params) {

```

```

    params = this.cnb_params(params);
    const data = Buffer.from(JSON.stringify(params)).toString('base64');
    const signature = this.str_to_sign(private_key + data + private_key);

    return { data, signature };
  };

  return this;
}
}

```

// ./client/src/reducer/filterReducer.js

```

const filterReducer = (state, action) => {
  switch(action.type){
    case"LOAD_FILTER_PRODUCTS":
      let priceArr = action.payload.map((curElem) => curElem.price.raw);

      let maxPrice = Math.max(...priceArr)
      if(maxPrice < 0) maxPrice=0;

      return {
        ...state,
        filterProducts: [...action.payload],
        allProducts: [...action.payload],
        filters: {...state.filters, maxPrice, price: maxPrice}
      }
    case"SET_GRID_VIEW":
      return {
        ...state,
        gridView: true,
      }
    case"SET_LIST_VIEW":
      return {
        ...state,
        gridView: false,
      }
    case"GET_SORT_VALUE":
      return {
        ...state,
        sortingValue: action.payload,
      }
    case "SORTING_PRODUCTS":
      let newSortData;

```

```

const { filterProducts, sortingValue } = state;
let tempSortProduct = [...filterProducts];

const sortingProducts = (a, b) => {
  if (sortingValue === "lowest") {
    return a.price.raw - b.price.raw;
  }

  if (sortingValue === "highest") {
    return b.price.raw - a.price.raw;
  }

  if (sortingValue === "a-z") {
    return a.name.localeCompare(b.name);
  }

  if (sortingValue === "z-a") {
    return b.name.localeCompare(a.name);
  }
};

newSortData = tempSortProduct.sort(sortingProducts);

return {
  ...state,
  filterProducts: newSortData,
};
case "UPDATE_FILTERS_VALUE":
  const { name, value } = action.payload;

  return {
    ...state,
    filters: {
      ...state.filters,
      [name]: value,
    },
  };
case "FILTER_PRODUCTS":
  let { allProducts } = state;
  let tempFilterProduct = [...allProducts];

  const { text, category, price } = state.filters;

  if (text) {

```

```

        tempFilterProduct = tempFilterProduct.filter((curElem) => {
            return curElem.name.toLowerCase().includes(text);
        });
    }

    if (category !== "Bce") {
        tempFilterProduct = tempFilterProduct.filter((curElem) => {
            return curElem.categories[0].name === category;
        });
    }

    if(price === 0){
        tempFilterProduct = tempFilterProduct.filter((curElem) =>
curElem.price.raw === price);
    } else {
        tempFilterProduct = tempFilterProduct.filter((curElem) =>
curElem.price.raw <= price);
    }

    return {
        ...state,
        filterProducts: tempFilterProduct,
    };
    case "CLEAR_FILTERS":
        return {
            ...state,
            filters: {
                ...state.filters,
                text: "",
                category: "Bce",
                company: "Bce",
                color: "Bce",
                maxPrice: state.filters.maxPrice,
                price: state.filters.maxPrice,
                minPrice: state.filters.minPrice,
            },
        };
    default: return state;
}
}

export default filterReducer;

// ./client/src/reducer/checkoutReducer.js

```

```
// ./client/src/reducer/productReducer.js

const ProductReducer = (state, action) => {
  switch (action.type) {
    case "SET_LOADING":
      return {
        ...state,
        isLoading: true,
      };

    case "SET_API_DATA":
      return {
        ...state,
        isLoading: false,
        products: action.payload,
      };
    case "SET_RECS_DATA":
      return {
        ...state,
        isLoading: false,
        recommendations: action.payload,
      };

    case "API_ERROR":
      return {
        ...state,
        isLoading: false,
        isError: true,
      };
    case "SET_SINGLE_LOADING":
      return {
        ...state,
        isSingleLoading: true,
      };

    case "SET_SINGLE_PRODUCT":
      return {
        ...state,
        isSingleLoading: false,
        singleProduct: action.payload,
      };
  }
}
```

```

    case "SET_SINGLE_ERROR":
      return {
        ...state,
        isSingleLoading: false,
        isError: true,
      };

    default:
      return state;
  }
}

export default ProductReducer;

// ./client/src/reducer/cartReducer.js

import { renderDefaultCoffee } from "../functions/images";

const cartReducer = (state, action) => {
  if (action.type === "ADD_TO_CART") {
    let { id, amount, product } = action.payload;

    let cartProduct;
    let coffeeI = "";
    if (product.image) {
      coffeeI = product.image.url;
    } else {
      coffeeI = renderDefaultCoffee(product.categories);
    }
    const existingCartItemIndex = state.cart.findIndex(item => item.id === id);
    if (existingCartItemIndex !== -1) {
      const updatedCart = [...state.cart];
      updatedCart[existingCartItemIndex].amount += amount;
      if (updatedCart[existingCartItemIndex].amount > state.max)
        updatedCart[existingCartItemIndex].amount = state.max;
      return {
        ...state,
        cart: updatedCart,
      };
    } else {
      cartProduct = {
        id: id,
        name: product.name,
        amount,

```

```

        image: coffeeI,
        price: product.price.raw,
        max: 20,
    };
    return {
        ...state,
        cart: [...state.cart, cartProduct],
    };
}
}

if (action.type === "SET_DECREMENT") {
    let updatedProduct = state.cart.map((curElem) => {
        if (curElem.id === action.payload) {
            let decAmount = curElem.amount - 1;

            if (decAmount <= 1) {
                decAmount = 1;
            }

            return {
                ...curElem,
                amount: decAmount,
            };
        } else {
            return curElem;
        }
    });
    return { ...state, cart: updatedProduct };
}

if (action.type === "SET_INCREMENT") {
    let updatedProduct = state.cart.map((curElem) => {
        if (curElem.id === action.payload) {
            let incAmount = curElem.amount + 1;
            if (incAmount >= curElem.max) {
                incAmount = curElem.max;
            }
            return {
                ...curElem,
                amount: incAmount,
            };
        } else {
            return curElem;
        }
    });
}

```



```

    });
    return { ...state, cart: updatedProduct };
  }

  if (action.type === "REMOVE_ITEM") {
    let updatedCart = state.cart.filter(
      (curItem) => curItem.id !== action.payload
    );
    return {
      ...state,
      cart: updatedCart,
    };
  }

  if (action.type === "CLEAR_CART") {
    return {
      ...state,
      cart: [],
    };
  }

  if (action.type === "CART_ITEM_PRICE_TOTAL") {
    let { totalItem, totalPrice } = state.cart.reduce(
      (accum, curElem) => {
        let { price, amount } = curElem;
        accum.totalItem += amount;
        accum.totalPrice += price * amount;
        return accum;
      },
      {
        totalItem: 0,
        totalPrice: 0,
      }
    );
    return {
      ...state,
      totalItem,
      totalPrice,
    };
  }

  return state;
};

export default cartReducer;

```

```
// ./client/src/styles/Button.js

import styled from "styled-components";

export const Button = styled.button`
  text-decoration: none;
  max-width: auto;
  ${" /* background-color: rgb(98 84 243); */"}
  background-color: ${({ theme }) => theme.colors.helper};
  color: rgb(0 0 0);
  padding: 1.4rem 2.4rem;
  border: none;
  text-transform: uppercase;
  text-align: center;
  cursor: pointer;
  transition: all 0.3s ease;
  -webkit-transition: all 0.3s ease 0s;
  -moz-transition: all 0.3s ease 0s;
  -o-transition: all 0.3s ease 0s;

  &:hover,
  &:active {
    box-shadow: 0 2rem 2rem 0 rgb(132 144 255 / 30%);
    box-shadow: ${({ theme }) => theme.colors.shadowSupport};
    transform: scale(0.96);
  }

  a {
    text-decoration: none;
    color: rgb(255 255 255);
    font-size: 1.8rem;
  }
`;

// ./client/src/styles/Container.js

import styled from "styled-components";

export const Container = styled.div`
  width: 100%;
  padding: 0rem 12rem;
`;
```

```

// ./client/src/components/Header.jsx

import React from "react";
import { NavLink } from "react-router-dom";
import styled from "styled-components";
import Nav from "./Nav";
import "./stars.scss"

const Header = () => {
  return (
    <MainHeader>
      <NavLink to="/">
        
      </NavLink>
      <div className="bg-animation">
        <div id="stars"></div>
        <div id="stars2"></div>
        <div id="stars3"></div>
        <div id="stars4"></div>
      </div>
      <Nav/>
    </MainHeader>
  )
}

const MainHeader = styled.header`
  padding: 0 4.8rem;
  height: 8rem;
  background-color: ${({theme}) => theme.colors.bg };
  display: flex;
  justify-content: space-between;
  align-items: center;
  position: relative;

  .logo {
    height: 6rem;
    border: 3px solid black;
    border-radius: 50%;
  }
`;

export default Header;

```

```
// ./client/src/components/HeroSection.jsx
```

```
import React from "react";
import { NavLink } from "react-router-dom";
import styled from "styled-components";
import { Button } from "../styles/Button";
```

```
const HeroSection = ({myData}) => {
  const name = myData.name
  return (
    <Wrapper>
      <div className="container">
        <div className="grid grid-two-column">
          <div className="jero-section-data">
            <div className="intro-data">Ласкаво просимо до</div>
            <h1>{name}</h1>
            <p>
              Вітаємо у світі справжнього задоволення кавою! У нашому
інтернет-магазині SuperCoffee ви знайдете найкращі види кави з усього світу,
які призначені задовольнити навіть найвибагливіших любителів цього
ароматного напою.
            </p>
            <br/>
            <p>
              Наш асортимент включає ексклюзивні види кави, вибір якої
ми здійснюємо з великою увагою до якості та смакових особливостей. Ми
працюємо тільки з провідними кавоварнями та постачальниками, щоб
гарантувати вам лише найкращі продукти.
            </p>
            <br/>
            <p>
              Обирайте SuperCoffee – ваш провідний провідник у світі
смачної кави. Замовляйте прямо зараз та насолоджуйтеся кожним ковзанням
ароматного нектару, який ми привозимо для вас з усього світу.
            </p>
            <br/>
            <NavLink to="/products">
              <Button>Перейти до меню</Button>
            </NavLink>
          </div>
          <div className="hero-section-image">
            <figure>
              
    </figure>
  </div>
</div>
</div>
</Wrapper>
)
}

```

```

const Wrapper = styled.section`
  padding: 12rem 0 8rem 0;

```

```

  img {
    min-width: 10rem;
    height: 10rem;
  }

```

```

  .hero-section-data {
    p {
      margin: 2rem 0;
    }
  }

```

```

  h1 {
    text-transform: capitalize;
    font-weight: bold;
  }

```

```

  .intro-data {
    margin-bottom: 0;
  }
}

```

```

  .hero-section-image {
    width: 100%;
    height: auto;
    display: flex;
    justify-content: center;
    align-items: center;
  }
  figure {
    position: relative;

```

```

&::after {
  content: "";
  width: 60%;
  height: 80%;
  background-color: rgba(196,153,82, 0.4);
  position: absolute;
  left: 50%;
  top: -5rem;
  z-index: -1;
}
}
.img-style {
width: 80%;
height: auto;
}

@media (max-width: ${({ theme }) => theme.media.mobile}) {
.grid {
  gap: 10rem;
}

figure::after {
  content: "";
  width: 50%;
  height: 100%;
  left: 0;
  top: 10%;
  /* bottom: 10%; */
  background-color: rgba(196,153,82, 0.4);
}
}
`;

```

```
export default HeroSection;
```

```
// ./client/src/components/MyImage.jsx
```

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

```
const MyImage = ({ imgs }) => {
```

```
  return (
    <Wrapper>
      <div className="main-screen">
```

```

    <img src={imgs} alt="SuperCoffee" />
  </div>
</Wrapper>
);
};

```

```

const Wrapper = styled.section`
  display: grid;
  grid-template-columns: 0.7fr;
  ${" /* gap: 1rem; */"}

```

```

.grid {
  flex-direction: row;
  justify-items: center;
  align-items: center;
  width: 100%;
  gap: 1rem;
  /* order: 2; */

```

```

img {
  max-width: 100%;
  max-height: 100%;
  background-size: cover;
  object-fit: contain;
  cursor: pointer;
  box-shadow: ${({ theme }) => theme.colors.shadow};
}
}

```

```

.main-screen {
  display: grid;
  place-items: center;
  order: 1;
  img {
    max-width: 100%;
    height: auto;
    box-shadow: ${({ theme }) => theme.colors.shadow};
  }
}

```

```

.grid-four-column {
  grid-template-columns: 1fr;
  grid-template-rows: repeat(4, 1fr);
}

```

```

@media (max-width: ${({ theme }) => theme.media.mobile}) {

```

```

display: flex;
flex-direction: column;
order: 1;

.grid-four-column {
  grid-template-rows: 1fr;
  grid-template-columns: repeat(4, 1fr);
}
}
`;

export default MyImage;

// ./client/src/components/stars.scss

.bg-animation {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  pointer-events: none;
}

#stars {
  width: 1px;
  height: 1px;
  background: transparent;
  box-shadow: 117px 1613px #c49952, 1488px 635px #c49952, 944px 914px
#c49952, 647px 277px #c49952, 1792px 1205px #c49952, 656px 1517px
#c49952, 820px 1839px #c49952, 1153px 1400px #c49952, 870px 13px #c49952,
550px 702px #c49952, 1155px 1056px #c49952, 88px 1709px #c49952, 1450px
1090px #c49952, 1929px 457px #c49952, 1390px 905px #c49952, 1771px 269px
#c49952, 1741px 669px #c49952, 432px 64px #c49952, 563px 996px #c49952,
1918px 1873px #c49952, 1845px 1211px #c49952, 231px 1503px #c49952, 37px
220px #c49952, 1970px 495px #c49952, 1812px 925px #c49952, 67px 1398px
#c49952, 535px 279px #c49952, 1837px 829px #c49952, 1945px 685px #c49952,
1677px 1817px #c49952, 1317px 1415px #c49952, 1785px 905px #c49952,
1787px 1554px #c49952, 802px 1296px #c49952, 512px 1101px #c49952, 583px
1364px #c49952, 336px 558px #c49952, 979px 334px #c49952, 106px 792px
#c49952, 204px 34px #c49952, 1845px 1763px #c49952, 445px 1599px #c49952,
386px 453px #c49952, 471px 952px #c49952, 1466px 1676px #c49952, 1885px
303px #c49952, 51px 1717px #c49952, 1211px 299px #c49952, 1546px 1887px
#c49952, 1067px 33px #c49952, 1088px 1326px #c49952, 1938px 760px

```


#c49952, 470px 648px #c49952, 1213px 269px #c49952, 1767px 78px #c49952, 977px 976px #c49952, 1926px 175px #c49952, 722px 1512px #c49952, 945px 227px #c49952, 1811px 99px #c49952, 1912px 1406px #c49952, 1602px 1243px #c49952, 610px 449px #c49952, 654px 1393px #c49952, 1930px 1193px #c49952, 258px 1184px #c49952, 89px 265px #c49952, 824px 1494px #c49952, 1506px 1435px #c49952, 1027px 753px #c49952, 1px 1197px #c49952, 530px 1161px #c49952, 864px 1555px #c49952, 1610px 1604px #c49952, 1035px 1114px #c49952, 1456px 133px #c49952, 1196px 1253px #c49952, 361px 1037px #c49952, 834px 351px #c49952, 436px 1676px #c49952, 1194px 1007px #c49952, 1141px 647px #c49952, 319px 454px #c49952, 937px 1769px #c49952, 1872px 1013px #c49952, 733px 643px #c49952, 1250px 511px #c49952, 189px 296px #c49952, 1639px 163px #c49952, 1584px 336px #c49952, 1912px 1343px #c49952, 1298px 1307px #c49952, 1750px 902px #c49952, 1129px 845px #c49952, 1899px 1470px #c49952, 1427px 232px #c49952, 1391px 838px #c49952, 1225px 1819px #c49952, 190px 1366px #c49952, 1865px 518px #c49952, 203px 1383px #c49952, 1455px 614px #c49952, 423px 354px #c49952, 1678px 1790px #c49952, 241px 608px #c49952, 1089px 730px #c49952, 1342px 38px #c49952, 1848px 249px #c49952, 1874px 1785px #c49952, 1040px 1837px #c49952, 751px 261px #c49952, 510px 1975px #c49952, 52px 795px #c49952, 1786px 1310px #c49952, 498px 712px #c49952, 190px 375px #c49952, 1341px 722px #c49952, 43px 1394px #c49952, 1821px 1687px #c49952, 106px 130px #c49952, 1717px 1978px #c49952, 168px 151px #c49952, 183px 740px #c49952, 945px 1381px #c49952, 669px 1170px #c49952, 1285px 1816px #c49952, 110px 1217px #c49952, 1623px 813px #c49952, 869px 647px #c49952, 867px 582px #c49952, 735px 1240px #c49952, 519px 1896px #c49952, 132px 156px #c49952, 1649px 193px #c49952, 241px 1109px #c49952, 643px 484px #c49952, 574px 1282px #c49952, 1952px 564px #c49952, 1978px 145px #c49952, 329px 903px #c49952, 1674px 617px #c49952, 1978px 558px #c49952, 1808px 1715px #c49952, 1526px 1238px #c49952, 475px 1330px #c49952, 810px 425px #c49952, 1709px 634px #c49952, 1658px 336px #c49952, 425px 194px #c49952, 352px 96px #c49952, 148px 180px #c49952, 1139px 1046px #c49952, 1809px 1233px #c49952, 1669px 171px #c49952, 263px 1394px #c49952, 534px 715px #c49952, 396px 1008px #c49952, 589px 1445px #c49952, 1190px 381px #c49952, 1709px 279px #c49952, 520px 891px #c49952, 1136px 1867px #c49952, 1280px 1233px #c49952, 836px 296px #c49952, 1348px 646px #c49952, 1539px 913px #c49952, 423px 781px #c49952, 1271px 1805px #c49952, 696px 564px #c49952, 1549px 804px #c49952, 303px 1555px #c49952, 1449px 1903px #c49952, 66px 687px #c49952, 1164px 856px #c49952, 1958px 1326px #c49952, 125px 157px #c49952, 508px 1669px #c49952, 465px 725px #c49952, 1925px 1440px #c49952, 405px 793px #c49952, 278px 110px #c49952, 1084px 1065px #c49952, 1077px 705px #c49952, 663px 1844px #c49952, 734px 263px #c49952, 870px 1761px #c49952, 103px 1169px #c49952, 1506px 1295px #c49952, 1883px 926px #c49952, 335px 1361px #c49952, 1126px 1284px #c49952, 257px 1165px #c49952, 837px 580px #c49952, 1211px 1362px #c49952, 1137px 1380px #c49952, 135px 632px #c49952, 1491px 1965px

#c49952, 1098px 195px #c49952, 506px 417px #c49952, 693px 1243px #c49952, 622px 1862px #c49952, 1412px 1343px #c49952, 948px 1894px #c49952, 1315px 1363px #c49952, 754px 1098px #c49952, 1931px 930px #c49952, 1831px 342px #c49952, 1751px 1839px #c49952, 84px 775px #c49952, 1662px 1488px #c49952, 617px 1769px #c49952, 1869px 1292px #c49952, 963px 432px #c49952, 371px 1114px #c49952, 37px 642px #c49952, 21px 1184px #c49952, 602px 366px #c49952, 414px 524px #c49952, 282px 244px #c49952, 1689px 868px #c49952, 943px 681px #c49952, 898px 679px #c49952, 449px 1774px #c49952, 1678px 1313px #c49952, 475px 1811px #c49952, 1146px 1509px #c49952, 1151px 1863px #c49952, 1617px 846px #c49952, 82px 1077px #c49952, 324px 1317px #c49952, 1516px 885px #c49952, 1706px 1526px #c49952, 1925px 1180px #c49952, 553px 967px #c49952, 1072px 536px #c49952, 1715px 1816px #c49952, 185px 286px #c49952, 1362px 1600px #c49952, 628px 1938px #c49952, 1187px 412px #c49952, 569px 211px #c49952, 1959px 1356px #c49952, 1571px 105px #c49952, 319px 1111px #c49952, 36px 1364px #c49952, 502px 1788px #c49952, 1051px 1993px #c49952, 1617px 773px #c49952, 424px 1507px #c49952, 1623px 1955px #c49952, 307px 662px #c49952, 183px 1048px #c49952, 1919px 1453px #c49952, 1006px 1817px #c49952, 468px 673px #c49952, 1142px 1375px #c49952, 1228px 443px #c49952, 1734px 552px #c49952, 20px 1041px #c49952, 1783px 334px #c49952, 98px 1237px #c49952, 1356px 1940px #c49952, 853px 1779px #c49952, 1910px 560px #c49952, 1174px 1656px #c49952, 110px 1724px #c49952, 542px 1771px #c49952, 1758px 1931px #c49952, 1463px 1401px #c49952, 1155px 84px #c49952, 1504px 835px #c49952, 750px 322px #c49952, 407px 1900px #c49952, 1600px 1141px #c49952, 657px 886px #c49952, 526px 714px #c49952, 18px 836px #c49952, 1546px 1548px #c49952, 22px 469px #c49952, 594px 1466px #c49952, 1160px 1078px #c49952, 627px 1055px #c49952, 195px 699px #c49952, 1099px 684px #c49952, 530px 551px #c49952, 1160px 1325px #c49952, 894px 727px #c49952, 1157px 98px #c49952, 136px 1483px #c49952, 1875px 1975px #c49952, 1803px 566px #c49952, 318px 1073px #c49952, 1866px 1656px #c49952, 543px 414px #c49952, 719px 474px #c49952, 1115px 738px #c49952, 353px 875px #c49952, 184px 1938px #c49952, 1854px 1534px #c49952, 420px 1698px #c49952, 1480px 1550px #c49952, 522px 203px #c49952, 1897px 1904px #c49952, 975px 1708px #c49952, 1774px 602px #c49952, 1908px 274px #c49952, 61px 715px #c49952, 983px 1156px #c49952, 326px 1013px #c49952, 641px 290px #c49952, 1522px 120px #c49952, 405px 1637px #c49952, 1021px 1099px #c49952, 631px 1145px #c49952, 982px 1967px #c49952, 200px 651px #c49952, 795px 351px #c49952, 790px 1082px #c49952, 144px 1572px #c49952, 1542px 901px #c49952, 158px 1524px #c49952, 849px 1843px #c49952, 1807px 203px #c49952, 1747px 45px #c49952, 1603px 1738px #c49952, 617px 1966px #c49952, 342px 748px #c49952, 1779px 1173px #c49952, 1428px 152px #c49952, 589px 1998px #c49952, 1940px 1838px #c49952, 115px 272px #c49952, 1217px 1395px #c49952, 1402px 1491px #c49952, 1833px 1814px #c49952, 243px 966px #c49952, 319px 578px #c49952, 813px 364px #c49952, 669px 882px #c49952, 551px 134px #c49952, 1819px

920px #c49952, 740px 1826px #c49952, 1021px 952px #c49952, 1575px 453px
#c49952, 324px 419px #c49952, 929px 417px #c49952, 885px 1112px #c49952,
503px 187px #c49952, 1908px 362px #c49952, 1063px 1601px #c49952, 169px
1792px #c49952, 789px 963px #c49952, 1697px 948px #c49952, 1761px 1810px
#c49952, 1844px 1591px #c49952, 1709px 949px #c49952, 1402px 1396px
#c49952, 1037px 225px #c49952, 1832px 518px #c49952, 1728px 1782px
#c49952, 194px 1421px #c49952, 1395px 742px #c49952, 1478px 1325px
#c49952, 40px 593px #c49952, 1732px 117px #c49952, 51px 158px #c49952,
1598px 1672px #c49952, 701px 849px #c49952, 1403px 1979px #c49952, 145px
1414px #c49952, 550px 906px #c49952, 1366px 460px #c49952, 142px 1379px
#c49952, 34px 1864px #c49952, 1346px 308px #c49952, 293px 998px #c49952,
21px 1868px #c49952, 540px 1033px #c49952, 60px 746px #c49952, 1602px
1476px #c49952, 180px 804px #c49952, 345px 1982px #c49952, 1439px 640px
#c49952, 939px 1834px #c49952, 20px 432px #c49952, 492px 1549px #c49952,
109px 1579px #c49952, 1796px 1403px #c49952, 1079px 519px #c49952, 1664px
389px #c49952, 1627px 1061px #c49952, 823px 419px #c49952, 1399px 1882px
#c49952, 1906px 344px #c49952, 1189px 848px #c49952, 117px 882px #c49952,
1262px 33px #c49952, 1048px 434px #c49952, 1208px 1309px #c49952, 1616px
408px #c49952, 1833px 853px #c49952, 1433px 1656px #c49952, 811px 1861px
#c49952, 439px 1672px #c49952, 1105px 248px #c49952, 328px 1652px
#c49952, 13px 1658px #c49952, 685px 987px #c49952, 985px 403px #c49952,
1664px 1206px #c49952, 1993px 1925px #c49952, 440px 917px #c49952, 1835px
319px #c49952, 1404px 1907px #c49952, 624px 1443px #c49952, 843px 954px
#c49952, 478px 1567px #c49952, 895px 1602px #c49952, 1231px 871px
#c49952, 1267px 1646px #c49952, 475px 334px #c49952, 784px 796px #c49952,
1294px 199px #c49952, 109px 702px #c49952, 1978px 362px #c49952, 291px
940px #c49952, 971px 1343px #c49952, 74px 719px #c49952, 36px 715px
#c49952, 1007px 1423px #c49952, 860px 314px #c49952, 631px 177px #c49952,
1900px 1590px #c49952, 1239px 1348px #c49952, 1346px 1270px #c49952,
1934px 1475px #c49952, 1553px 559px #c49952, 588px 1969px #c49952, 670px
1269px #c49952, 1484px 376px #c49952, 20px 1424px #c49952, 1396px 8px
#c49952, 969px 244px #c49952, 1807px 538px #c49952, 1873px 891px #c49952,
636px 1142px #c49952, 1474px 1562px #c49952, 763px 350px #c49952, 663px
700px #c49952, 500px 1469px #c49952, 1302px 722px #c49952, 181px 291px
#c49952, 266px 893px #c49952, 1403px 654px #c49952, 492px 460px #c49952,
1503px 1369px #c49952, 23px 1662px #c49952, 349px 333px #c49952, 1435px
1017px #c49952, 1441px 705px #c49952, 1708px 1446px #c49952, 1041px 911px
#c49952, 1063px 780px #c49952, 1158px 1356px #c49952, 767px 1454px
#c49952, 1912px 797px #c49952, 1731px 1759px #c49952, 1378px 1390px
#c49952, 1815px 1364px #c49952, 960px 270px #c49952, 1343px 427px
#c49952, 275px 203px #c49952, 1319px 1092px #c49952, 1455px 770px
#c49952, 283px 1503px #c49952, 1505px 901px #c49952, 1738px 1561px
#c49952, 1526px 1935px #c49952, 1757px 669px #c49952, 1640px 620px
#c49952, 1750px 722px #c49952, 748px 66px #c49952, 1149px 540px #c49952,
159px 953px #c49952, 200px 1426px #c49952, 515px 1110px #c49952, 1552px

737px #c49952, 1094px 1459px #c49952, 778px 799px #c49952, 1031px 523px
#c49952, 743px 1825px #c49952, 1100px 882px #c49952, 1088px 1836px
#c49952, 255px 599px #c49952, 67px 1361px #c49952, 247px 1721px #c49952,
1722px 346px #c49952, 1822px 155px #c49952, 452px 1973px #c49952, 415px
1960px #c49952, 1109px 57px #c49952, 273px 1392px #c49952, 404px 1071px
#c49952, 1212px 353px #c49952, 370px 460px #c49952, 795px 1523px #c49952,
1932px 340px #c49952, 51px 1473px #c49952, 1268px 364px #c49952, 1512px
1862px #c49952, 1678px 1801px #c49952, 1796px 579px #c49952, 254px 251px
#c49952, 1466px 1717px #c49952, 893px 379px #c49952, 1153px 923px
#c49952, 913px 1808px #c49952, 791px 789px #c49952, 417px 1924px #c49952,
1336px 1599px #c49952, 1695px 908px #c49952, 1120px 114px #c49952, 493px
1949px #c49952, 68px 1905px #c49952, 969px 481px #c49952, 1420px 1095px
#c49952, 800px 1117px #c49952, 390px 234px #c49952, 356px 1644px #c49952,
1098px 1486px #c49952, 1360px 521px #c49952, 149px 1198px #c49952, 354px
747px #c49952, 1749px 487px #c49952, 470px 76px #c49952, 1672px 289px
#c49952, 1731px 545px #c49952, 1547px 1590px #c49952, 498px 692px
#c49952, 398px 1592px #c49952, 1846px 1237px #c49952, 1537px 1474px
#c49952, 1726px 1374px #c49952, 1922px 858px #c49952, 376px 321px
#c49952, 985px 227px #c49952, 234px 1421px #c49952, 760px 745px #c49952,
1990px 1132px #c49952, 1560px 1597px #c49952, 338px 1310px #c49952,
1924px 1664px #c49952, 547px 1747px #c49952, 1639px 1282px #c49952,
1202px 337px #c49952, 1985px 779px #c49952, 737px 456px #c49952, 89px
501px #c49952, 963px 792px #c49952, 655px 1447px #c49952, 1492px 1994px
#c49952, 1171px 254px #c49952, 892px 827px #c49952, 1735px 442px #c49952,
1474px 1187px #c49952, 846px 1518px #c49952, 557px 1805px #c49952, 738px
945px #c49952, 795px 68px #c49952, 663px 1956px #c49952, 1607px 290px
#c49952, 1524px 15px #c49952, 1097px 1911px #c49952, 157px 1939px
#c49952, 935px 1065px #c49952, 1809px 1708px #c49952, 164px 1157px
#c49952, 83px 855px #c49952, 625px 501px #c49952, 814px 398px #c49952,
552px 695px #c49952, 597px 1546px #c49952, 1237px 1417px #c49952, 628px
284px #c49952, 866px 767px #c49952, 1403px 1394px #c49952, 765px 1563px
#c49952, 1648px 109px #c49952, 1205px 1659px #c49952, 921px 1313px
#c49952, 1319px 243px #c49952, 18px 125px #c49952, 7px 777px #c49952,
181px 418px #c49952, 1062px 1892px #c49952, 382px 106px #c49952, 994px
751px #c49952, 964px 234px #c49952, 40px 118px #c49952, 278px 706px
#c49952, 1540px 1978px #c49952, 425px 1661px #c49952, 1050px 321px
#c49952, 735px 1729px #c49952, 1438px 260px #c49952, 1229px 1109px
#c49952, 186px 1041px #c49952, 244px 1184px #c49952, 392px 1472px
#c49952, 670px 1249px #c49952, 1260px 1443px #c49952, 1977px 1511px
#c49952, 1240px 773px #c49952, 303px 513px #c49952, 63px 1530px #c49952,
610px 792px #c49952, 1987px 1647px #c49952, 676px 1597px #c49952, 1740px
1244px #c49952, 816px 1661px #c49952, 351px 802px #c49952, 252px 1082px
#c49952, 31px 365px #c49952, 1453px 984px #c49952, 667px 1233px #c49952,
1247px 1800px #c49952, 839px 270px #c49952, 775px 913px #c49952, 1966px
1398px #c49952, 499px 813px #c49952, 922px 1982px #c49952, 1409px 1902px

#c49952, 1499px 1766px #c49952, 721px 899px #c49952, 788px 807px #c49952, 989px 1355px #c49952, 1248px 1274px #c49952, 849px 1091px #c49952, 1799px 1036px #c49952, 1486px 700px #c49952, 170px 1989px #c49952, 1275px 799px #c49952, 772px 2000px #c49952, 1642px 362px #c49952, 216px 940px #c49952, 1893px 281px #c49952, 1944px 1298px #c49952, 1294px 400px #c49952, 1523px 441px #c49952, 1829px 340px #c49952, 468px 170px #c49952, 1099px 967px #c49952, 1331px 665px #c49952, 1174px 1553px #c49952, 1567px 325px #c49952, 1028px 1399px #c49952, 781px 1451px #c49952, 1912px 1954px #c49952, 874px 873px #c49952, 1298px 1722px #c49952, 1879px 706px #c49952, 57px 1221px #c49952, 1116px 1432px #c49952, 48px 811px #c49952, 101px 916px #c49952, 677px 304px #c49952, 1203px 639px #c49952, 1391px 199px #c49952, 1895px 1988px #c49952, 1462px 1023px #c49952, 1216px 1751px #c49952, 1261px 663px #c49952, 1290px 1119px #c49952, 137px 1793px #c49952, 1052px 1470px #c49952, 1561px 226px #c49952, 1156px 402px #c49952, 709px 693px #c49952, 1040px 1911px #c49952, 1624px 1115px #c49952, 551px 475px #c49952, 416px 1090px #c49952, 1183px 451px #c49952, 58px 765px #c49952, 743px 1016px #c49952, 198px 369px #c49952, 1645px 1503px #c49952, 997px 22px #c49952, 1447px 1323px #c49952, 379px 883px #c49952, 1171px 1195px #c49952, 919px 133px #c49952, 1400px 517px #c49952, 725px 804px #c49952, 1600px 699px #c49952, 357px 581px #c49952, 266px 1713px #c49952, 848px 1749px #c49952, 1963px 1045px #c49952, 119px 1136px #c49952;

-webkit-animation: animStar 50s linear infinite;

-moz-animation: animStar 50s linear infinite;

-ms-animation: animStar 50s linear infinite;

animation: animStar 50s linear infinite;

}

#stars:after {

content: " ";

position: absolute;

top: 2000px;

width: 1px;

height: 1px;

background: transparent;

box-shadow: 117px 1613px #c49952, 1488px 635px #c49952, 944px 914px #c49952, 647px 277px #c49952, 1792px 1205px #c49952, 656px 1517px #c49952, 820px 1839px #c49952, 1153px 1400px #c49952, 870px 13px #c49952, 550px 702px #c49952, 1155px 1056px #c49952, 88px 1709px #c49952, 1450px 1090px #c49952, 1929px 457px #c49952, 1390px 905px #c49952, 1771px 269px #c49952, 1741px 669px #c49952, 432px 64px #c49952, 563px 996px #c49952, 1918px 1873px #c49952, 1845px 1211px #c49952, 231px 1503px #c49952, 37px 220px #c49952, 1970px 495px #c49952, 1812px 925px #c49952, 67px 1398px #c49952, 535px 279px #c49952, 1837px 829px #c49952, 1945px 685px #c49952, 1677px 1817px #c49952, 1317px 1415px #c49952, 1785px 905px #c49952,

1787px 1554px #c49952, 802px 1296px #c49952, 512px 1101px #c49952, 583px 1364px #c49952, 336px 558px #c49952, 979px 334px #c49952, 106px 792px #c49952, 204px 34px #c49952, 1845px 1763px #c49952, 445px 1599px #c49952, 386px 453px #c49952, 471px 952px #c49952, 1466px 1676px #c49952, 1885px 303px #c49952, 51px 1717px #c49952, 1211px 299px #c49952, 1546px 1887px #c49952, 1067px 33px #c49952, 1088px 1326px #c49952, 1938px 760px #c49952, 470px 648px #c49952, 1213px 269px #c49952, 1767px 78px #c49952, 977px 976px #c49952, 1926px 175px #c49952, 722px 1512px #c49952, 945px 227px #c49952, 1811px 99px #c49952, 1912px 1406px #c49952, 1602px 1243px #c49952, 610px 449px #c49952, 654px 1393px #c49952, 1930px 1193px #c49952, 258px 1184px #c49952, 89px 265px #c49952, 824px 1494px #c49952, 1506px 1435px #c49952, 1027px 753px #c49952, 1px 1197px #c49952, 530px 1161px #c49952, 864px 1555px #c49952, 1610px 1604px #c49952, 1035px 1114px #c49952, 1456px 133px #c49952, 1196px 1253px #c49952, 361px 1037px #c49952, 834px 351px #c49952, 436px 1676px #c49952, 1194px 1007px #c49952, 1141px 647px #c49952, 319px 454px #c49952, 937px 1769px #c49952, 1872px 1013px #c49952, 733px 643px #c49952, 1250px 511px #c49952, 189px 296px #c49952, 1639px 163px #c49952, 1584px 336px #c49952, 1912px 1343px #c49952, 1298px 1307px #c49952, 1750px 902px #c49952, 1129px 845px #c49952, 1899px 1470px #c49952, 1427px 232px #c49952, 1391px 838px #c49952, 1225px 1819px #c49952, 190px 1366px #c49952, 1865px 518px #c49952, 203px 1383px #c49952, 1455px 614px #c49952, 423px 354px #c49952, 1678px 1790px #c49952, 241px 608px #c49952, 1089px 730px #c49952, 1342px 38px #c49952, 1848px 249px #c49952, 1874px 1785px #c49952, 1040px 1837px #c49952, 751px 261px #c49952, 510px 1975px #c49952, 52px 795px #c49952, 1786px 1310px #c49952, 498px 712px #c49952, 190px 375px #c49952, 1341px 722px #c49952, 43px 1394px #c49952, 1821px 1687px #c49952, 106px 130px #c49952, 1717px 1978px #c49952, 168px 151px #c49952, 183px 740px #c49952, 945px 1381px #c49952, 669px 1170px #c49952, 1285px 1816px #c49952, 110px 1217px #c49952, 1623px 813px #c49952, 869px 647px #c49952, 867px 582px #c49952, 735px 1240px #c49952, 519px 1896px #c49952, 132px 156px #c49952, 1649px 193px #c49952, 241px 1109px #c49952, 643px 484px #c49952, 574px 1282px #c49952, 1952px 564px #c49952, 1978px 145px #c49952, 329px 903px #c49952, 1674px 617px #c49952, 1978px 558px #c49952, 1808px 1715px #c49952, 1526px 1238px #c49952, 475px 1330px #c49952, 810px 425px #c49952, 1709px 634px #c49952, 1658px 336px #c49952, 425px 194px #c49952, 352px 96px #c49952, 148px 180px #c49952, 1139px 1046px #c49952, 1809px 1233px #c49952, 1669px 171px #c49952, 263px 1394px #c49952, 534px 715px #c49952, 396px 1008px #c49952, 589px 1445px #c49952, 1190px 381px #c49952, 1709px 279px #c49952, 520px 891px #c49952, 1136px 1867px #c49952, 1280px 1233px #c49952, 836px 296px #c49952, 1348px 646px #c49952, 1539px 913px #c49952, 423px 781px #c49952, 1271px 1805px #c49952, 696px 564px #c49952, 1549px 804px #c49952, 303px 1555px #c49952, 1449px 1903px #c49952, 66px 687px #c49952, 1164px 856px #c49952, 1958px 1326px #c49952, 125px 157px #c49952, 508px 1669px #c49952, 465px 725px

#c49952, 1925px 1440px #c49952, 405px 793px #c49952, 278px 110px #c49952, 1084px 1065px #c49952, 1077px 705px #c49952, 663px 1844px #c49952, 734px 263px #c49952, 870px 1761px #c49952, 103px 1169px #c49952, 1506px 1295px #c49952, 1883px 926px #c49952, 335px 1361px #c49952, 1126px 1284px #c49952, 257px 1165px #c49952, 837px 580px #c49952, 1211px 1362px #c49952, 1137px 1380px #c49952, 135px 632px #c49952, 1491px 1965px #c49952, 1098px 195px #c49952, 506px 417px #c49952, 693px 1243px #c49952, 622px 1862px #c49952, 1412px 1343px #c49952, 948px 1894px #c49952, 1315px 1363px #c49952, 754px 1098px #c49952, 1931px 930px #c49952, 1831px 342px #c49952, 1751px 1839px #c49952, 84px 775px #c49952, 1662px 1488px #c49952, 617px 1769px #c49952, 1869px 1292px #c49952, 963px 432px #c49952, 371px 1114px #c49952, 37px 642px #c49952, 21px 1184px #c49952, 602px 366px #c49952, 414px 524px #c49952, 282px 244px #c49952, 1689px 868px #c49952, 943px 681px #c49952, 898px 679px #c49952, 449px 1774px #c49952, 1678px 1313px #c49952, 475px 1811px #c49952, 1146px 1509px #c49952, 1151px 1863px #c49952, 1617px 846px #c49952, 82px 1077px #c49952, 324px 1317px #c49952, 1516px 885px #c49952, 1706px 1526px #c49952, 1925px 1180px #c49952, 553px 967px #c49952, 1072px 536px #c49952, 1715px 1816px #c49952, 185px 286px #c49952, 1362px 1600px #c49952, 628px 1938px #c49952, 1187px 412px #c49952, 569px 211px #c49952, 1959px 1356px #c49952, 1571px 105px #c49952, 319px 1111px #c49952, 36px 1364px #c49952, 502px 1788px #c49952, 1051px 1993px #c49952, 1617px 773px #c49952, 424px 1507px #c49952, 1623px 1955px #c49952, 307px 662px #c49952, 183px 1048px #c49952, 1919px 1453px #c49952, 1006px 1817px #c49952, 468px 673px #c49952, 1142px 1375px #c49952, 1228px 443px #c49952, 1734px 552px #c49952, 20px 1041px #c49952, 1783px 334px #c49952, 98px 1237px #c49952, 1356px 1940px #c49952, 853px 1779px #c49952, 1910px 560px #c49952, 1174px 1656px #c49952, 110px 1724px #c49952, 542px 1771px #c49952, 1758px 1931px #c49952, 1463px 1401px #c49952, 1155px 84px #c49952, 1504px 835px #c49952, 750px 322px #c49952, 407px 1900px #c49952, 1600px 1141px #c49952, 657px 886px #c49952, 526px 714px #c49952, 18px 836px #c49952, 1546px 1548px #c49952, 22px 469px #c49952, 594px 1466px #c49952, 1160px 1078px #c49952, 627px 1055px #c49952, 195px 699px #c49952, 1099px 684px #c49952, 530px 551px #c49952, 1160px 1325px #c49952, 894px 727px #c49952, 1157px 98px #c49952, 136px 1483px #c49952, 1875px 1975px #c49952, 1803px 566px #c49952, 318px 1073px #c49952, 1866px 1656px #c49952, 543px 414px #c49952, 719px 474px #c49952, 1115px 738px #c49952, 353px 875px #c49952, 184px 1938px #c49952, 1854px 1534px #c49952, 420px 1698px #c49952, 1480px 1550px #c49952, 522px 203px #c49952, 1897px 1904px #c49952, 975px 1708px #c49952, 1774px 602px #c49952, 1908px 274px #c49952, 61px 715px #c49952, 983px 1156px #c49952, 326px 1013px #c49952, 641px 290px #c49952, 1522px 120px #c49952, 405px 1637px #c49952, 1021px 1099px #c49952, 631px 1145px #c49952, 982px 1967px #c49952, 200px 651px #c49952, 795px 351px #c49952, 790px 1082px #c49952, 144px 1572px #c49952, 1542px 901px #c49952, 158px 1524px #c49952, 849px

1843px #c49952, 1807px 203px #c49952, 1747px 45px #c49952, 1603px 1738px
#c49952, 617px 1966px #c49952, 342px 748px #c49952, 1779px 1173px
#c49952, 1428px 152px #c49952, 589px 1998px #c49952, 1940px 1838px
#c49952, 115px 272px #c49952, 1217px 1395px #c49952, 1402px 1491px
#c49952, 1833px 1814px #c49952, 243px 966px #c49952, 319px 578px #c49952,
813px 364px #c49952, 669px 882px #c49952, 551px 134px #c49952, 1819px
920px #c49952, 740px 1826px #c49952, 1021px 952px #c49952, 1575px 453px
#c49952, 324px 419px #c49952, 929px 417px #c49952, 885px 1112px #c49952,
503px 187px #c49952, 1908px 362px #c49952, 1063px 1601px #c49952, 169px
1792px #c49952, 789px 963px #c49952, 1697px 948px #c49952, 1761px 1810px
#c49952, 1844px 1591px #c49952, 1709px 949px #c49952, 1402px 1396px
#c49952, 1037px 225px #c49952, 1832px 518px #c49952, 1728px 1782px
#c49952, 194px 1421px #c49952, 1395px 742px #c49952, 1478px 1325px
#c49952, 40px 593px #c49952, 1732px 117px #c49952, 51px 158px #c49952,
1598px 1672px #c49952, 701px 849px #c49952, 1403px 1979px #c49952, 145px
1414px #c49952, 550px 906px #c49952, 1366px 460px #c49952, 142px 1379px
#c49952, 34px 1864px #c49952, 1346px 308px #c49952, 293px 998px #c49952,
21px 1868px #c49952, 540px 1033px #c49952, 60px 746px #c49952, 1602px
1476px #c49952, 180px 804px #c49952, 345px 1982px #c49952, 1439px 640px
#c49952, 939px 1834px #c49952, 20px 432px #c49952, 492px 1549px #c49952,
109px 1579px #c49952, 1796px 1403px #c49952, 1079px 519px #c49952, 1664px
389px #c49952, 1627px 1061px #c49952, 823px 419px #c49952, 1399px 1882px
#c49952, 1906px 344px #c49952, 1189px 848px #c49952, 117px 882px #c49952,
1262px 33px #c49952, 1048px 434px #c49952, 1208px 1309px #c49952, 1616px
408px #c49952, 1833px 853px #c49952, 1433px 1656px #c49952, 811px 1861px
#c49952, 439px 1672px #c49952, 1105px 248px #c49952, 328px 1652px
#c49952, 13px 1658px #c49952, 685px 987px #c49952, 985px 403px #c49952,
1664px 1206px #c49952, 1993px 1925px #c49952, 440px 917px #c49952, 1835px
319px #c49952, 1404px 1907px #c49952, 624px 1443px #c49952, 843px 954px
#c49952, 478px 1567px #c49952, 895px 1602px #c49952, 1231px 871px
#c49952, 1267px 1646px #c49952, 475px 334px #c49952, 784px 796px #c49952,
1294px 199px #c49952, 109px 702px #c49952, 1978px 362px #c49952, 291px
940px #c49952, 971px 1343px #c49952, 74px 719px #c49952, 36px 715px
#c49952, 1007px 1423px #c49952, 860px 314px #c49952, 631px 177px #c49952,
1900px 1590px #c49952, 1239px 1348px #c49952, 1346px 1270px #c49952,
1934px 1475px #c49952, 1553px 559px #c49952, 588px 1969px #c49952, 670px
1269px #c49952, 1484px 376px #c49952, 20px 1424px #c49952, 1396px 8px
#c49952, 969px 244px #c49952, 1807px 538px #c49952, 1873px 891px #c49952,
636px 1142px #c49952, 1474px 1562px #c49952, 763px 350px #c49952, 663px
700px #c49952, 500px 1469px #c49952, 1302px 722px #c49952, 181px 291px
#c49952, 266px 893px #c49952, 1403px 654px #c49952, 492px 460px #c49952,
1503px 1369px #c49952, 23px 1662px #c49952, 349px 333px #c49952, 1435px
1017px #c49952, 1441px 705px #c49952, 1708px 1446px #c49952, 1041px 911px
#c49952, 1063px 780px #c49952, 1158px 1356px #c49952, 767px 1454px
#c49952, 1912px 797px #c49952, 1731px 1759px #c49952, 1378px 1390px

#c49952, 1815px 1364px #c49952, 960px 270px #c49952, 1343px 427px
#c49952, 275px 203px #c49952, 1319px 1092px #c49952, 1455px 770px
#c49952, 283px 1503px #c49952, 1505px 901px #c49952, 1738px 1561px
#c49952, 1526px 1935px #c49952, 1757px 669px #c49952, 1640px 620px
#c49952, 1750px 722px #c49952, 748px 66px #c49952, 1149px 540px #c49952,
159px 953px #c49952, 200px 1426px #c49952, 515px 1110px #c49952, 1552px
737px #c49952, 1094px 1459px #c49952, 778px 799px #c49952, 1031px 523px
#c49952, 743px 1825px #c49952, 1100px 882px #c49952, 1088px 1836px
#c49952, 255px 599px #c49952, 67px 1361px #c49952, 247px 1721px #c49952,
1722px 346px #c49952, 1822px 155px #c49952, 452px 1973px #c49952, 415px
1960px #c49952, 1109px 57px #c49952, 273px 1392px #c49952, 404px 1071px
#c49952, 1212px 353px #c49952, 370px 460px #c49952, 795px 1523px #c49952,
1932px 340px #c49952, 51px 1473px #c49952, 1268px 364px #c49952, 1512px
1862px #c49952, 1678px 1801px #c49952, 1796px 579px #c49952, 254px 251px
#c49952, 1466px 1717px #c49952, 893px 379px #c49952, 1153px 923px
#c49952, 913px 1808px #c49952, 791px 789px #c49952, 417px 1924px #c49952,
1336px 1599px #c49952, 1695px 908px #c49952, 1120px 114px #c49952, 493px
1949px #c49952, 68px 1905px #c49952, 969px 481px #c49952, 1420px 1095px
#c49952, 800px 1117px #c49952, 390px 234px #c49952, 356px 1644px #c49952,
1098px 1486px #c49952, 1360px 521px #c49952, 149px 1198px #c49952, 354px
747px #c49952, 1749px 487px #c49952, 470px 76px #c49952, 1672px 289px
#c49952, 1731px 545px #c49952, 1547px 1590px #c49952, 498px 692px
#c49952, 398px 1592px #c49952, 1846px 1237px #c49952, 1537px 1474px
#c49952, 1726px 1374px #c49952, 1922px 858px #c49952, 376px 321px
#c49952, 985px 227px #c49952, 234px 1421px #c49952, 760px 745px #c49952,
1990px 1132px #c49952, 1560px 1597px #c49952, 338px 1310px #c49952,
1924px 1664px #c49952, 547px 1747px #c49952, 1639px 1282px #c49952,
1202px 337px #c49952, 1985px 779px #c49952, 737px 456px #c49952, 89px
501px #c49952, 963px 792px #c49952, 655px 1447px #c49952, 1492px 1994px
#c49952, 1171px 254px #c49952, 892px 827px #c49952, 1735px 442px #c49952,
1474px 1187px #c49952, 846px 1518px #c49952, 557px 1805px #c49952, 738px
945px #c49952, 795px 68px #c49952, 663px 1956px #c49952, 1607px 290px
#c49952, 1524px 15px #c49952, 1097px 1911px #c49952, 157px 1939px
#c49952, 935px 1065px #c49952, 1809px 1708px #c49952, 164px 1157px
#c49952, 83px 855px #c49952, 625px 501px #c49952, 814px 398px #c49952,
552px 695px #c49952, 597px 1546px #c49952, 1237px 1417px #c49952, 628px
284px #c49952, 866px 767px #c49952, 1403px 1394px #c49952, 765px 1563px
#c49952, 1648px 109px #c49952, 1205px 1659px #c49952, 921px 1313px
#c49952, 1319px 243px #c49952, 18px 125px #c49952, 7px 777px #c49952,
181px 418px #c49952, 1062px 1892px #c49952, 382px 106px #c49952, 994px
751px #c49952, 964px 234px #c49952, 40px 118px #c49952, 278px 706px
#c49952, 1540px 1978px #c49952, 425px 1661px #c49952, 1050px 321px
#c49952, 735px 1729px #c49952, 1438px 260px #c49952, 1229px 1109px
#c49952, 186px 1041px #c49952, 244px 1184px #c49952, 392px 1472px
#c49952, 670px 1249px #c49952, 1260px 1443px #c49952, 1977px 1511px

#c49952, 1240px 773px #c49952, 303px 513px #c49952, 63px 1530px #c49952, 610px 792px #c49952, 1987px 1647px #c49952, 676px 1597px #c49952, 1740px 1244px #c49952, 816px 1661px #c49952, 351px 802px #c49952, 252px 1082px #c49952, 31px 365px #c49952, 1453px 984px #c49952, 667px 1233px #c49952, 1247px 1800px #c49952, 839px 270px #c49952, 775px 913px #c49952, 1966px 1398px #c49952, 499px 813px #c49952, 922px 1982px #c49952, 1409px 1902px #c49952, 1499px 1766px #c49952, 721px 899px #c49952, 788px 807px #c49952, 989px 1355px #c49952, 1248px 1274px #c49952, 849px 1091px #c49952, 1799px 1036px #c49952, 1486px 700px #c49952, 170px 1989px #c49952, 1275px 799px #c49952, 772px 2000px #c49952, 1642px 362px #c49952, 216px 940px #c49952, 1893px 281px #c49952, 1944px 1298px #c49952, 1294px 400px #c49952, 1523px 441px #c49952, 1829px 340px #c49952, 468px 170px #c49952, 1099px 967px #c49952, 1331px 665px #c49952, 1174px 1553px #c49952, 1567px 325px #c49952, 1028px 1399px #c49952, 781px 1451px #c49952, 1912px 1954px #c49952, 874px 873px #c49952, 1298px 1722px #c49952, 1879px 706px #c49952, 57px 1221px #c49952, 1116px 1432px #c49952, 48px 811px #c49952, 101px 916px #c49952, 677px 304px #c49952, 1203px 639px #c49952, 1391px 199px #c49952, 1895px 1988px #c49952, 1462px 1023px #c49952, 1216px 1751px #c49952, 1261px 663px #c49952, 1290px 1119px #c49952, 137px 1793px #c49952, 1052px 1470px #c49952, 1561px 226px #c49952, 1156px 402px #c49952, 709px 693px #c49952, 1040px 1911px #c49952, 1624px 1115px #c49952, 551px 475px #c49952, 416px 1090px #c49952, 1183px 451px #c49952, 58px 765px #c49952, 743px 1016px #c49952, 198px 369px #c49952, 1645px 1503px #c49952, 997px 22px #c49952, 1447px 1323px #c49952, 379px 883px #c49952, 1171px 1195px #c49952, 919px 133px #c49952, 1400px 517px #c49952, 725px 804px #c49952, 1600px 699px #c49952, 357px 581px #c49952, 266px 1713px #c49952, 848px 1749px #c49952, 1963px 1045px #c49952, 119px 1136px #c49952;

}

#stars2 {
width: 2px;
height: 2px;
background: transparent;
box-shadow: 1117px 1306px #c49952, 1078px 1783px #c49952, 1179px 1085px #c49952, 1145px 920px #c49952, 422px 1233px #c49952, 387px 98px #c49952, 1153px 637px #c49952, 1084px 782px #c49952, 476px 453px #c49952, 926px 1306px #c49952, 60px 1086px #c49952, 753px 1575px #c49952, 272px 1684px #c49952, 1285px 750px #c49952, 1416px 1327px #c49952, 1931px 473px #c49952, 736px 1395px #c49952, 1816px 763px #c49952, 438px 879px #c49952, 665px 1902px #c49952, 1341px 677px #c49952, 1404px 1073px #c49952, 100px 597px #c49952, 357px 1689px #c49952, 1044px 1342px #c49952, 1954px 502px #c49952, 1192px 1308px #c49952, 540px 1239px #c49952, 1360px 552px #c49952, 89px 752px #c49952, 659px 1253px #c49952, 62px 517px #c49952, 1375px 1705px #c49952, 1343px 1511px #c49952, 1659px 1922px #c49952,

1560px 289px #c49952, 1362px 1799px #c49952, 1886px 1480px #c49952, 1718px 1885px #c49952, 824px 738px #c49952, 1060px 1370px #c49952, 1781px 1171px #c49952, 255px 273px #c49952, 1197px 120px #c49952, 213px 7px #c49952, 1226px 1920px #c49952, 1844px 207px #c49952, 1675px 970px #c49952, 1435px 1283px #c49952, 37px 353px #c49952, 59px 417px #c49952, 921px 1602px #c49952, 1549px 1490px #c49952, 638px 1845px #c49952, 1328px 198px #c49952, 1050px 1149px #c49952, 1884px 711px #c49952, 333px 263px #c49952, 342px 1508px #c49952, 1388px 1810px #c49952, 1377px 1558px #c49952, 890px 487px #c49952, 1081px 759px #c49952, 890px 1515px #c49952, 911px 1284px #c49952, 335px 735px #c49952, 1140px 549px #c49952, 1239px 1064px #c49952, 226px 71px #c49952, 1100px 1278px #c49952, 1851px 1805px #c49952, 1370px 1999px #c49952, 1008px 1122px #c49952, 785px 813px #c49952, 1358px 601px #c49952, 1833px 1305px #c49952, 1768px 1304px #c49952, 1303px 532px #c49952, 860px 598px #c49952, 1329px 593px #c49952, 1038px 1088px #c49952, 408px 405px #c49952, 965px 82px #c49952, 1483px 1438px #c49952, 310px 1479px #c49952, 1786px 1500px #c49952, 1866px 852px #c49952, 18px 1757px #c49952, 1473px 1004px #c49952, 1542px 1933px #c49952, 633px 1970px #c49952, 1334px 1713px #c49952, 175px 28px #c49952, 592px 894px #c49952, 121px 1162px #c49952, 1601px 1567px #c49952, 1095px 657px #c49952, 640px 1233px #c49952, 1073px 1255px #c49952, 840px 1087px #c49952, 718px 250px #c49952, 967px 709px #c49952, 731px 239px #c49952, 1623px 593px #c49952, 1058px 1820px #c49952, 516px 1898px #c49952, 666px 12px #c49952, 1997px 1382px #c49952, 112px 1690px #c49952, 687px 1309px #c49952, 63px 539px #c49952, 185px 1897px #c49952, 1055px 1691px #c49952, 435px 1517px #c49952, 1175px 1119px #c49952, 1721px 133px #c49952, 1212px 47px #c49952, 166px 18px #c49952, 1416px 1652px #c49952, 1409px 1745px #c49952, 1357px 1232px #c49952, 1677px 1998px #c49952, 448px 1415px #c49952, 705px 1736px #c49952, 1031px 1466px #c49952, 543px 1651px #c49952, 1592px 1888px #c49952, 1749px 1175px #c49952, 639px 1114px #c49952, 1591px 508px #c49952, 759px 1244px #c49952, 824px 380px #c49952, 942px 955px #c49952, 723px 732px #c49952, 113px 1369px #c49952, 203px 1739px #c49952, 868px 733px #c49952, 713px 971px #c49952, 341px 833px #c49952, 762px 824px #c49952, 1359px 310px #c49952, 1858px 1349px #c49952, 1531px 692px #c49952, 1075px 1512px #c49952, 1677px 142px #c49952, 1912px 1478px #c49952, 1810px 1078px #c49952, 426px 844px #c49952, 1426px 588px #c49952, 1909px 654px #c49952, 1107px 295px #c49952, 1351px 527px #c49952, 1393px 599px #c49952, 1379px 1068px #c49952, 228px 1846px #c49952, 1271px 374px #c49952, 1348px 612px #c49952, 7px 1301px #c49952, 1501px 1782px #c49952, 1795px 423px #c49952, 1475px 1918px #c49952, 1328px 1861px #c49952, 1624px 51px #c49952, 1791px 672px #c49952, 1594px 1467px #c49952, 1655px 1603px #c49952, 919px 850px #c49952, 523px 609px #c49952, 1196px 207px #c49952, 753px 410px #c49952, 686px 1097px #c49952, 1570px 133px #c49952, 1996px 1137px #c49952, 361px 116px #c49952, 1015px 462px #c49952, 76px 1143px #c49952, 491px 1818px #c49952, 1563px 795px #c49952, 982px 1721px #c49952, 831px 1204px

```

#c49952, 1737px 589px #c49952, 861px 1579px #c49952, 1666px 130px
#c49952, 698px 1799px #c49952, 726px 1519px #c49952, 109px 1208px
#c49952, 1184px 1057px #c49952, 835px 451px #c49952, 896px 594px #c49952,
35px 893px #c49952, 895px 542px #c49952, 706px 225px #c49952, 56px 1040px
#c49952, 1954px 108px #c49952, 1439px 1423px #c49952, 26px 1881px
#c49952, 802px 1564px #c49952, 273px 708px #c49952, 40px 31px #c49952,
859px 108px #c49952;
    -webkit-animation: animStar 100s linear infinite;
    -moz-animation: animStar 100s linear infinite;
    -ms-animation: animStar 100s linear infinite;
    animation: animStar 100s linear infinite;
}

```

```

#stars2:after {
    content: " ";
    position: absolute;
    top: 2000px;
    width: 2px;
    height: 2px;
    background: transparent;
    box-shadow: 1117px 1306px #c49952, 1078px 1783px #c49952, 1179px
1085px #c49952, 1145px 920px #c49952, 422px 1233px #c49952, 387px 98px
#c49952, 1153px 637px #c49952, 1084px 782px #c49952, 476px 453px #c49952,
926px 1306px #c49952, 60px 1086px #c49952, 753px 1575px #c49952, 272px
1684px #c49952, 1285px 750px #c49952, 1416px 1327px #c49952, 1931px 473px
#c49952, 736px 1395px #c49952, 1816px 763px #c49952, 438px 879px #c49952,
665px 1902px #c49952, 1341px 677px #c49952, 1404px 1073px #c49952, 100px
597px #c49952, 357px 1689px #c49952, 1044px 1342px #c49952, 1954px 502px
#c49952, 1192px 1308px #c49952, 540px 1239px #c49952, 1360px 552px
#c49952, 89px 752px #c49952, 659px 1253px #c49952, 62px 517px #c49952,
1375px 1705px #c49952, 1343px 1511px #c49952, 1659px 1922px #c49952,
1560px 289px #c49952, 1362px 1799px #c49952, 1886px 1480px #c49952,
1718px 1885px #c49952, 824px 738px #c49952, 1060px 1370px #c49952, 1781px
1171px #c49952, 255px 273px #c49952, 1197px 120px #c49952, 213px 7px
#c49952, 1226px 1920px #c49952, 1844px 207px #c49952, 1675px 970px
#c49952, 1435px 1283px #c49952, 37px 353px #c49952, 59px 417px #c49952,
921px 1602px #c49952, 1549px 1490px #c49952, 638px 1845px #c49952, 1328px
198px #c49952, 1050px 1149px #c49952, 1884px 711px #c49952, 333px 263px
#c49952, 342px 1508px #c49952, 1388px 1810px #c49952, 1377px 1558px
#c49952, 890px 487px #c49952, 1081px 759px #c49952, 890px 1515px #c49952,
911px 1284px #c49952, 335px 735px #c49952, 1140px 549px #c49952, 1239px
1064px #c49952, 226px 71px #c49952, 1100px 1278px #c49952, 1851px 1805px
#c49952, 1370px 1999px #c49952, 1008px 1122px #c49952, 785px 813px
#c49952, 1358px 601px #c49952, 1833px 1305px #c49952, 1768px 1304px
#c49952, 1303px 532px #c49952, 860px 598px #c49952, 1329px 593px #c49952,

```

1038px 1088px #c49952, 408px 405px #c49952, 965px 82px #c49952, 1483px 1438px #c49952, 310px 1479px #c49952, 1786px 1500px #c49952, 1866px 852px #c49952, 18px 1757px #c49952, 1473px 1004px #c49952, 1542px 1933px #c49952, 633px 1970px #c49952, 1334px 1713px #c49952, 175px 28px #c49952, 592px 894px #c49952, 121px 1162px #c49952, 1601px 1567px #c49952, 1095px 657px #c49952, 640px 1233px #c49952, 1073px 1255px #c49952, 840px 1087px #c49952, 718px 250px #c49952, 967px 709px #c49952, 731px 239px #c49952, 1623px 593px #c49952, 1058px 1820px #c49952, 516px 1898px #c49952, 666px 12px #c49952, 1997px 1382px #c49952, 112px 1690px #c49952, 687px 1309px #c49952, 63px 539px #c49952, 185px 1897px #c49952, 1055px 1691px #c49952, 435px 1517px #c49952, 1175px 1119px #c49952, 1721px 133px #c49952, 1212px 47px #c49952, 166px 18px #c49952, 1416px 1652px #c49952, 1409px 1745px #c49952, 1357px 1232px #c49952, 1677px 1998px #c49952, 448px 1415px #c49952, 705px 1736px #c49952, 1031px 1466px #c49952, 543px 1651px #c49952, 1592px 1888px #c49952, 1749px 1175px #c49952, 639px 1114px #c49952, 1591px 508px #c49952, 759px 1244px #c49952, 824px 380px #c49952, 942px 955px #c49952, 723px 732px #c49952, 113px 1369px #c49952, 203px 1739px #c49952, 868px 733px #c49952, 713px 971px #c49952, 341px 833px #c49952, 762px 824px #c49952, 1359px 310px #c49952, 1858px 1349px #c49952, 1531px 692px #c49952, 1075px 1512px #c49952, 1677px 142px #c49952, 1912px 1478px #c49952, 1810px 1078px #c49952, 426px 844px #c49952, 1426px 588px #c49952, 1909px 654px #c49952, 1107px 295px #c49952, 1351px 527px #c49952, 1393px 599px #c49952, 1379px 1068px #c49952, 228px 1846px #c49952, 1271px 374px #c49952, 1348px 612px #c49952, 7px 1301px #c49952, 1501px 1782px #c49952, 1795px 423px #c49952, 1475px 1918px #c49952, 1328px 1861px #c49952, 1624px 51px #c49952, 1791px 672px #c49952, 1594px 1467px #c49952, 1655px 1603px #c49952, 919px 850px #c49952, 523px 609px #c49952, 1196px 207px #c49952, 753px 410px #c49952, 686px 1097px #c49952, 1570px 133px #c49952, 1996px 1137px #c49952, 361px 116px #c49952, 1015px 462px #c49952, 76px 1143px #c49952, 491px 1818px #c49952, 1563px 795px #c49952, 982px 1721px #c49952, 831px 1204px #c49952, 1737px 589px #c49952, 861px 1579px #c49952, 1666px 130px #c49952, 698px 1799px #c49952, 726px 1519px #c49952, 109px 1208px #c49952, 1184px 1057px #c49952, 835px 451px #c49952, 896px 594px #c49952, 35px 893px #c49952, 895px 542px #c49952, 706px 225px #c49952, 56px 1040px #c49952, 1954px 108px #c49952, 1439px 1423px #c49952, 26px 1881px #c49952, 802px 1564px #c49952, 273px 708px #c49952, 40px 31px #c49952, 859px 108px #c49952;

}

```
#stars3 {
  width: 3px;
  height: 3px;
  background: transparent;
```

box-shadow: 940px 1360px #c49952, 1071px 539px #c49952, 1710px 1414px #c49952, 836px 299px #c49952, 1944px 1420px #c49952, 253px 1449px #c49952, 1257px 1250px #c49952, 1588px 1830px #c49952, 1077px 1204px #c49952, 273px 1081px #c49952, 1993px 766px #c49952, 1808px 479px #c49952, 917px 263px #c49952, 663px 1820px #c49952, 342px 1988px #c49952, 727px 1250px #c49952, 636px 1666px #c49952, 692px 1112px #c49952, 248px 1211px #c49952, 1422px 1121px #c49952, 881px 46px #c49952, 1531px 1977px #c49952, 1643px 1023px #c49952, 684px 1071px #c49952, 1142px 1873px #c49952, 292px 1313px #c49952, 256px 1237px #c49952, 89px 912px #c49952, 964px 1783px #c49952, 877px 760px #c49952, 1641px 1474px #c49952, 1492px 24px #c49952, 1776px 1642px #c49952, 183px 602px #c49952, 1998px 62px #c49952, 1560px 367px #c49952, 1333px 995px #c49952, 704px 1815px #c49952, 1809px 712px #c49952, 1503px 288px #c49952, 630px 556px #c49952, 1715px 125px #c49952, 353px 1878px #c49952, 975px 333px #c49952, 1740px 1409px #c49952, 1341px 1871px #c49952, 1279px 1064px #c49952, 169px 874px #c49952, 161px 528px #c49952, 1671px 1669px #c49952, 169px 632px #c49952, 547px 1724px #c49952, 1904px 110px #c49952, 679px 1670px #c49952, 196px 123px #c49952, 786px 871px #c49952, 1840px 324px #c49952, 356px 967px #c49952, 61px 549px #c49952, 99px 677px #c49952, 1719px 87px #c49952, 1713px 1990px #c49952, 1717px 1358px #c49952, 108px 1187px #c49952, 51px 869px #c49952, 1461px 902px #c49952, 1034px 891px #c49952, 962px 1881px #c49952, 1723px 595px #c49952, 479px 901px #c49952, 1546px 1823px #c49952, 285px 1208px #c49952, 1056px 347px #c49952, 261px 988px #c49952, 466px 990px #c49952, 1657px 648px #c49952, 1249px 933px #c49952, 1552px 1555px #c49952, 147px 62px #c49952, 292px 1157px #c49952, 1816px 423px #c49952, 1714px 757px #c49952, 1036px 961px #c49952, 1955px 710px #c49952, 1842px 516px #c49952, 479px 1870px #c49952, 1579px 1445px #c49952, 1225px 1309px #c49952, 1965px 566px #c49952, 1575px 1072px #c49952, 923px 329px #c49952, 651px 1514px #c49952, 865px 1100px #c49952, 782px 1873px #c49952, 115px 299px #c49952, 14px 1668px #c49952, 1666px 1817px #c49952, 1096px 1068px #c49952, 1462px 742px #c49952, 1384px 1750px #c49952;

-webkit-animation: animStar 150s linear infinite;

-moz-animation: animStar 150s linear infinite;

-ms-animation: animStar 150s linear infinite;

animation: animStar 150s linear infinite;

}

#stars3:after {

content: " ";

position: absolute;

top: 2000px;

width: 3px;

height: 3px;

background: transparent;

```

    box-shadow: 940px 1360px #c49952, 1071px 539px #c49952, 1710px 1414px
#c49952, 836px 299px #c49952, 1944px 1420px #c49952, 253px 1449px
#c49952, 1257px 1250px #c49952, 1588px 1830px #c49952, 1077px 1204px
#c49952, 273px 1081px #c49952, 1993px 766px #c49952, 1808px 479px
#c49952, 917px 263px #c49952, 663px 1820px #c49952, 342px 1988px #c49952,
727px 1250px #c49952, 636px 1666px #c49952, 692px 1112px #c49952, 248px
1211px #c49952, 1422px 1121px #c49952, 881px 46px #c49952, 1531px 1977px
#c49952, 1643px 1023px #c49952, 684px 1071px #c49952, 1142px 1873px
#c49952, 292px 1313px #c49952, 256px 1237px #c49952, 89px 912px #c49952,
964px 1783px #c49952, 877px 760px #c49952, 1641px 1474px #c49952, 1492px
24px #c49952, 1776px 1642px #c49952, 183px 602px #c49952, 1998px 62px
#c49952, 1560px 367px #c49952, 1333px 995px #c49952, 704px 1815px
#c49952, 1809px 712px #c49952, 1503px 288px #c49952, 630px 556px #c49952,
1715px 125px #c49952, 353px 1878px #c49952, 975px 333px #c49952, 1740px
1409px #c49952, 1341px 1871px #c49952, 1279px 1064px #c49952, 169px 874px
#c49952, 161px 528px #c49952, 1671px 1669px #c49952, 169px 632px #c49952,
547px 1724px #c49952, 1904px 110px #c49952, 679px 1670px #c49952, 196px
123px #c49952, 786px 871px #c49952, 1840px 324px #c49952, 356px 967px
#c49952, 61px 549px #c49952, 99px 677px #c49952, 1719px 87px #c49952,
1713px 1990px #c49952, 1717px 1358px #c49952, 108px 1187px #c49952, 51px
869px #c49952, 1461px 902px #c49952, 1034px 891px #c49952, 962px 1881px
#c49952, 1723px 595px #c49952, 479px 901px #c49952, 1546px 1823px
#c49952, 285px 1208px #c49952, 1056px 347px #c49952, 261px 988px #c49952,
466px 990px #c49952, 1657px 648px #c49952, 1249px 933px #c49952, 1552px
1555px #c49952, 147px 62px #c49952, 292px 1157px #c49952, 1816px 423px
#c49952, 1714px 757px #c49952, 1036px 961px #c49952, 1955px 710px
#c49952, 1842px 516px #c49952, 479px 1870px #c49952, 1579px 1445px
#c49952, 1225px 1309px #c49952, 1965px 566px #c49952, 1575px 1072px
#c49952, 923px 329px #c49952, 651px 1514px #c49952, 865px 1100px #c49952,
782px 1873px #c49952, 115px 299px #c49952, 14px 1668px #c49952, 1666px
1817px #c49952, 1096px 1068px #c49952, 1462px 742px #c49952, 1384px
1750px #c49952;
}

```

```

#stars4 {
    width: 1px;
    height: 1px;
    background: transparent;
    box-shadow: 233px 1976px #c49952, 1196px 1119px #c49952, 646px 740px
#c49952, 335px 645px #c49952, 1119px 1452px #c49952, 176px 1870px
#c49952, 639px 1711px #c49952, 647px 1388px #c49952, 1516px 1108px
#c49952, 464px 66px #c49952, 331px 344px #c49952, 772px 1189px #c49952,
1516px 1850px #c49952, 1500px 1463px #c49952, 1275px 876px #c49952,
1107px 645px #c49952, 977px 478px #c49952, 583px 1179px #c49952, 284px
395px #c49952, 1220px 461px #c49952, 1160px 249px #c49952, 196px 865px

```

#c49952, 670px 1915px #c49952, 1449px 382px #c49952, 1191px 546px
#c49952, 1329px 605px #c49952, 1945px 458px #c49952, 995px 749px #c49952,
1495px 861px #c49952, 708px 1731px #c49952, 348px 653px #c49952, 548px
1298px #c49952, 1606px 990px #c49952, 1049px 1204px #c49952, 253px 1501px
#c49952, 1154px 166px #c49952, 1087px 104px #c49952, 1034px 1161px
#c49952, 1681px 462px #c49952, 577px 1897px #c49952, 193px 1901px
#c49952, 1701px 1755px #c49952, 864px 1297px #c49952, 800px 1289px
#c49952, 676px 28px #c49952, 185px 1341px #c49952, 379px 1151px #c49952,
1224px 1725px #c49952, 280px 541px #c49952, 473px 1196px #c49952, 921px
1628px #c49952, 969px 432px #c49952, 1475px 758px #c49952, 1195px 993px
#c49952, 876px 1840px #c49952, 1274px 1689px #c49952, 1977px 1101px
#c49952, 837px 527px #c49952, 1785px 1610px #c49952, 1650px 1843px
#c49952, 1127px 1508px #c49952, 401px 1050px #c49952, 51px 1105px
#c49952, 545px 880px #c49952, 1786px 1672px #c49952, 318px 260px #c49952,
568px 254px #c49952, 1026px 1527px #c49952, 1242px 852px #c49952, 1785px
982px #c49952, 1318px 1071px #c49952, 398px 1061px #c49952, 1509px 257px
#c49952, 599px 928px #c49952, 1195px 1800px #c49952, 1254px 906px
#c49952, 141px 26px #c49952, 1384px 1502px #c49952, 476px 767px #c49952,
1973px 722px #c49952, 1339px 1031px #c49952, 778px 818px #c49952, 213px
1320px #c49952, 184px 221px #c49952, 983px 1911px #c49952, 923px 1439px
#c49952, 1936px 581px #c49952, 1105px 625px #c49952, 325px 729px #c49952,
1475px 204px #c49952, 1483px 1564px #c49952, 1327px 1272px #c49952,
1187px 1944px #c49952, 1945px 1471px #c49952, 116px 960px #c49952, 1660px
1610px #c49952, 412px 1022px #c49952, 1552px 1516px #c49952, 1517px
1892px #c49952, 306px 829px #c49952, 1416px 462px #c49952, 1575px 1460px
#c49952, 424px 1500px #c49952, 1530px 1169px #c49952, 1388px 1608px
#c49952, 185px 416px #c49952, 634px 1446px #c49952, 767px 479px #c49952,
71px 426px #c49952, 1937px 145px #c49952, 1955px 1312px #c49952, 1811px
611px #c49952, 1145px 569px #c49952, 1460px 676px #c49952, 131px 1858px
#c49952, 1557px 473px #c49952, 735px 130px #c49952, 112px 1531px #c49952,
1312px 305px #c49952, 409px 1032px #c49952, 149px 1964px #c49952, 535px
1215px #c49952, 1382px 630px #c49952, 1437px 1368px #c49952, 362px 1181px
#c49952, 388px 181px #c49952, 274px 1287px #c49952, 1858px 1414px
#c49952, 661px 1935px #c49952, 675px 1205px #c49952, 1829px 1725px
#c49952, 1937px 1145px #c49952, 237px 908px #c49952, 1059px 1185px
#c49952, 824px 1248px #c49952, 1167px 1730px #c49952, 180px 1961px
#c49952, 1663px 203px #c49952, 374px 221px #c49952, 724px 1883px #c49952,
970px 1362px #c49952, 832px 505px #c49952, 313px 233px #c49952, 1909px
597px #c49952, 434px 201px #c49952, 587px 995px #c49952, 1833px 623px
#c49952, 1464px 561px #c49952, 231px 593px #c49952, 1558px 1433px
#c49952, 1986px 1767px #c49952, 1753px 1728px #c49952, 1153px 1623px
#c49952, 249px 229px #c49952, 1503px 1186px #c49952, 1784px 137px
#c49952, 841px 403px #c49952, 1400px 354px #c49952, 197px 499px #c49952,
1188px 681px #c49952, 158px 391px #c49952, 443px 1099px #c49952, 723px
1445px #c49952, 1408px 1235px #c49952, 1908px 195px #c49952, 271px 891px

#c49952, 469px 1693px #c49952, 580px 11px #c49952, 1533px 70px #c49952, 859px 761px #c49952, 1510px 1844px #c49952, 421px 558px #c49952, 1132px 1453px #c49952, 757px 1987px #c49952, 212px 293px #c49952, 569px 323px #c49952, 1404px 1394px #c49952, 252px 1386px #c49952, 1668px 1857px #c49952, 123px 1684px #c49952, 105px 490px #c49952, 1083px 1769px #c49952, 1071px 1953px #c49952, 1271px 1159px #c49952, 699px 1491px #c49952, 1744px 1997px #c49952, 1868px 1973px #c49952, 1438px 1449px #c49952, 1222px 1921px #c49952, 1328px 1210px #c49952, 438px 873px #c49952, 809px 780px #c49952, 491px 1524px #c49952, 447px 1830px #c49952, 927px 1936px #c49952, 564px 691px #c49952, 1784px 1747px #c49952, 1978px 1722px #c49952, 1599px 1480px #c49952, 1276px 729px #c49952, 731px 1174px #c49952, 1586px 1711px #c49952, 451px 1340px #c49952, 1075px 1899px #c49952, 13px 575px #c49952, 309px 1340px #c49952, 981px 183px #c49952, 248px 1315px #c49952, 849px 80px #c49952, 1754px 1540px #c49952, 73px 1432px #c49952, 1208px 1828px #c49952, 65px 575px #c49952, 1098px 730px #c49952, 127px 1358px #c49952, 185px 19px #c49952, 1222px 1679px #c49952, 1122px 315px #c49952, 1906px 452px #c49952, 761px 284px #c49952, 813px 492px #c49952, 1344px 843px #c49952, 118px 1834px #c49952, 1620px 359px #c49952, 1755px 1246px #c49952, 299px 1076px #c49952, 1746px 158px #c49952, 6px 1635px #c49952, 143px 190px #c49952, 101px 468px #c49952, 137px 971px #c49952, 1221px 1929px #c49952, 1752px 650px #c49952, 1635px 1761px #c49952, 1522px 833px #c49952, 908px 153px #c49952, 1044px 350px #c49952, 1151px 1940px #c49952, 822px 210px #c49952, 1774px 310px #c49952, 796px 1447px #c49952, 1069px 1903px #c49952, 217px 565px #c49952, 662px 1370px #c49952, 1876px 1570px #c49952, 847px 46px #c49952, 1042px 1689px #c49952, 1584px 1434px #c49952, 1791px 908px #c49952, 973px 908px #c49952, 793px 747px #c49952, 122px 483px #c49952, 1137px 1374px #c49952, 1757px 1791px #c49952, 513px 225px #c49952, 63px 731px #c49952, 1179px 1926px #c49952, 346px 18px #c49952, 589px 175px #c49952, 87px 302px #c49952, 380px 1295px #c49952, 450px 921px #c49952, 1667px 1973px #c49952, 1495px 1373px #c49952, 1462px 1850px #c49952, 540px 288px #c49952, 1208px 1051px #c49952, 1554px 1095px #c49952, 1009px 1516px #c49952, 181px 572px #c49952, 165px 387px #c49952, 549px 1835px #c49952, 960px 16px #c49952, 1360px 403px #c49952, 1251px 43px #c49952, 1905px 1813px #c49952, 1106px 866px #c49952, 1809px 277px #c49952, 1828px 1720px #c49952, 295px 1610px #c49952, 523px 166px #c49952, 1069px 692px #c49952, 1292px 217px #c49952, 11px 1721px #c49952, 99px 1045px #c49952, 51px 1584px #c49952, 1053px 266px #c49952, 1287px 1235px #c49952, 747px 1722px #c49952, 1542px 736px #c49952, 1256px 18px #c49952, 102px 609px #c49952, 586px 1339px #c49952, 1843px 1697px #c49952, 824px 1687px #c49952, 1124px 882px #c49952, 395px 501px #c49952, 1456px 672px #c49952, 1472px 1648px #c49952, 1326px 1164px #c49952, 777px 1672px #c49952, 81px 345px #c49952, 91px 386px #c49952, 243px 411px #c49952, 1560px 90px #c49952, 6px 1771px #c49952, 1601px 616px #c49952, 1220px 1808px #c49952, 1160px 836px #c49952, 246px 1777px #c49952, 456px 863px #c49952, 97px 1138px #c49952,

1811px 942px #c49952, 213px 414px #c49952, 891px 392px #c49952, 1044px
927px #c49952, 1856px 216px #c49952, 957px 347px #c49952, 1486px 406px
#c49952, 838px 912px #c49952, 803px 361px #c49952, 564px 826px #c49952,
1597px 949px #c49952, 1206px 289px #c49952, 33px 1035px #c49952, 1762px
1377px #c49952, 789px 1815px #c49952, 1594px 1342px #c49952, 1668px 880px
#c49952, 1539px 1581px #c49952, 1547px 53px #c49952, 861px 1433px
#c49952, 693px 1618px #c49952, 1762px 782px #c49952, 1568px 682px
#c49952, 1126px 1762px #c49952, 1242px 134px #c49952, 495px 959px
#c49952, 1606px 219px #c49952, 1878px 1415px #c49952, 1652px 797px
#c49952, 782px 1903px #c49952, 1774px 1133px #c49952, 1430px 408px
#c49952, 265px 394px #c49952, 890px 336px #c49952, 1051px 311px #c49952,
461px 1559px #c49952, 1931px 91px #c49952, 1160px 380px #c49952, 1442px
1058px #c49952, 1157px 364px #c49952, 586px 227px #c49952, 1365px 715px
#c49952, 1658px 1655px #c49952, 1923px 1664px #c49952, 1023px 1844px
#c49952, 1939px 1367px #c49952, 1203px 1305px #c49952, 359px 642px
#c49952, 1056px 425px #c49952, 787px 202px #c49952, 1609px 1850px
#c49952, 1964px 200px #c49952, 1537px 586px #c49952, 1589px 903px
#c49952, 1063px 1694px #c49952, 760px 1185px #c49952, 597px 1396px
#c49952, 294px 452px #c49952, 433px 818px #c49952, 199px 840px #c49952,
1332px 1937px #c49952, 169px 1907px #c49952, 591px 834px #c49952, 1716px
1032px #c49952, 45px 1879px #c49952, 686px 1469px #c49952, 1520px 475px
#c49952, 1122px 859px #c49952, 973px 1541px #c49952, 269px 477px #c49952,
1390px 716px #c49952, 1791px 783px #c49952, 824px 2000px #c49952, 1211px
1717px #c49952, 1008px 1587px #c49952, 1422px 204px #c49952, 234px 556px
#c49952, 506px 550px #c49952, 942px 1670px #c49952, 397px 853px #c49952,
599px 795px #c49952, 762px 1926px #c49952, 1202px 1424px #c49952, 135px
1316px #c49952, 1442px 1692px #c49952, 977px 652px #c49952, 564px 1648px
#c49952, 997px 1474px #c49952, 67px 1366px #c49952, 1860px 1451px
#c49952, 1105px 772px #c49952, 1886px 1396px #c49952, 1510px 658px
#c49952, 976px 1544px #c49952, 894px 543px #c49952, 1098px 1189px
#c49952, 690px 77px #c49952, 770px 733px #c49952, 557px 1403px #c49952,
1758px 1623px #c49952, 1341px 812px #c49952, 699px 967px #c49952, 277px
866px #c49952, 1526px 1828px #c49952, 8px 977px #c49952, 1707px 952px
#c49952, 12px 1900px #c49952, 72px 921px #c49952, 496px 1067px #c49952,
1288px 1749px #c49952, 273px 984px #c49952, 1197px 1991px #c49952, 242px
789px #c49952, 903px 1035px #c49952, 480px 1492px #c49952, 102px 1331px
#c49952, 738px 1343px #c49952, 560px 1475px #c49952, 367px 846px #c49952,
1420px 962px #c49952, 1976px 892px #c49952, 1911px 1763px #c49952, 1639px
1002px #c49952, 437px 1522px #c49952, 1906px 1025px #c49952, 730px 1364px
#c49952, 1127px 521px #c49952, 1401px 1792px #c49952, 1954px 1066px
#c49952, 232px 250px #c49952, 1685px 660px #c49952, 1011px 999px #c49952,
1970px 790px #c49952, 750px 499px #c49952, 1738px 660px #c49952, 1621px
1849px #c49952, 446px 52px #c49952, 1055px 1396px #c49952, 1165px 1497px
#c49952, 1740px 1425px #c49952, 1012px 1920px #c49952, 1258px 1560px
#c49952, 1020px 1152px #c49952, 362px 673px #c49952, 1065px 975px

#c49952, 582px 755px #c49952, 1271px 1479px #c49952, 719px 548px #c49952, 1602px 879px #c49952, 590px 499px #c49952, 721px 1412px #c49952, 1180px 113px #c49952, 1801px 1961px #c49952, 589px 941px #c49952, 883px 476px #c49952, 214px 890px #c49952, 1028px 892px #c49952, 1107px 1832px #c49952, 944px 361px #c49952, 480px 1453px #c49952, 1466px 683px #c49952, 981px 745px #c49952, 1968px 828px #c49952, 657px 1830px #c49952, 11px 1338px #c49952, 179px 730px #c49952, 1713px 197px #c49952, 51px 955px #c49952, 1243px 319px #c49952, 1175px 624px #c49952, 446px 46px #c49952, 5px 1158px #c49952, 82px 1352px #c49952, 1877px 402px #c49952, 708px 1778px #c49952, 903px 1625px #c49952, 1824px 352px #c49952, 1229px 140px #c49952, 1518px 24px #c49952, 1017px 512px #c49952, 515px 699px #c49952, 295px 265px #c49952, 69px 1773px #c49952, 1640px 1163px #c49952, 536px 342px #c49952, 970px 1766px #c49952, 560px 1416px #c49952, 577px 193px #c49952, 469px 9px #c49952, 466px 276px #c49952, 711px 853px #c49952, 401px 685px #c49952, 85px 506px #c49952, 865px 558px #c49952, 631px 105px #c49952, 887px 866px #c49952, 1704px 1001px #c49952, 1051px 1199px #c49952, 275px 1909px #c49952, 1462px 829px #c49952, 375px 1057px #c49952, 1531px 1501px #c49952, 205px 403px #c49952, 33px 1869px #c49952, 967px 1176px #c49952, 376px 863px #c49952, 1769px 1545px #c49952, 535px 51px #c49952, 1972px 1569px #c49952, 1773px 960px #c49952, 487px 620px #c49952, 1660px 687px #c49952, 1632px 972px #c49952, 1362px 42px #c49952, 479px 1655px #c49952, 1531px 1808px #c49952, 1450px 1412px #c49952, 1549px 170px #c49952, 1904px 1305px #c49952, 1209px 48px #c49952, 1933px 820px #c49952, 1623px 595px #c49952, 48px 643px #c49952, 179px 1754px #c49952, 589px 1032px #c49952, 1199px 356px #c49952, 1755px 1418px #c49952, 780px 1174px #c49952, 1905px 758px #c49952, 1567px 713px #c49952, 1372px 705px #c49952, 456px 654px #c49952, 759px 690px #c49952, 452px 673px #c49952, 993px 1610px #c49952, 1271px 188px #c49952, 343px 1750px #c49952, 1943px 1735px #c49952, 1717px 853px #c49952, 1247px 303px #c49952, 1314px 1895px #c49952, 1203px 489px #c49952, 741px 469px #c49952, 4px 246px #c49952, 1515px 115px #c49952, 606px 218px #c49952, 1966px 1471px #c49952, 177px 87px #c49952, 1575px 588px #c49952, 1136px 1386px #c49952, 70px 1868px #c49952, 1053px 18px #c49952, 1124px 721px #c49952, 1748px 1181px #c49952, 191px 1387px #c49952, 1931px 840px #c49952, 1088px 1603px #c49952, 634px 1255px #c49952, 814px 1434px #c49952, 585px 64px #c49952, 1074px 1618px #c49952, 1692px 761px #c49952, 651px 643px #c49952, 193px 335px #c49952, 1103px 1447px #c49952, 491px 1142px #c49952, 521px 408px #c49952, 536px 340px #c49952, 411px 1091px #c49952, 1646px 193px #c49952, 1595px 1285px #c49952, 870px 1349px #c49952, 1085px 1013px #c49952, 204px 1864px #c49952, 1359px 299px #c49952, 807px 964px #c49952, 219px 509px #c49952, 36px 1227px #c49952, 702px 1873px #c49952, 1471px 934px #c49952, 1763px 792px #c49952, 973px 1957px #c49952, 987px 68px #c49952, 593px 1282px #c49952, 1900px 607px #c49952, 407px 1659px #c49952, 587px 17px #c49952, 632px 158px #c49952; -webkit-animation: animStar 600s linear infinite;

```

-moz-animation: animStar 600s linear infinite;
-ms-animation: animStar 600s linear infinite;
animation: animStar 600s linear infinite;
}

#stars4:after {
  content: " ";
  position: absolute;
  top: 2000px;
  width: 1px;
  height: 1px;
  background: transparent;
  box-shadow: 233px 1976px #c49952, 1196px 1119px #c49952, 646px 740px
#c49952, 335px 645px #c49952, 1119px 1452px #c49952, 176px 1870px
#c49952, 639px 1711px #c49952, 647px 1388px #c49952, 1516px 1108px
#c49952, 464px 66px #c49952, 331px 344px #c49952, 772px 1189px #c49952,
1516px 1850px #c49952, 1500px 1463px #c49952, 1275px 876px #c49952,
1107px 645px #c49952, 977px 478px #c49952, 583px 1179px #c49952, 284px
395px #c49952, 1220px 461px #c49952, 1160px 249px #c49952, 196px 865px
#c49952, 670px 1915px #c49952, 1449px 382px #c49952, 1191px 546px
#c49952, 1329px 605px #c49952, 1945px 458px #c49952, 995px 749px #c49952,
1495px 861px #c49952, 708px 1731px #c49952, 348px 653px #c49952, 548px
1298px #c49952, 1606px 990px #c49952, 1049px 1204px #c49952, 253px 1501px
#c49952, 1154px 166px #c49952, 1087px 104px #c49952, 1034px 1161px
#c49952, 1681px 462px #c49952, 577px 1897px #c49952, 193px 1901px
#c49952, 1701px 1755px #c49952, 864px 1297px #c49952, 800px 1289px
#c49952, 676px 28px #c49952, 185px 1341px #c49952, 379px 1151px #c49952,
1224px 1725px #c49952, 280px 541px #c49952, 473px 1196px #c49952, 921px
1628px #c49952, 969px 432px #c49952, 1475px 758px #c49952, 1195px 993px
#c49952, 876px 1840px #c49952, 1274px 1689px #c49952, 1977px 1101px
#c49952, 837px 527px #c49952, 1785px 1610px #c49952, 1650px 1843px
#c49952, 1127px 1508px #c49952, 401px 1050px #c49952, 51px 1105px
#c49952, 545px 880px #c49952, 1786px 1672px #c49952, 318px 260px #c49952,
568px 254px #c49952, 1026px 1527px #c49952, 1242px 852px #c49952, 1785px
982px #c49952, 1318px 1071px #c49952, 398px 1061px #c49952, 1509px 257px
#c49952, 599px 928px #c49952, 1195px 1800px #c49952, 1254px 906px
#c49952, 141px 26px #c49952, 1384px 1502px #c49952, 476px 767px #c49952,
1973px 722px #c49952, 1339px 1031px #c49952, 778px 818px #c49952, 213px
1320px #c49952, 184px 221px #c49952, 983px 1911px #c49952, 923px 1439px
#c49952, 1936px 581px #c49952, 1105px 625px #c49952, 325px 729px #c49952,
1475px 204px #c49952, 1483px 1564px #c49952, 1327px 1272px #c49952,
1187px 1944px #c49952, 1945px 1471px #c49952, 116px 960px #c49952, 1660px
1610px #c49952, 412px 1022px #c49952, 1552px 1516px #c49952, 1517px
1892px #c49952, 306px 829px #c49952, 1416px 462px #c49952, 1575px 1460px
#c49952, 424px 1500px #c49952, 1530px 1169px #c49952, 1388px 1608px

```

#c49952, 185px 416px #c49952, 634px 1446px #c49952, 767px 479px #c49952, 71px 426px #c49952, 1937px 145px #c49952, 1955px 1312px #c49952, 1811px 611px #c49952, 1145px 569px #c49952, 1460px 676px #c49952, 131px 1858px #c49952, 1557px 473px #c49952, 735px 130px #c49952, 112px 1531px #c49952, 1312px 305px #c49952, 409px 1032px #c49952, 149px 1964px #c49952, 535px 1215px #c49952, 1382px 630px #c49952, 1437px 1368px #c49952, 362px 1181px #c49952, 388px 181px #c49952, 274px 1287px #c49952, 1858px 1414px #c49952, 661px 1935px #c49952, 675px 1205px #c49952, 1829px 1725px #c49952, 1937px 1145px #c49952, 237px 908px #c49952, 1059px 1185px #c49952, 824px 1248px #c49952, 1167px 1730px #c49952, 180px 1961px #c49952, 1663px 203px #c49952, 374px 221px #c49952, 724px 1883px #c49952, 970px 1362px #c49952, 832px 505px #c49952, 313px 233px #c49952, 1909px 597px #c49952, 434px 201px #c49952, 587px 995px #c49952, 1833px 623px #c49952, 1464px 561px #c49952, 231px 593px #c49952, 1558px 1433px #c49952, 1986px 1767px #c49952, 1753px 1728px #c49952, 1153px 1623px #c49952, 249px 229px #c49952, 1503px 1186px #c49952, 1784px 137px #c49952, 841px 403px #c49952, 1400px 354px #c49952, 197px 499px #c49952, 1188px 681px #c49952, 158px 391px #c49952, 443px 1099px #c49952, 723px 1445px #c49952, 1408px 1235px #c49952, 1908px 195px #c49952, 271px 891px #c49952, 469px 1693px #c49952, 580px 11px #c49952, 1533px 70px #c49952, 859px 761px #c49952, 1510px 1844px #c49952, 421px 558px #c49952, 1132px 1453px #c49952, 757px 1987px #c49952, 212px 293px #c49952, 569px 323px #c49952, 1404px 1394px #c49952, 252px 1386px #c49952, 1668px 1857px #c49952, 123px 1684px #c49952, 105px 490px #c49952, 1083px 1769px #c49952, 1071px 1953px #c49952, 1271px 1159px #c49952, 699px 1491px #c49952, 1744px 1997px #c49952, 1868px 1973px #c49952, 1438px 1449px #c49952, 1222px 1921px #c49952, 1328px 1210px #c49952, 438px 873px #c49952, 809px 780px #c49952, 491px 1524px #c49952, 447px 1830px #c49952, 927px 1936px #c49952, 564px 691px #c49952, 1784px 1747px #c49952, 1978px 1722px #c49952, 1599px 1480px #c49952, 1276px 729px #c49952, 731px 1174px #c49952, 1586px 1711px #c49952, 451px 1340px #c49952, 1075px 1899px #c49952, 13px 575px #c49952, 309px 1340px #c49952, 981px 183px #c49952, 248px 1315px #c49952, 849px 80px #c49952, 1754px 1540px #c49952, 73px 1432px #c49952, 1208px 1828px #c49952, 65px 575px #c49952, 1098px 730px #c49952, 127px 1358px #c49952, 185px 19px #c49952, 1222px 1679px #c49952, 1122px 315px #c49952, 1906px 452px #c49952, 761px 284px #c49952, 813px 492px #c49952, 1344px 843px #c49952, 118px 1834px #c49952, 1620px 359px #c49952, 1755px 1246px #c49952, 299px 1076px #c49952, 1746px 158px #c49952, 6px 1635px #c49952, 143px 190px #c49952, 101px 468px #c49952, 137px 971px #c49952, 1221px 1929px #c49952, 1752px 650px #c49952, 1635px 1761px #c49952, 1522px 833px #c49952, 908px 153px #c49952, 1044px 350px #c49952, 1151px 1940px #c49952, 822px 210px #c49952, 1774px 310px #c49952, 796px 1447px #c49952, 1069px 1903px #c49952, 217px 565px #c49952, 662px 1370px #c49952, 1876px 1570px #c49952, 847px 46px #c49952, 1042px 1689px #c49952, 1584px 1434px #c49952, 1791px 908px #c49952, 973px

908px #c49952, 793px 747px #c49952, 122px 483px #c49952, 1137px 1374px
#c49952, 1757px 1791px #c49952, 513px 225px #c49952, 63px 731px #c49952,
1179px 1926px #c49952, 346px 18px #c49952, 589px 175px #c49952, 87px
302px #c49952, 380px 1295px #c49952, 450px 921px #c49952, 1667px 1973px
#c49952, 1495px 1373px #c49952, 1462px 1850px #c49952, 540px 288px
#c49952, 1208px 1051px #c49952, 1554px 1095px #c49952, 1009px 1516px
#c49952, 181px 572px #c49952, 165px 387px #c49952, 549px 1835px #c49952,
960px 16px #c49952, 1360px 403px #c49952, 1251px 43px #c49952, 1905px
1813px #c49952, 1106px 866px #c49952, 1809px 277px #c49952, 1828px 1720px
#c49952, 295px 1610px #c49952, 523px 166px #c49952, 1069px 692px #c49952,
1292px 217px #c49952, 11px 1721px #c49952, 99px 1045px #c49952, 51px
1584px #c49952, 1053px 266px #c49952, 1287px 1235px #c49952, 747px 1722px
#c49952, 1542px 736px #c49952, 1256px 18px #c49952, 102px 609px #c49952,
586px 1339px #c49952, 1843px 1697px #c49952, 824px 1687px #c49952, 1124px
882px #c49952, 395px 501px #c49952, 1456px 672px #c49952, 1472px 1648px
#c49952, 1326px 1164px #c49952, 777px 1672px #c49952, 81px 345px #c49952,
91px 386px #c49952, 243px 411px #c49952, 1560px 90px #c49952, 6px 1771px
#c49952, 1601px 616px #c49952, 1220px 1808px #c49952, 1160px 836px
#c49952, 246px 1777px #c49952, 456px 863px #c49952, 97px 1138px #c49952,
1811px 942px #c49952, 213px 414px #c49952, 891px 392px #c49952, 1044px
927px #c49952, 1856px 216px #c49952, 957px 347px #c49952, 1486px 406px
#c49952, 838px 912px #c49952, 803px 361px #c49952, 564px 826px #c49952,
1597px 949px #c49952, 1206px 289px #c49952, 33px 1035px #c49952, 1762px
1377px #c49952, 789px 1815px #c49952, 1594px 1342px #c49952, 1668px 880px
#c49952, 1539px 1581px #c49952, 1547px 53px #c49952, 861px 1433px
#c49952, 693px 1618px #c49952, 1762px 782px #c49952, 1568px 682px
#c49952, 1126px 1762px #c49952, 1242px 134px #c49952, 495px 959px
#c49952, 1606px 219px #c49952, 1878px 1415px #c49952, 1652px 797px
#c49952, 782px 1903px #c49952, 1774px 1133px #c49952, 1430px 408px
#c49952, 265px 394px #c49952, 890px 336px #c49952, 1051px 311px #c49952,
461px 1559px #c49952, 1931px 91px #c49952, 1160px 380px #c49952, 1442px
1058px #c49952, 1157px 364px #c49952, 586px 227px #c49952, 1365px 715px
#c49952, 1658px 1655px #c49952, 1923px 1664px #c49952, 1023px 1844px
#c49952, 1939px 1367px #c49952, 1203px 1305px #c49952, 359px 642px
#c49952, 1056px 425px #c49952, 787px 202px #c49952, 1609px 1850px
#c49952, 1964px 200px #c49952, 1537px 586px #c49952, 1589px 903px
#c49952, 1063px 1694px #c49952, 760px 1185px #c49952, 597px 1396px
#c49952, 294px 452px #c49952, 433px 818px #c49952, 199px 840px #c49952,
1332px 1937px #c49952, 169px 1907px #c49952, 591px 834px #c49952, 1716px
1032px #c49952, 45px 1879px #c49952, 686px 1469px #c49952, 1520px 475px
#c49952, 1122px 859px #c49952, 973px 1541px #c49952, 269px 477px #c49952,
1390px 716px #c49952, 1791px 783px #c49952, 824px 2000px #c49952, 1211px
1717px #c49952, 1008px 1587px #c49952, 1422px 204px #c49952, 234px 556px
#c49952, 506px 550px #c49952, 942px 1670px #c49952, 397px 853px #c49952,
599px 795px #c49952, 762px 1926px #c49952, 1202px 1424px #c49952, 135px

1316px #c49952, 1442px 1692px #c49952, 977px 652px #c49952, 564px 1648px
#c49952, 997px 1474px #c49952, 67px 1366px #c49952, 1860px 1451px
#c49952, 1105px 772px #c49952, 1886px 1396px #c49952, 1510px 658px
#c49952, 976px 1544px #c49952, 894px 543px #c49952, 1098px 1189px
#c49952, 690px 77px #c49952, 770px 733px #c49952, 557px 1403px #c49952,
1758px 1623px #c49952, 1341px 812px #c49952, 699px 967px #c49952, 277px
866px #c49952, 1526px 1828px #c49952, 8px 977px #c49952, 1707px 952px
#c49952, 12px 1900px #c49952, 72px 921px #c49952, 496px 1067px #c49952,
1288px 1749px #c49952, 273px 984px #c49952, 1197px 1991px #c49952, 242px
789px #c49952, 903px 1035px #c49952, 480px 1492px #c49952, 102px 1331px
#c49952, 738px 1343px #c49952, 560px 1475px #c49952, 367px 846px #c49952,
1420px 962px #c49952, 1976px 892px #c49952, 1911px 1763px #c49952, 1639px
1002px #c49952, 437px 1522px #c49952, 1906px 1025px #c49952, 730px 1364px
#c49952, 1127px 521px #c49952, 1401px 1792px #c49952, 1954px 1066px
#c49952, 232px 250px #c49952, 1685px 660px #c49952, 1011px 999px #c49952,
1970px 790px #c49952, 750px 499px #c49952, 1738px 660px #c49952, 1621px
1849px #c49952, 446px 52px #c49952, 1055px 1396px #c49952, 1165px 1497px
#c49952, 1740px 1425px #c49952, 1012px 1920px #c49952, 1258px 1560px
#c49952, 1020px 1152px #c49952, 362px 673px #c49952, 1065px 975px
#c49952, 582px 755px #c49952, 1271px 1479px #c49952, 719px 548px #c49952,
1602px 879px #c49952, 590px 499px #c49952, 721px 1412px #c49952, 1180px
113px #c49952, 1801px 1961px #c49952, 589px 941px #c49952, 883px 476px
#c49952, 214px 890px #c49952, 1028px 892px #c49952, 1107px 1832px
#c49952, 944px 361px #c49952, 480px 1453px #c49952, 1466px 683px #c49952,
981px 745px #c49952, 1968px 828px #c49952, 657px 1830px #c49952, 11px
1338px #c49952, 179px 730px #c49952, 1713px 197px #c49952, 51px 955px
#c49952, 1243px 319px #c49952, 1175px 624px #c49952, 446px 46px #c49952,
5px 1158px #c49952, 82px 1352px #c49952, 1877px 402px #c49952, 708px
1778px #c49952, 903px 1625px #c49952, 1824px 352px #c49952, 1229px 140px
#c49952, 1518px 24px #c49952, 1017px 512px #c49952, 515px 699px #c49952,
295px 265px #c49952, 69px 1773px #c49952, 1640px 1163px #c49952, 536px
342px #c49952, 970px 1766px #c49952, 560px 1416px #c49952, 577px 193px
#c49952, 469px 9px #c49952, 466px 276px #c49952, 711px 853px #c49952,
401px 685px #c49952, 85px 506px #c49952, 865px 558px #c49952, 631px 105px
#c49952, 887px 866px #c49952, 1704px 1001px #c49952, 1051px 1199px
#c49952, 275px 1909px #c49952, 1462px 829px #c49952, 375px 1057px
#c49952, 1531px 1501px #c49952, 205px 403px #c49952, 33px 1869px #c49952,
967px 1176px #c49952, 376px 863px #c49952, 1769px 1545px #c49952, 535px
51px #c49952, 1972px 1569px #c49952, 1773px 960px #c49952, 487px 620px
#c49952, 1660px 687px #c49952, 1632px 972px #c49952, 1362px 42px #c49952,
479px 1655px #c49952, 1531px 1808px #c49952, 1450px 1412px #c49952,
1549px 170px #c49952, 1904px 1305px #c49952, 1209px 48px #c49952, 1933px
820px #c49952, 1623px 595px #c49952, 48px 643px #c49952, 179px 1754px
#c49952, 589px 1032px #c49952, 1199px 356px #c49952, 1755px 1418px
#c49952, 780px 1174px #c49952, 1905px 758px #c49952, 1567px 713px

```
#c49952, 1372px 705px #c49952, 456px 654px #c49952, 759px 690px #c49952,
452px 673px #c49952, 993px 1610px #c49952, 1271px 188px #c49952, 343px
1750px #c49952, 1943px 1735px #c49952, 1717px 853px #c49952, 1247px 303px
#c49952, 1314px 1895px #c49952, 1203px 489px #c49952, 741px 469px
#c49952, 4px 246px #c49952, 1515px 115px #c49952, 606px 218px #c49952,
1966px 1471px #c49952, 177px 87px #c49952, 1575px 588px #c49952, 1136px
1386px #c49952, 70px 1868px #c49952, 1053px 18px #c49952, 1124px 721px
#c49952, 1748px 1181px #c49952, 191px 1387px #c49952, 1931px 840px
#c49952, 1088px 1603px #c49952, 634px 1255px #c49952, 814px 1434px
#c49952, 585px 64px #c49952, 1074px 1618px #c49952, 1692px 761px #c49952,
651px 643px #c49952, 193px 335px #c49952, 1103px 1447px #c49952, 491px
1142px #c49952, 521px 408px #c49952, 536px 340px #c49952, 411px 1091px
#c49952, 1646px 193px #c49952, 1595px 1285px #c49952, 870px 1349px
#c49952, 1085px 1013px #c49952, 204px 1864px #c49952, 1359px 299px
#c49952, 807px 964px #c49952, 219px 509px #c49952, 36px 1227px #c49952,
702px 1873px #c49952, 1471px 934px #c49952, 1763px 792px #c49952, 973px
1957px #c49952, 987px 68px #c49952, 593px 1282px #c49952, 1900px 607px
#c49952, 407px 1659px #c49952, 587px 17px #c49952, 632px 158px #c49952;
}
```

```
@-webkit-keyframes animStar {
  from {
    -webkit-transform: translateY(0px);
  }
  to {
    -webkit-transform: translateY(-2000px);
  }
}
```

```
@-moz-keyframes animStar {
  from {
    -moz-transform: translateY(0px);
  }
  to {
    -moz-transform: translateY(-2000px);
  }
}
```

```
@-ms-keyframes animStar {
  from {
    -ms-transform: translateY(0px);
  }
  to {
    -ms-transform: translateY(-2000px);
  }
}
```



```

}

@keyframes animStar {
  from {
    transform: translateY(0px);
  }
  to {
    transform: translateY(-2000px);
  }
}

// ./client/src/components/Footer.jsx

import React from "react";
import styled from "styled-components";
import { FaInstagram } from "react-icons/fa";

import "./stars.scss"

const Footer = () => {
  return (
    <Wrapper>

      <footer>
        <div className="container grid grid-two-column">
          <div className="footer-about">
            <h3>SuperCoffee</h3>
            <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. </p>
          </div>
          <div className="footer-social">
            <h3>Слідкуй за нами</h3>
            <div className="footer-social--icons">
              <div>
                <a
                  href="https://www.instagram.com/"
                  target="_blank" rel="noreferrer">
                    <FaInstagram className="icons" />
                </a>
              </div>
            </div>
          </div>
        </div>

        <div className="footer-bottom--section">
          <hr />
        </div>
      </footer>
    </Wrapper>
  )
}

```

```

    <div className="container grid grid-two-column ">
      <p>
        @{{new Date().getFullYear}} SuperCoffee. Всі права захищені
      </p>
      <div>
        <p>PRIVACY POLICY</p>
        <p>TERMS & CONDITIONS</p>
      </div>
    </div>
  </div>
</footer>
</Wrapper>
// </>
);
};

```

```

const Wrapper = styled.section`
  footer {
    bottom: 0;
    padding: 4rem 0 4rem 0;
    background-color: ${({ theme }) => theme.colors.footer_bg};
    h3 {
      color: ${({ theme }) => theme.colors.hr};
      margin-bottom: 2.4rem;
    }
    p {
      color: ${({ theme }) => theme.colors.white};
    }

    .footer-social--icons {
      display: flex;
      gap: 2rem;

      div {
        padding: 1rem;
        border-radius: 50%;
        border: 2px solid ${({ theme }) => theme.colors.white};

        .icons {
          color: ${({ theme }) => theme.colors.white};
          font-size: 2.4rem;
          position: relative;
          cursor: pointer;
        }
      }
    }
  }

```

```

    }
  }

.footer-bottom--section {
  padding-top: 9rem;

  hr {
    margin-bottom: 2rem;
    color: ${({ theme }) => theme.colors.hr};
    height: 0.1px;
  }
}

@media (max-width: ${({ theme }) => theme.media.mobile}) {

  footer {
    padding: 9rem 0 9rem 0;
  }

  .footer-bottom--section {
    padding-top: 4.8rem;
  }
}
`;

export default Footer;

// ./client/src/components/Sort.jsx

import React from "react";
import styled from "styled-components";
import { BsFillGridFill, BsList } from "react-icons/bs";
import { useFilterContext } from "../context/filterContext"

const Sort = () => {
  const { filterProducts, gridView, setGridView, setListView, sorting } =
    useFilterContext();
  return (
    <Wrapper className="sort-section">
      { /* 1st column */ }
      <div className="sorting-list--grid">
        <button
          className={gridView ? "active sort-btn" : "sort-btn"}
          onClick={setGridView}>

```

```

    <BsFillGridFill className="icon" />
  </button>

  <button
    className={!gridView ? "active sort-btn" : " sort-btn"}
    onClick={setListView}>
    <BsList className="icon" />
  </button>
</div>
{ /* 2nd column */}
<div className="product-data">
  <p>`${filterProducts.length} Сортів Доступно`</p>
</div>

{ /* 3rd column */}
<div className="sort-selection">
  <form action="#">
    <label htmlFor="sort"></label>
    <select
      name="sort"
      id="sort"
      className="sort-selection--style"
      onChange={sorting}>
      <option value="lowest">Ціна(найнижча)</option>
      <option value="#" disabled></option>
      <option value="highest">Ціна(найвижча)</option>
      <option value="#" disabled></option>
      <option value="a-z">Ціна(а-я)</option>
      <option value="#" disabled></option>
      <option value="z-a">Ціна(я-а)</option>
    </select>
  </form>
</div>
</Wrapper>
);
};

const Wrapper = styled.section`
  display: flex;
  justify-content: space-between;
  margin-top: 5rem;

  .sorting-list--grid {
    display: flex;
    gap: 2rem;

```

```

.sort-btn {
  padding: 0.8rem 1rem;
  border: none;
  display: flex;
  justify-content: center;
  align-items: center;
  cursor: pointer;
  background-color: ${({ theme }) => theme.colors.bg};
  color: #fff;
}

.icon {
  font-size: 1.6rem;
}
.active {
  background-color: ${({ theme }) => theme.colors.black};
  color: #fff;
}
}

.sort-selection .sort-selection--style {
  padding: 0.5rem;
  cursor: pointer;

  .sort-select--option {
    padding: 0.5rem 0;
    cursor: pointer;
    height: 2rem;
    padding: 10px;
  }
}
`;

export default Sort;

// ./client/src/components/RecProducts.jsx

import { useProductContext } from "../context/productContext";
import styled from "styled-components";
import Product from "../Product";
import { useAuthState } from "../config/firebase";
import { postRecommendations } from "../api/actions";
import { useEffect, useState } from "react";

```

```

const RecProducts = () => {
  const { isLoading, products } = useProductContext();
  const { uid, isAuthenticated } = useAuthState();
  const [recs, setRecs] = useState("");

  useEffect(() => {

    async function getRecs(uid, isAuthenticated, products) {
      try {

        if(isAuthenticated){
          const recs = await postRecommendations({ user_id: uid, products:
products});

          if (Array.isArray(recs)){
            setRecs(recs)
          }
        } else {
          const recs = await postRecommendations({ user_id: "", products:
products});

          if (Array.isArray(recs)){
            setRecs(recs)
          }
        }
      }
      ;
    } catch (error) {
      console.error('Error performing action and navigating:', error);
    }
  }
  };

  if(!isLoading)getRecs(uid, isAuthenticated, products)

  // eslint-disable-next-line react-hooks/exhaustive-deps
  }, [isLoading])

  if (isLoading || !(Array.isArray(recs) && recs.length)) {
    return <div> .....Loading </div>;
  }
}

```

```
let selectedProducts = recs;
```

```
return (  
  <WrapperRecs className="section">  
    <div className="container">  
      <div className="intro-data">Спробуй!</div>  
      <div className="common-heading">Рекомендуемо</div>  
      <div className="grid grid-three-column">  
        {selectedProducts.map((curElem) => {  
          return <Product key={curElem.id} {...curElem} />;  
        })}  
      </div>  
    </WrapperRecs>  
  );  
};
```

```
const WrapperRecs = styled.section`  
  margin: 0 0 4rem 0;  
  padding: 4rem 0;  
  background-color: ${({ theme }) => theme.colors.bg};  
  border: 1px solid black;
```

```
.container {  
  max-width: 120rem;  
}
```

```
figure {  
  width: auto;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  position: relative;  
  overflow: hidden;  
  transition: all 0.5s linear;  
  border-radius: 1rem 1rem 0 0;  
  &::after {  
    content: "";  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 0%;  
    height: 100%;
```

```

background-color: rgba(0, 0, 0, 0.5);
transition: all 0.2s linear;
cursor: pointer;
}
&:hover::after {
width: 100%;

}
&:hover img {
transform: scale(1.2);
}
img {
max-width: 90%;
margin-top: 1.5rem;
height: 20rem;
transition: all 0.2s linear;
}

.caption {
position: absolute;
top: 15%;
right: 10%;
text-transform: uppercase;
background-color: ${({ theme }) => theme.colors.footer_bg};
color: ${({ theme }) => theme.colors.white};
border: 1px solid;
border-color: ${({ theme }) => theme.colors.helper};
padding: 0.8rem 2rem;
font-size: 1.2rem;
border-radius: 2rem;
}
}

.card {
background-color: ${({ theme }) => theme.colors.bgS};;
border-radius: 1rem;

.card-data {
padding: 0 2rem;
border-radius: 0 0 1rem 1rem;
}

.card-data-flex {
margin: 2rem 0;
display: flex;

```



```

    justify-content: space-between;
    align-items: center;
  }

  h3 {
    color: ${({ theme }) => theme.colors.white};
    text-transform: capitalize;
  }

  .card-data--price {
    color: ${({ theme }) => theme.colors.helper};
  }

  .btn {
    margin: 2rem auto;
    background-color: rgb(0 0 0 / 0%);
    border: 0.1rem solid rgb(98 84 243);
    display: flex;
    justify-content: center;
    align-items: center;

    &:hover {
      background-color: rgb(98 84 243);
    }

    &:hover a {
      color: #fff;
    }
    a {
      color: rgb(98 84 243);
      font-size: 1.4rem;
    }
  }
}
`;

```

```
export default RecProducts;
```

```
// ./client/src/components/GridView.jsx
```

```

import React from "react";
import styled from "styled-components";
import Product from "../Product";
import RecProducts from "../RecProducts";

```

```
const GridView = ({ products }) => {
  return (
    <Wrapper className="section">
      <RecProducts/>
      <div className="container grid grid-three-column">
        {products.map((curElem) => {
          return <Product key={curElem.id} {...curElem} />;
        })}
      </div>
    </Wrapper>
  );
};
```

```
const Wrapper = styled.section`
  padding: 9rem 0;
```

```
.container {
  max-width: 120rem;
}
```

```
.grid {
  gap: 3.2rem;
}
```

```
figure {
  width: auto;
  display: flex;
  justify-content: center;
  align-items: center;
  position: relative;
  overflow: hidden;
  transition: all 0.5s linear;
  border-radius: 1rem 1rem 0 0;
  &::after {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 0%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
    transition: all 0.2s linear;
    cursor: pointer;
  }
}
```

```

&:hover::after {
  width: 100%;
}
&:hover img {
  transform: scale(1.2);
}
img {
  max-width: 90%;
  margin-top: 1.5rem;
  height: 20rem;
  transition: all 0.2s linear;
}
}

.card {
  background-color: ${({ theme }) => theme.colors.bg};
  border-radius: 1rem;

  .card-data {
    padding: 0 1rem;
  }

  .card-data-flex {
    margin: 2rem 0;
    display: flex;
    justify-content: space-between;
    align-items: center;
  }

  .card-data--price {
    color: ${({ theme }) => theme.colors.helper};
  }

  h3 {
    color: ${({ theme }) => theme.colors.text};
    text-transform: capitalize;
  }

  .caption {
    position: absolute;
    top: 15%;
    right: 10%;
    text-transform: uppercase;
    background-color: ${({ theme }) => theme.colors.footer_bg};
    color: ${({ theme }) => theme.colors.white};
  }

```

```

border: 1px solid;
border-color: ${({ theme }) => theme.colors.helper};
padding: 0.8rem 2rem;
font-size: 1.2rem;
border-radius: 2rem;
}

.btn {
margin: 2rem auto;
background-color: rgb(0 0 0 / 0%);
border: 0.1rem solid rgb(98 84 243);
display: flex;
justify-content: center;
align-items: center;

&:hover {
background-color: rgb(98 84 243);
}

&:hover a {
color: #fff;
}
a {
color: rgb(98 84 243);
font-size: 1.4rem;
}
}
}
`
;

```

```
export default GridView;
```

```
// ./client/src/components/ListView.jsx
```

```

import { NavLink } from "react-router-dom";
import styled from "styled-components";
import { Button } from "../styles/Button";
import { renderDefaultCoffee } from "../functions/images";
import RecProducts from "../RecProducts";

```

```
const ListView = ({ products }) => {
```

```

return (
  <Wrapper className="section">
    <RecProducts/>

```

```

<div className="container grid">
  {products.map((curElem) => {
    const { id, name, image, price, description, categories } = curElem;
    let truncatedDescription = description.slice(0, 90) + "...";
    let defCoffee = "";
    let coffee = ""
    if(image)coffee = image.url;
    defCoffee = renderDefaultCoffee(categories);
    return (
      <div key= {id} className="card grid grid-two-column">
        <figure>
          <img src={coffee || defCoffee} alt={name} />
        </figure>

        <div className="card-data">
          <h3>{name}</h3>
          <p>
            {price.formatted_with_symbol}
          </p>
          <p dangerouslySetInnerHTML={{ __html: truncatedDescription
        }}></p>

          <NavLink to={`/${singleproduct}/${id}`} className="btn-main">
            <Button className="btn">Показати більше</Button>
          </NavLink>
        </div>
      </div>
    );
  })}
</div>
</Wrapper>
);
};

```

```

const Wrapper = styled.section`
padding: 9rem 0;

```

```

.container {
  max-width: 120rem;

}

```

```

.grid {
  gap: 3.2rem;

```

```

}

figure {
  width: auto;
  display: flex;
  justify-content: center;
  align-items: center;
  position: relative;
  overflow: hidden;
  transition: all 0.5s linear;
  &::after {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 0%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
    transition: all 0.2s linear;
    cursor: pointer;
  }
  &:hover::after {
    width: 100%;
  }
  &:hover img {
    transform: scale(1.2);
  }
  img {
    max-width: 90%;
    margin-top: 1.5rem;
    height: 20rem;
    transition: all 0.2s linear;
  }
}

.card {
  border: 0.1rem solid rgb(170 170 170 / 40%);
  background-color: ${({ theme }) => theme.colors.bg};

  .card-data {
    padding: 0 2rem;
  }

  h3 {
    margin: 2rem 0;
  }

```

```

    font-weight: 300;
    font-size: 2.4rem;
    text-transform: capitalize;
    color: rgba(196,153,82, 1);
  }

  .btn {
    margin: 2rem 0;
    background-color: rgb(0 0 0 / 0%);
    border: 0.1rem solid rgba(196,153,82, 1);
    display: flex;
    justify-content: center;
    align-items: center;
    color: rgba(196,153,82, 1);

    &:hover {
      background-color: rgba(196,153,82, 1);
    }

    &:hover a {
      color: #fff;
    }
    a {
      color: rgba(196,153,82, 1);
      font-size: 1.4rem;
    }
  }

  .btn-main .btn:hover {
    color: #fff;
  }
}
`;

export default ListView;

// ./client/src/components/AddToCart.jsx

import { useState } from "react";
import styled from "styled-components";
import CartAmountToggle from "./CartAmountToggle";
import { NavLink } from "react-router-dom";
import { Button } from "../styles/Button";
import { useCartContext } from "../context/cartContext";

```

```

const AddToCart = ({ updateQuantity, product }) => {
  const {addToCart} = useCartContext();
  const {id} = product;

  const [amount, setAmount] = useState(1);

  const setDecrease = () => {
    amount > 1 ? setAmount(amount - 1) : setAmount(1);
    amount > 1 ? updateQuantity(amount - 1) : setAmount(1);
  };

  const setIncrease = () => {
    amount < 20 ? setAmount(amount + 1) : setAmount(20);
    amount < 20 ? updateQuantity(amount + 1) : setAmount(20);
  };

  return (
    <Wrapper>
      <CartAmountToggle
        amount={amount}
        setDecrease={setDecrease}
        setIncrease={setIncrease}
      />

      <NavLink to="/cart" onClick={() => addToCart(id, amount, product)}>
        <Button className="btn">Додати до кошика</Button>
      </NavLink>
    </Wrapper>
  );
};

const Wrapper = styled.section`
  color: white;

  .amount-toggle {
    margin-top: 3rem;
    margin-bottom: 1rem;
    display: flex;
    justify-content: space-around;
    align-items: center;
    font-size: 1.4rem;

    button {
      border: none;
      background-color: ${({ theme }) => theme.colors.bgS};
    }
  }

```



```

    cursor: pointer;
  }

  .amount-style {
    font-size: 2.4rem;
    color: rgb(140, 154, 249);
  }
}
`;
export default AddToCart;

// ./client/src/components/PageNavigation.jsx

import React from "react";
import { NavLink } from "react-router-dom";
import styled from "styled-components";

const PageNavigation = ({ title }) => {
  return (
    <Wrapper>
      <NavLink to="/">Home</NavLink>/ {title}
    </Wrapper>
  );
};

const Wrapper = styled.section`
  height: 10rem;
  background-color: ${({ theme }) => theme.colors.bg};
  display: flex;
  justify-content: flex-start;
  align-items: center;
  font-size: 3.2rem;
  padding-left: 1.2rem;
  color: white;

  a {
    font-size: 3.2rem;
    color: ${({ theme }) => theme.colors.helper};
  }
`;

export default PageNavigation;

```

```
// ./client/src/components/Product.jsx

import React from "react";
import { useNavigate } from "react-router-dom";
import { renderDefaultCoffee } from "../functions/images";
import { postAction } from "../api/actions";
import { useAuthState } from "../config/firebase";

const Product = (curElem) => {
  const { id, name, image, price, categories } = curElem;
  const navigate = useNavigate();
  let defCoffee = "";
  let coffee = ""
  if(image)coffee = image.url;
  defCoffee = renderDefaultCoffee(categories);

  const { uid, isAuthenticated } = useAuthState();

  const handleClick = async () => {
    try {

      if(isAuthenticated){
        await postAction({ user_id: uid, product_id: id, action_type: 'click' });
      }

      navigate(`/singleproduct/${id}`);
    } catch (error) {
      console.error('Error performing action and navigating:', error);
    }
  };

  return (

    <div className="card" onClick={handleClick}>
      <figure>
        <img src={coffee || defCoffee} alt={name} />
        <figcaption className="caption">{categories[0].name}</figcaption>
      </figure>

      <div className="card-data">
        <div className="card-data-flex">
          <h3>{name}</h3>

```

```

        <p className="card-data--price">{price.formatted_with_symbol}</p>
      </div>
    </div>
  </div>

  );
};

export default Product;

// ./client/src/components/ProductList.jsx

import React from "react";
import { useFilterContext } from "../context/filterContext";
import GridView from "./GridView";
import ListView from "./ListView";

const ProductList = () => {
  const { filterProducts, gridView } = useFilterContext();

  if(gridView === true){
    return <GridView products={filterProducts} />
  }
  if(gridView === false){
    return <ListView products={filterProducts} />
  }
}

export default ProductList;

// ./client/src/components/Nav.jsx

import React, { useState } from "react";
import { NavLink, useNavigate } from "react-router-dom";
import styled from "styled-components";
import { FaCoffee } from "react-icons/fa";
import { CgMenu, CgClose } from "react-icons/cg";
import { useCartContext } from "../context/cartContext";
import { signOut } from "firebase/auth";
import { auth, useAuthState } from "../config/firebase";
import { Button } from "../styles/Button";

const Nav = () => {
  const [menuIcon, setMenuIcon] = useState();

```

```

const { totalItem } = useCartContext();

const navigate = useNavigate();

const { isAuthenticated } = useAuthState();

const logout = async () => {
  await signOut(auth);
};

return (
  <Navbar>
    <div className={menuIcon ? "navbar active" : "navbar"}>
      <ul className="navbar-lists">
        <li>
          <NavLink
            to="/"
            className="navbar-link "
            onClick={() => setMenuIcon(false)}>
            Головна
          </NavLink>
        </li>
        <li>
          <NavLink
            to="/about"
            className="navbar-link "
            onClick={() => setMenuIcon(false)}>
            Про нас
          </NavLink>
        </li>
        <li>
          <NavLink
            to="/products"
            className="navbar-link "
            onClick={() => setMenuIcon(false)}>
            Меню
          </NavLink>
        </li>
        <li>
          <NavLink
            to="/contact"
            className="navbar-link "
            onClick={() => setMenuIcon(false)}>
            Контакти
          </NavLink>
        </li>
      </ul>
    </div>
  </Navbar>
)

```

```

</li>

<li>
  {isAuthenticated ? (
    <Button onClick={logout}>Logout</Button>
  ) : (
    <Button onClick={() => navigate("/login")}>Login</Button>
  )}
</li>

<li>
  <NavLink to="/cart" className="navbar-link cart-trolley--link">
    <FaCoffee className="cart-trolley" />
    <span className="cart-total--item">{totalItem}</span>
  </NavLink>
</li>
</ul>

<div className="mobile-navbar-btn">
  <CgMenu
    name="menu-outline"
    className="mobile-nav-icon"
    onClick={() => setMenuIcon(true)}
  />
  <CgClose
    name="close-outline"
    className="mobile-nav-icon close-outline"
    onClick={() => setMenuIcon(false)}
  />
</div>
</div>
</Navbar>
);
};

const Navbar = styled.nav`
.navbar-lists {
  display: flex;
  gap: 4.8rem;
  align-items: center;

  .navbar-link {
    &:link,
    &:visited {

```

```

    display: inline-block;
    text-decoration: none;
    font-size: 1.8rem;
    font-weight: 500;
    text-transform: uppercase;
    color: #fff;
    transition: color 0.3s linear;
  }

  &:hover,
  &:active {
    color: ${({ theme }) => theme.colors.helper};
  }
}

.mobile-navbar-btn {
  display: none;
  background-color: transparent;
  cursor: pointer;
  border: none;
}

.mobile-nav-icon[name="close-outline"] {
  display: none;
}

.close-outline {
  display: none;
}

.cart-trolley--link {
  position: relative;

  .cart-trolley {
    position: relative;
    font-size: 3.2rem;
  }

  .cart-total--item {
    width: 2.4rem;
    height: 2.4rem;
    position: absolute;
    background-color: #000;
    color: #000;
  }
}

```

```

border-radius: 50%;
display: grid;
place-items: center;
top: -20%;
left: 70%;
background-color: ${({ theme }) => theme.colors.helper};
}
}

.user-login--name {
  text-transform: capitalize;
}

.user-logout,
.user-login {
  font-size: 1.4rem;
  padding: 0.8rem 1.4rem;
}

@media (max-width: ${({ theme }) => theme.media.mobile}) {
  .mobile-navbar-btn {
    display: inline-block;
    z-index: 9999;
    border: #fff;

    .mobile-nav-icon {
      font-size: 4.2rem;
      color: #fff;
    }
  }

  .active .mobile-nav-icon {
    display: none;
    font-size: 4.2rem;
    position: absolute;
    top: 30%;
    right: 10%;
    color: #000;
    z-index: 9999;
  }

  .active .close-outline {
    display: inline-block;
  }
}

```

```

.navbar-lists {
  width: 100vw;
  height: 100vh;
  position: absolute;
  top: 0;
  left: 0;
  background-color: rgba(255,255,255, 0.8);

  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;

  visibility: hidden;
  opacity: 0;
  transform: translateX(100%);
  /* transform-origin: top; */
  transition: all 3s linear;
}

.active .navbar-lists {
  visibility: visible;
  opacity: 1;
  transform: translateX(0);
  z-index: 999;
  transform-origin: right;
  transition: all 3s linear;

  .navbar-link {
    font-size: 4.2rem;
    color: #000;
  }
}

.cart-trolley--link {
  position: relative;

  .cart-trolley {
    position: relative;
    font-size: 5.2rem;
  }

  .cart-total--item {
    width: 4.2rem;
    height: 4.2rem;
    font-size: 2rem;
  }
}

```



```

    }
  }

  .user-logout,
  .user-login {
    font-size: 2.2rem;
    padding: 0.8rem 1.4rem;
  }
}
`;

export default Nav;

// ./client/src/components/FilterSection.jsx

import React from "react";
import styled from "styled-components";
import { useFilterContext } from "../context/filterContext";
import { getCategory } from "../functions/filter";
import { Button } from "../styles/Button";

const FilterSection = () => {
  const {
    filters: {text, category, price, maxPrice, minPrice},
    updateFilterValue,
    allProducts,
    clearFilters
  } = useFilterContext();

  const categoryData = getCategory(allProducts);

  return (
    <Wrapper>
      <div className="filter-search">
        <form onSubmit={(e) => e.preventDefault()}>
          <input
            type="text"
            name="text"
            value={text}
            placeholder="ПОИСК"
            onChange={updateFilterValue}
          />
        </form>
      </div>
    </Wrapper>
  );
};

```

```

<div className="filter-category">
  <h3>Категорії</h3>
  <div>
    {categoryData.map((curElem, index) => {
      return (
        <button
          key={index}
          type="button"
          name="category"
          value={curElem}
          className={curElem === category ? "active" : ""}
          onClick={updateFilterValue}>
            {curElem}
          </button>
        );
      )}
    )}
  </div>
</div>

<div className="filter_price">
  <h3>Ціна</h3>
  <p>
    {price}
  </p>
  <input
    type="range"
    name="price"
    min={minPrice}
    max={Math.ceil(maxPrice)}
    value={price}
    onChange={updateFilterValue}
  />
</div>

<div className="filter-clear">
  <Button className="btn" onClick={clearFilters}>
    Очистити Фільтри
  </Button>
</div>

</Wrapper>
)
}

const Wrapper = styled.section`

```

```

padding: 5rem 0;
display: flex;
flex-direction: column;
gap: 3rem;

h3 {
  padding: 2rem 0;
  background: linear-gradient(to right, #fff, yellow);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  font-size: 2rem;
  font-weight: bold;
  text-shadow: 2px 2px 4px rgba(196,153,82, 0.5);
}

.filter-search {
  input {
    padding: 0.6rem 1rem;
    width: 80%;
  }
}

.filter-category {
  div {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    gap: 1.4rem;

    button {
      border: none;
      background-color: ${({ theme }) => theme.colors.bgS};
      text-transform: capitalize;
      cursor: pointer;

      &:hover {
        color: ${({ theme }) => theme.colors.helper};
      }
    }

    .active {
      border-bottom: 1px solid #000;
      color: ${({ theme }) => theme.colors.helper};
    }
  }
}

```

```
}  
}
```

```
.btnStyle {  
  width: 2rem;  
  height: 2rem;  
  background-color: #000;  
  border-radius: 50%;  
  margin-left: 1rem;  
  border: none;  
  outline: none;  
  opacity: 0.5;  
  cursor: pointer;
```

```
  &:hover {  
    opacity: 1;  
  }  
}
```

```
.active {  
  opacity: 1;  
}
```

```
.checkStyle {  
  font-size: 1rem;  
  color: #fff;  
}
```

```
.filter_price {  
  input[type=range] {  
    height: 0;  
    margin: 0.5rem 0 1rem 0;  
    padding: 0;  
    box-shadow: none;  
    background: #969632;  
    cursor: pointer;  
    -webkit-appearance: none;  
    margin: 0.5rem 0 1rem 0;  
    width: 90%;  
  }  
  input[type=range]:focus {  
    outline: none;  
  }  
  input[type=range]::-webkit-slider-runnable-track {  
    width: 100%;
```

```

height: 9px;
cursor: pointer;
animate: 0.2s;
box-shadow: 1px 1px 1px #002200;
background: #DED51B;
border-radius: 16px;
border: 1px solid #D5CF26;
}
input[type=range]::-webkit-slider-thumb {
box-shadow: 1px 1px 1px #AAA520;
border: 2px solid #E5E532;
height: 15px;
width: 15px;
margin-top: -5.5px;
border-radius: 23px;
background: #969632;
cursor: pointer;
-webkit-appearance: none;
}
input[type=range]:focus::-webkit-slider-runnable-track {
background: #DED51B;
}
input[type=range]::-moz-range-track {
width: 100%;
height: 9px;
cursor: pointer;
animate: 0.2s;
box-shadow: 1px 1px 1px #002200;
background: #DED51B;
border-radius: 16px;
border: 1px solid #D5CF26;
}
input[type=range]::-moz-range-thumb {
box-shadow: 3px 3px 3px #AAA520;
border: 2px solid #E5E532;
height: 15px;
width: 15px;
border-radius: 23px;
background: #969632;
cursor: pointer;
}
input[type=range]::-ms-track {
width: 100%;
height: 9px;
cursor: pointer;

```

```

    animate: 0.2s;
    background: transparent;
    border-color: transparent;
    color: transparent;
  }
  input[type=range]::-ms-fill-lower {
    background: #DED51B;
    border: 1px solid #D5CF26;
    border-radius: 32px;
    box-shadow: 1px 1px 1px #002200;
  }
  input[type=range]::-ms-fill-upper {
    background: #DED51B;
    border: 1px solid #D5CF26;
    border-radius: 32px;
    box-shadow: 1px 1px 1px #002200;
  }
  input[type=range]::-ms-thumb {
    margin-top: 1px;
    box-shadow: 3px 3px 3px #AAA520;
    border: 2px solid #E5E532;
    height: 15px;
    width: 15px;
    border-radius: 23px;
    background: #969632;
    cursor: pointer;
  }
  input[type=range]:focus::-ms-fill-lower {
    background: #DED51B;
  }
  input[type=range]:focus::-ms-fill-upper {
    background: #DED51B;
  }
}

.filter-shipping {
  display: flex;
  align-items: center;
  gap: 1rem;
}

.filter-clear .btn {
  background-color: #ec7063;
  font-size: 1.3rem;
  color: #000;

```

```

    }
  `;

export default FilterSection;

// ./client/src/components/CartAmountToggle.jsx

import React from "react";
import { FaMinus, FaPlus } from "react-icons/fa";

const CartAmountToggle = ({ amount, setDecrease, setIncrease }) => {
  return (

    <div className="cart-button">
      <div className="amount-toggle">
        <button onClick={() => setDecrease()}>
          <FaMinus />
        </button>
        <div className="amount-style">{amount}</div>
        <button onClick={() => setIncrease()}>
          <FaPlus />
        </button>
      </div>
    </div>

  );
};

export default CartAmountToggle;

// ./client/src/components/CartItem.jsx

import React from "react";
import CartAmountToggle from "../CartAmountToggle";
import { FaTrash } from "react-icons/fa";
import { useCartContext } from "../context/cartContext";

const CartItem = ({ id, name, image, price, amount }) => {
  const { removeItem, setDecrease, setIncrease } = useCartContext();

  return (
    <div className="cart-heading grid grid-five-column">
      <div className="cart-image--name">
        <div>

```

```

        <figure>
          <img src={image} alt={id} />
        </figure>
      </div>
      <div>
        <p>{name} {amount} шт.</p>
      </div>
    </div>
    { /* price */ }
    <div className="cart-hide">
      <p>
        {price}
      </p>
    </div>

    { /* Quantity */ }
    <CartAmountToggle
      amount={amount}
      setDecrease={() => setDecrease(id)}
      setIncrease={() => setIncrease(id)}
    />

    { /* //Subtotal */ }
    <div className="cart-hide">
      <p>
        {price * amount}
      </p>
    </div>

    <div>
      <FaTrash className="remove_icon" onClick={() => removeItem(id)} />
    </div>
  </div>
);
};

```

```
export default CartItem;
```

```
// ./client/src/components/Services.jsx
```

```

import React from "react";
import styled from "styled-components";
import { TbTruckDelivery } from "react-icons/tb";

```



```

import { GiUfo } from "react-icons/gi";
import { SiCoffeescript } from "react-icons/si";
import { RiSecurePaymentLine } from "react-icons/ri";

const Services = () => {

  return (
    <Wrapper>
      <div className="container">
        <div className="grid grid-three-column">
          <div className="services-1">
            <div>
              <TbTruckDelivery className="icon" />
              <h3>Доставка з космічною швидкістю</h3>
            </div>
          </div>

          <div className="services-2">
            <div className="services-colum-2">
              <div>
                <GiUfo className="icon" />
                <h3>Тематичні вечори</h3>
              </div>
            </div>
            <div className="services-colum-2">
              <div>
                <SiCoffeescript className="icon" />
                <h3>Різноманітні напої</h3>
              </div>
            </div>
          </div>

          <div className="services-3">
            <div>
              <RiSecurePaymentLine className="icon" />
              <h3>Надбезпечна платіжна система</h3>
            </div>
          </div>
        </div>
      </div>
    </Wrapper>
  )
}

const Wrapper = styled.section`

```

```
padding: 7rem 0;
```

```
.grid {  
  gap: 4.8rem;  
}
```

```
.services-1,  
.services-2,  
.services-3 {  
  width: auto;  
  height: 30rem;  
  display: flex;  
  flex-direction: column;  
  color: #fff;  
  justify-content: center;  
  align-content: center;  
  ${" /* background: ${{({ theme }) => theme.colors.bg}}; */}  
  background: rgba(255,255,255,0.1);  
  text-align: center;  
  border-radius: 2rem;  
  box-shadow: rgba(0, 0, 0, 0.05) 0px 1px 2px 0px;  
}
```

```
.services-2 {  
  gap: 4rem;  
  background-color: transparent;  
  box-shadow: none;
```

```
.services-column-2 {  
  ${" /* background: ${{({ theme }) => theme.colors.bg}}; */}  
  background: rgba(255,255,255,0.1);  
  display: flex;  
  flex-direction: row;  
  flex: 1;  
  justify-content: center;  
  align-items: center;  
  border-radius: 2rem;  
  box-shadow: rgba(0, 0, 0, 0.05) 0px 1px 2px 0px;
```

```
div {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: center;  
  gap: 1rem;
```

```

    }
  }
}

h3 {
margin-top: 1.4rem;
font-size: 2rem;
}

.icon {
/* font-size: rem; */
width: 8rem;
height: 8rem;
padding: 2rem;
border-radius: 50%;
background-color:${({ theme }) => theme.colors.bg};
color: #c49952;
}
`;

```

```
export default Services;
```

```
// ./client/src/lib/commerce.js
```

```
import Commerce from "@chec/commerce.js";
```

```
export const commerce = new
Commerce(process.env.REACT_APP_CHECK_PUBLIC_KEY, true);
```

```
// ./client/src/api/actions.js
```

```
const API_URL = 'http://127.0.0.1:8000';
```

```
export const postAction = async (data) => {
  try {
    const response = await fetch(`${API_URL}/action`, {
      method: 'POST',
      headers: {
        'Content-Type': 'text/plain',
      },
      body: JSON.stringify(data),
    });

    if (!response.ok) {

```

```

        throw new Error(`HTTP error! Status: ${response.status}`);
    }

    return response.json();
} catch (error) {
    console.error('Error:', error);
    throw error;
}
};

export const postRecommendations = async (data) => {
    try {
        const response = await fetch(`${API_URL}/recommendation`, {
            method: 'POST',
            headers: {
                'Content-Type': 'text/plain',
            },
            body: JSON.stringify(data),
        });

        if (!response.ok) {
            throw new Error(`HTTP error! Status: ${response.status}`);
        }

        return response.json();
    } catch (error) {
        console.error('Error:', error);
        throw error;
    }
};

// ./client/src/functions/filter.js

export const getCategory = (data) => {
    const attributeName = "categories";
    let newVal = data.map((curElem) => {
        return curElem[attributeName];
    });
    newVal = newVal.map((curElem) => {
        return curElem[0].name;
    })
    return ["bce", ...new Set(newVal)];
};

```

```
// ./client/src/functions/images.js

export const renderDefaultCoffee = (categories) => {
  switch (categories[0].slug) {
    case "coffee":
      return "../images/coffee1.jpeg";
    case "drink":
      return "../images/drink.png";
    case "espresso":
      return "../images/espresso.jpeg";
    case "bakery":
      return "../images/bakery.jpeg";
    default:
      return "../images/coffee1.jpeg";
  }
};
```

```
// ./client/src/pages/ErrorPage.jsx
```

```
import { NavLink } from "react-router-dom";
import styled from "styled-components";
import { Button } from "../styles/Button"
```

```
const ErrorPage = () => {
  return (
    <Wrapper>
      <div className="container">
        <div>
          <h2>404</h2>
          <h3>Загублені в космосі</h3>
          <p>
            Ви поринули в невідоме. Небесний шлях, якого ви шукаєте,
            затьмарений.
            Але не бійтеся, адже космос містить багато чудес. Натисніть кнопку
            нижче
            щоб повернутися на головну сторінку та ще раз орієнтуватися по
            зірках.
          </p>

          <NavLink to="/">
            <Button>Повернутися на головну</Button>
          </NavLink>
        </div>
      </div>
    </Wrapper>
  );
};
```

```

        </div>
      </Wrapper>
    );
  };

const Wrapper = styled.section`
  .container {
    padding: 9rem 0;
    text-align: center;

    h2 {
      font-size: 10rem;
    }

    h3 {
      font-size: 4.2rem;
      color: white;
    }

    p {
      margin: 2rem 10rem;
    }
  }
`;

export default ErrorPage;

// ./client/src/pages/About.jsx

import React from "react";
import HeroSection from "../components/HeroSection";

const About = () => {
  const data = {
    name: "SuperCoffee",
  }
  return (
    <
      <HeroSection myData={data}/>
    </>
  )
}

```

```

export default About;

// ./client/src/pages/Products.jsx

import React from "react";
import styled from "styled-components";
import FilterSection from "../components/FilterSection";
import Sort from "../components/Sort";
import ProductList from "../components/ProductList";

const Products = () => {

  return <Wrapper>
    <div className="container grid grid-filter-column">
      <div>
        <FilterSection />
      </div>

      <section className="product-view--sort">
        <div className="sort-filter">
          <Sort/>
        </div>
        <div className="main-product">
          <ProductList />
        </div>
      </section>
    </div>
  </Wrapper>;
};

const Wrapper = styled.section`
  .grid-filter-column {
    grid-template-columns: 0.2fr 1fr;
  }

  @media (max-width: ${({ theme }) => theme.media.mobile}) {
    .grid-filter-column {
      grid-template-columns: 1fr;
    }
  }
`;

```

```
export default Products;
```

```
// ./client/src/pages/Callback.jsx
```

```
import { useNavigate } from "react-router-dom";  
import { useCartContext } from "../context/cartContext";
```

```
const Callback = () => {  
  const { clearCart } = useCartContext();  
  const navigate = useNavigate();  
  clearCart();  
  navigate('/')  
  return(  
    <></>  
  )  
};
```

```
export default Callback;
```

```
// ./client/src/pages/Cart.jsx
```

```
import React from "react";  
import styled from "styled-components";  
import { useCartContext } from "../context/cartContext";  
import CartItem from "../components/CartItem";  
import { NavLink, useNavigate } from "react-router-dom";  
import { Button } from "../styles/Button";  
import { postAction } from "../api/actions";  
import { useAuthState } from "../config/firebase";  
  
const Cart = () => {  
  const { cart, clearCart, totalPrice, shippingFee } = useCartContext();  
  const { uid, isAuthenticated } = useAuthState();  
  
  const navigate = useNavigate();  
  const handleCheckout = async () => {  
    try {  
      if (isAuthenticated) {  
        await Promise.all(  
          cart.map(async (elem) => {  
            await postAction({  
              uid, elem, shippingFee, totalPrice  
            });  
          })  
        );  
        clearCart();  
        navigate('/');  
      }  
    }  
  }  
};
```



```

        user_id: uid,
        action_type: 'payment',
        product_id: elem.id,
    });
    })
    );
}

    navigate('/checkout');
} catch (error) {
    console.error('Error during checkout:', error);
}
};

if (cart.length === 0) {
    return (
        <EmptyDiv>
        <h3>Кошик порожній</h3>
        <NavLink to="/products">
            <Button> Продовжити шопінг </Button>
        </NavLink>
        </EmptyDiv>
    );
}
return (
    <Wrapper>
        <div className="container">
            <div className="cart-heading grid grid-five-column">
                <p>Товар</p>
                <p className="cart-hide">Ціна</p>
                <p>Кількість</p>
                <p className="cart-hide">Вартість</p>
                <p>Видалити</p>
            </div>
            <hr />

            <div className="cart-item">
                {cart.map((curElem) => {
                    return <CartItem key={curElem.id} {...curElem} />;
                })}
            </div>
            <hr />
            <div className="cart-two-button">
                <NavLink to="/products">
                    <Button> Продовжити шопінг </Button>
                </NavLink>
            </div>
        </div>
    );
}

```

```

</NavLink>
<Button className="btn btn-clear" onClick={clearCart}>
  ОЧИСТИТИ КОШИК
</Button>
</div>
<div className="order-total--amount">

  <div className="order-total--subdata">
    <div>
      <p>Вартість:</p>
      <p>
        ₴{totalPrice}
      </p>
    </div>
    <div>
      <p>Плата за доставку:</p>
      <p>
        ₴{shippingFee}
      </p>
    </div>
    <hr />
    <div>
      <p>Вартість замовлення:</p>
      <p>
        ₴{shippingFee+totalPrice}
      </p>
    </div>
  </div>
</div>
<div className="check-out">

  <Button
    className="btn btn-checkout"
    onClick={handleCheckout}
  >
    Оформити замовлення
  </Button>

</div>
</div>
</Wrapper>
);
};

const EmptyDiv = styled.div`

```

```
display: grid;
place-items: center;
height: 50vh;
```

```
h3 {
  font-size: 4.2rem;
  text-transform: capitalize;
  color: ${({ theme }) => theme.colors.black};
  font-weight: 300;
}
`;
```

```
const Wrapper = styled.section`
padding: 9rem 0;
min-height: 47.5vh;
```

```
.grid-four-column {
  grid-template-columns: repeat(4, 1fr);
}
```

```
.grid-five-column {
  grid-template-columns: repeat(4, 1fr) 0.5fr;
  text-align: center;
  align-items: center;
}
```

```
.cart-heading {
  text-align: center;
  text-transform: uppercase;
}
```

```
hr {
  margin-top: 1rem;
}
```

```
.cart-item {
  padding: 3.2rem 0;
  display: flex;
  flex-direction: column;
  gap: 3.2rem;
}
```

```
.cart-user--profile {
  display: flex;
  justify-content: flex-start;
  align-items: center;
  gap: 1.2rem;
  margin-bottom: 5.4rem;
```

```

img {
  width: 8rem;
  height: 8rem;
  border-radius: 50%;
}
h2 {
  font-size: 2.4rem;
}
}
.cart-user--name {
  text-transform: capitalize;
}
.cart-image--name {
  /* background-color: red; */
  align-items: center;
  display: grid;
  gap: 1rem;
  grid-template-columns: 0.4fr 1fr;
  text-transform: capitalize;
  text-align: left;
  img {
    max-width: 5rem;
    height: 5rem;
    object-fit: contain;
    color: transparent;
  }

  .color-div {
    display: flex;
    align-items: center;
    justify-content: flex-start;
    gap: 1rem;

    .color-style {
      width: 1.4rem;
      height: 1.4rem;

      border-radius: 50%;
    }
  }
}

.cart-two-button {
  margin-top: 2rem;

```

```

display: flex;
justify-content: space-between;

.btn-clear {
  background-color: #e74c3c;
}
}

.amount-toggle {
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 2.4rem;
  font-size: 1.4rem;

  button {
    border: none;
    background-color: ${({ theme }) => theme.colors.bgS};
    cursor: pointer;
  }

  .amount-style {
    font-size: 2.4rem;
    color: rgb(140, 154, 249);
  }
}

.remove_icon {
  font-size: 1.6rem;
  color: #e74c3c;
  cursor: pointer;
}

.order-total--amount {
  width: 100%;
  margin: 4.8rem 0;
  text-transform: capitalize;
  display: flex;

  flex-direction: column;
  justify-content: flex-end;
  align-items: flex-end;

  .order-total--subdata {
    border: 0.1rem solid #f0f0f0;

```

```

    display: flex;
    flex-direction: column;
    gap: 1.8rem;
    padding: 3.2rem;
  }
  div {
    display: flex;
    gap: 3.2rem;
    justify-content: space-between;
  }

  div:last-child {
    background: ${({ theme }) => theme.colors.bg};
  }

  div p:last-child {
    font-weight: bold;
    color: ${({ theme }) => theme.colors.heading};
  }
}

.check-out{
  display: flex;
  flex-direction: row;
  justify-content: flex-end;
  align-items: flex-end;
  .btn-checkout {
    background-color: #527eec;
  }
}

@media (max-width: ${({ theme }) => theme.media.mobile}) {
  .grid-five-column {
    grid-template-columns: 1fr 1fr 1fr;
  }
  .cart-hide {
    display: none;
  }

  .cart-two-button {
    margin-top: 2rem;
    display: flex;
    justify-content: space-between;
    gap: 2.2rem;
  }
}

```

```

.order-total--amount {
  width: 100%;
  text-transform: capitalize;
  justify-content: flex-start;
  align-items: flex-start;

.order-total--subdata {
  width: 100%;
  border: 0.1rem solid #f0f0f0;
  display: flex;
  flex-direction: column;
  gap: 1.8rem;
  padding: 3.2rem;
}
}
}
\;

```

```
export default Cart;
```

```
// ./client/src/pages/Contact.jsx
```

```

/* eslint-disable jsx-a11y/iframe-has-title */
import React from "react";
// import { NavLink } from "react-router-dom";
import styled from "styled-components";
import { Button } from "../styles/Button";
import { BsInstagram } from "react-icons/bs";

const Contact = () => {

  const url = "https://www.instagram.com";
  const handleButtonClick = () => {
    window.location.href = url;
  };
  return (
    <Wrapper>
      <h2 className="common-heading">Контакти</h2>
      <div className="instagram">
        <div>
          <div>
            <BsInstagram className="icon" />
            <h3>Наш інстаграм</h3>

```

```

    </div>
    <Button onClick={handleButtonClick}>Перейти</Button>
  </div>
</div>
</Wrapper>
)
};

```

```

const Wrapper = styled.section`
padding: 9rem 0 5rem 0;
text-align: center;
.instagram {
width: auto;
height: 10rem;
margin: 5%;
display: flex;
flex-direction: column;
color: #fff;
justify-content: center;
align-content: center;
${" /* background: ${({ theme }) => theme.colors.bg}; */"}
background: rgba(255,255,255,0.1);
text-align: center;
border-radius: 2rem;
box-shadow: rgba(0, 0, 0, 0.05) 0px 1px 2px 0px;
div {
display: flex;
flex-direction: row;
justify-content: space-around;
align-items: center;
${" /* gap: 1rem; */"}

div {
gap: 1.5rem;
}
}
}

h3 {
font-size: 3rem;
font-weight: 600;
text-transform: capitalize;
}

.icon {

```



```

    /* font-size: rem; */
    width: 6rem;
    height: 6rem;

    color: #c49952;
  }

`;

export default Contact;

// ./client/src/pages/Home.jsx

import React from "react";
import RecProducts from "../components/RecProducts";
import HeroSection from "../components/HeroSection";
import Services from "../components/Services";

const Home = () => {
  const data = {
    name: "SuperCoffee",
  }
  return (
    <>
      <HeroSection myData={data}/>
      <RecProducts />
      <Services/>

    </>
  )
}

export default Home;

// ./client/src/pages/SingleProduct.jsx

import React, { useEffect, useState } from "react";
import { useParams } from "react-router-dom";
import styled from "styled-components";

```

```

import { useProductContext } from "../context/productContext";
import PageNavigation from "../components/PageNavigation";
import { Container } from "../styles/Container";
import MyImage from "../components/MyImage";
import AddToCart from "../components/AddToCart";
import { renderDefaultCoffee } from "../functions/images";
// import MyImage from "../components/MyImage";

const SingleProduct = () => {
  const {getSingleProduct, isSingleLoading, singleProduct} = useProductContext();
  const {id} = useParams();
  const [quantity, setQuantity] = useState(1);
  const [priceB, setPriceB] = useState(0);

  const updateQuantity = (newAmount) => {
    let newQuantity = newAmount;
    let newPriceB = uah*newAmount;
    setPriceB(newPriceB)
    setQuantity(newQuantity);
  };

  const {
    name,
    price,
    description,
    categories,
    image
  } = singleProduct;

  useEffect(() => {
    getSingleProduct(id);
    // eslint-disable-next-line react-hooks/exhaustive-deps
  },[])

  if (isSingleLoading) {
    return <div className="page_loading">Loading.....</div>;
  };

  let uah = "";
  if(price)uah = price.raw

```



```

gap: 2rem;

.product-data-price {
  font-weight: bold;
  font-size: 3rem;
}
.product-data-info {
  display: flex;
  flex-direction: column;
  gap: 1rem;
  font-size: 1.8rem;

  span {
    font-weight: bold;
  }
}

hr {
  max-width: 100%;
  width: 90%;
  /* height: 0.2rem; */
  border: 0.1rem solid #000;
  color: red;
}

.product-images {
  display: flex;
  justify-content: center;
  align-items: center;
}

.page_loading {
  font-size: 3.2rem;
  display: flex;
  justify-content: center;
  align-items: center;
}

@media (max-width: ${({ theme }) => theme.media.mobile}) {
  padding: 0 2.4rem;
}
`;

export default SingleProduct;

```

```
// ./client/src/pages/Checkout.js

/* eslint-disable no-useless-escape */
import React, { useRef, useState } from "react";
import styled from "styled-components";
import { Form, Button } from 'semantic-ui-react';
import 'react-phone-input-2/lib/style.css';
import PhoneInput from 'react-phone-input-2';
import "react-datepicker/dist/react-datepicker.css";
import { Controller, useForm } from 'react-hook-form'
import { useNavigate } from "react-router-dom";
import { useCartContext } from "../context/cartContext";
import { LiqPay } from "../payment/liqpay";
import { commerce } from "../lib/commerce";

const payment = new LiqPay('sandbox_i3877493362',
'sandbox_cz5z0KfADUcCRcRpYNpbToMcXMkDbAZDs4Fvg2Nh')

const Checkout = () => {
  const { register, control, handleSubmit, formState: { errors } } = useForm()
  const inputRef = useRef(null);
  const navigate = useNavigate();
  const {cart, totalPrice} = useCartContext();
  if(cart.length === 0)navigate('/cart');
  const [newData, setNewData] = useState(0);
  const [newForm, setNewFrom] = useState(0);

  const addItemToCart = async (items) => {
    try {
      await commerce.cart.empty();
      for (const cartItem of items) {
        await commerce.cart.add(cartItem.id, cartItem.amount)
      }
    } catch (error) {
      console.error(`Error adding items to the cart: ${error.message}`);
    }
  };

  const createOrder = async (customerData, shippingData) => {
    try {
      const cartId = commerce.cart.id();

```

```

    const token = await commerce.checkout.generateToken(cartId, { type: 'cart'
  });

  const order = await commerce.checkout.capture(token.id, {
    customer: {
      firstname: customerData.firstname,
      lastname: customerData.lastname,
      email: customerData.email,
      phone: customerData.numberPhone
    },
    shipping: {
      street: shippingData.street + ", будинок " + shippingData.house + ',
квартира ' + shippingData.apartment,
    },
    "billing": [],
    payment: {
      gateway: 'test_gateway',
      card: {
        number: '4242424242424242',
        expiry_month: '02',
        expiry_year: '24',
        cvc: '123',
        postal_zip_code: '94107',
      }
    },
  });
  console.log('Order created successfully:', order);
} catch (error) {
  console.error('Error creating order:', error.message);
}
};

const onSubmit = (data) => {
  const callbackurl = window.location.origin + '/callback';
  const itemsCart = cart.map((item)=>({
    quantity: item.amount,
    price: item.price,
    cost: item.amount * item.price,
    id: item.id,
    name: item.name,
  })))
  const paymentData = {
    language: 'en',
    amount: totalPrice,
    currency: 'UAH',
  }

```

```

    description: 'Bear payment',
    rro_info: {
      items: itemsCart
    },
    action: 'pay',
    version: '3',
    order_id: Math.random()*999,
    public_key:'sandbox_i3877493362',
    result_url: callbackurl,
  }

  addItemToCart(cart);
  createOrder(data, data);
  setNewData(paymentData)
  setNewFrom(payment.cnb_form(paymentData))

}

const returnToCart = () => {
  navigate('/cart');
}

return (
  <Wrapper>
    <Form className='checkout-form'
onSubmit={handleSubmit(onSubmit)}>
      <h1>Ваші контактні дані</h1>
      <Form.Group className='form-group' >

        <Form.Field className="form-half">
          <label>Прізвище</label>
          <input
            placeholder='Шевченко'
            type="text"
            {...register("lastname", {required: true, maxLength: 20})}
          />
          {errors.lastname &&
            <p className="ui negative mini message">Прізвище є
обов'язковим і має містити менше 20 символів</p>
          }
        </Form.Field>
        <Form.Field className="form-half">
          <label>Ім'я</label>

```

```

    <input
      placeholder='Микола'
      type="text"
      {...register("firstname", {required: true, maxLength: 20})}
    />
    {errors.firstname &&
      <p className="ui negative mini message">Ім'я є обов'язковим
і має містити менше 20 символів</p>
    }
  </Form.Field>
</Form.Group>
<Form.Group className='form-group' widths='equal'>
  <Form.Field className="form-half">
    <label>Email</label>
    <input
      placeholder='xyz@example.com'
      type="email"
      {...register("email", {
        required: true,
        pattern:
/^[^<>()\\[\]\\.,:;\\s@"]+(\\.\\[<>()\\[\]\\.,:;\\s@"]+)*|(\".+\"))@((\\[[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\)|((\\[a-zA-Z\\-0-9\\]+\\.)+[a-zA-Z]{2,}))$ /
      })}
    />
    {errors.email &&
      <p className="ui negative mini message">Введіть дійсну
адресу електронної пошти</p>
    }
  </Form.Field>
  <Form.Field className="form-half">
    <label>Телефон</label>
    <Controller
      control={control}
      name="phone"
      render={({ field }) => (
        <PhoneInput
          {...field}
          innerRef={inputRef}
          country='ua'
          enableAreaCodes={true}
          placeholder='Введіть номер телефону'
          inputStyle={{ width: '100%' }}
          inputProps={{
            autoComplete: 'off',
          }}
        />
      )
    />
  </Form.Field>

```



```

        />
    })
    {...register("numberPhone", {required: true, minLength: 10,
maxLength: 20})}
    />
    {errors.numberPhone && (
        <p className="ui negative mini message">Це обов'язкове
поле</p>
    )}
    </Form.Field>
</Form.Group>
<h1>Доставка</h1>
<Form.Group className='form-group' >
    <Form.Field className="form-half">
        <label>Вулиця</label>
        <input
            placeholder='Борщагівська'
            type="text"
            {...register("street", {required: true, maxLength: 40})}
        />
        {errors.street &&
            <p className="ui negative mini message">Це обов'язкове
поле і має містити менше 40 символів</p>
        }
    </Form.Field>
    <Form.Field className="form-half-half">
        <label>Будинок</label>
        <input
            placeholder='123'
            type="text"
            {...register("house", {required: true})}
        />
        {errors.house &&
            <p className="ui negative mini message">Це обов'язкове
поле</p>
        }
    </Form.Field>
    <Form.Field className="form-half-half">
        <label>Квартира</label>
        <input
            placeholder='23'
            type="text"
            {...register("apartment", {required: false})}
        />
    </Form.Field>

```

```

        </Form.Group>
        <Button className="submitButton" disabled={newData!==0}
type='submit'>Підтвердити платіж</Button>
    </Form>
    <div className='cartItems'>
        <div className="title">
            <h3>Поточний кошик</h3>
        </div>
        <div className="cart">
            <div className="cartReturn" onClick={returnToCart}>Повернутись
до кошика</div>
            <div className="items">
                {cart.map((item, index) => (
                <div key={index} className="cart-item">
                    <div className="items-data">
                        <h4>{item.name}</h4>
                        <p>{item.amount} шт.</p>
                    </div>
                    <div className="price">
                        {item.price * item.amount} ₪
                    </div>
                </div>
                )))}
            </div>
            <div className="items-amount">
                <h4>Кінцева Вартість</h4>
                {totalPrice} ₪
            </div>
        </div>
        <div className="total"></div>
        <div className="paymentButton" style={{ display: !newData ? 'none' :
'block' }}>
            {newData ? <div dangerouslySetInnerHTML={{ __html: newForm
}}/> : ""}
        </div>
    </div>
</Wrapper>
)
}

```

```

const Wrapper = styled.section`
    display: flex;
    flex-direction: row;
    justify-content: space-between;

```

```

align-items: flex-start;
.paymentButton{
  position:relative;
  input[type="image"]{
    z-index: 20;
  }
}
.checkout-form {
  position: relative;
  background-color: #ffffff;
  padding: 2rem;
  display: flex;
  flex-direction: column;
  width: 60%;
  margin: 5rem 10rem;
  gap: 2rem;

  h1 {
    color: #000;
    font-size: 2.5rem;
  }

  .form-group {
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    gap: 1rem;
    .form-half{
      width:50%
    }
    .form-half-half{
      width:25%
    }
    input {
      width: 100%;
      height: 4.5rem;
    }
  }

  input::-webkit-outer-spin-button,
  input::-webkit-inner-spin-button {
    -webkit-appearance: none;
    margin: 0;
  }
}

```

```

}

.ui.negative.mini.message {
  background-color: #e74c3c; /* Red background color */
  color: white; /* Text color */
  padding: 8px; /* Padding around the message content */
  border-radius: 4px; /* Rounded corners */
  font-size: 12px; /* Font size */
  text-align: center; /* Center-align the text */
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2); /* Optional: Add a subtle box
shadow */
}

.submitButton{
  width: 30%;
  background-color: rgb(40, 249, 127);
  border: none;
  color: black;
  cursor:pointer;
  padding: 0% 2%;
  border: 2px solid black;
  :hover, :active{
    background-color: rgba(24, 169, 85, 0.766);
  }
}

.cartItems{
  position: fixed;
  right:0%;
  padding: 2rem;
  width:25%;
  background-color:white;
  margin: 2rem 5rem 2rem 0rem;
  min-height: 10%;
  text-align:center;
  border: 1px solid black;
  .title{
    text-align:center;
    margin: 0 0 2rem 0;
    border-bottom: 1px solid grey;
    h3{
      font-weight: 700;
      font-size: 20px;
    }
  }
}

```

```

.cart{
  color: black;
  margin: 0 0 2rem 0;

  .cartReturn{
    font-size: 12px;
    color:rgb(13, 63, 128);
    z-index: 5;
    cursor:pointer;
    margin: 0 0 2rem 0;
  }
  .items{
    width:100%;
    display:flex;
    flex-direction:column;
    border-bottom: 1px solid grey;
    .cart-item{
      width:100%;
      display:flex;
      flex-direction:row;
      justify-content:space-between;
      .items-data{
        display:flex;
        flex-direction:column;
        text-align:left;
        h4{
          color: red;
          font-size: 11px;
        }
        p{
          color: grey;
          font-size: 9px;
        }
      }
      .price{
        font-size: 13px;
      }
    }
  }
  .items-amount{
    font-size: 15px;
    padding:1rem;
    padding-bottom:0;
  }
}

```

```

    }
  },
  // ...

export default Checkout;

// ./client/src/pages/Sign/Login.jsx

import { useState } from "react";
import { getAuth, signInWithEmailAndPassword, signInWithPopup,
  GoogleAuthProvider } from "firebase/auth";
import { auth } from "../../config/firebase";
import styled from "styled-components";
import { useNavigate } from "react-router-dom";

const SignInContainer = styled.div`
  width: 40vw;
  padding: 2em 0;
  min-width: 300px;
  margin: 3em auto;
  border: 1px solid grey;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  box-shadow: 0 0 40px rgba(0, 0, 0, 0.5);
`;

const Input = styled.input`
  width: 70%;
  padding: 0.5em 1em;
  margin: 0.5em 0;
  box-sizing: border-box;
  height: 4vw;
  border-radius: 30px;
`;

const Button = styled.button`
  width: 70%;
  padding: 0.5em;
  margin: 0.5em 0;
  box-sizing: border-box;
  background-color: #004ea1;

```

```

color: #fff;
border: none;
cursor: pointer;
height: 4vw;
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
border-radius: 30px;

```

```

&:hover {
  background-color: #2364a9;
}
`;

```

```

const GoogleButton = styled(Button)`
  background-color: #fff;
  color: #000;
  border: 1px solid #000;

```

```

  svg {
    height: 70%;
    padding-right: 1em;
  }
`;

```

```

const ErrorMessage = styled.p`
  margin-top: 1em;
`;

```

```

const Login = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [errMsg, setErrMsg] = useState("");
  const navigate = useNavigate();

```

```

  const signIn = () => {
    signInWithEmailAndPassword(auth, email, password)
      .then((data) => {
        if (data) navigate("/");
      })
      .catch((error) => {
        console.log(error.code);
        switch (error.code) {

```

```

    case "auth/invalid-email":
      setErrMsg("Invalid email");
      break;
    case "auth/user-not-found":
      setErrMsg("No account with that email was found");
      break;
    case "auth/wrong-password":
      setErrMsg("Incorrect password");
      break;
    default:
      setErrMsg("Email or password was incorrect");
      break;
  }
});
};

const signInWithGoogle = () => {
  const provider = new GoogleAuthProvider();
  signInWithPopup(getAuth(), provider)
    .then((result) => {
      if (result) navigate("/");
    })
    .catch((error) => {
      console.log(error.code);
    });
};

return (
  <SignInContainer>
    <h1>Login</h1>
    <Input type="text" placeholder="Email" value={email} onChange={(e) =>
setEmail(e.target.value)} />
    <Input type="password" placeholder="Password" value={password}
onChange={(e) => setPassword(e.target.value)} />
    <ErrorMessage {errMsg} </ErrorMessage>
    <Button onClick={signIn}>Submit</Button>
    <GoogleButton onClick={signInWithGoogle}>
    <svg viewBox="-0.5 0 48 48" version="1.1">

    <title>Google-color</title>
    <desc>Created with Sketch.</desc>
    <defs>

    </defs>
  </SignInContainer>
);

```



```

    <g id="Icons" stroke="none" strokeWidth="1" fill="none"
fillRule="evenodd">
      <g id="Color-" transform="translate(-401.000000, -860.000000)">
        <g id="Google" transform="translate(401.000000, 860.000000)">
          <path d="M9.82727273,24 C9.82727273,22.4757333
10.0804318,21.0144 10.5322727,19.6437333 L2.62345455,13.6042667
C1.08206818,16.7338667 0.213636364,20.2602667 0.213636364,24
C0.213636364,27.7365333 1.081,31.2608 2.62025,34.3882667
L10.5247955,28.3370667 C10.0772273,26.9728 9.82727273,25.5168
9.82727273,24" id="Fill-1" fill="#FBBC05">
            </path>

            <path d="M23.7136364,10.1333333 C27.025,10.1333333
30.0159091,11.3066667 32.3659091,13.2266667 L39.2022727,6.4
C35.0363636,2.77333333 29.6954545,0.533333333 23.7136364,0.533333333
C14.4268636,0.533333333 6.44540909,5.84426667 2.62345455,13.6042667
L10.5322727,19.6437333 C12.3545909,14.112 17.5491591,10.1333333
23.7136364,10.1333333" id="Fill-2" fill="#EB4335">
            </path>

            <path d="M23.7136364,37.8666667 C17.5491591,37.8666667
12.3545909,33.888 10.5322727,28.3562667 L2.62345455,34.3946667
C6.44540909,42.1557333 14.4268636,47.4666667 23.7136364,47.4666667
C29.4455,47.4666667 34.9177955,45.4314667 39.0249545,41.6181333
L31.5177727,35.8144 C29.3995682,37.1488 26.7323182,37.8666667
23.7136364,37.8666667" id="Fill-3" fill="#34A853">
            </path>

            <path d="M46.1454545,24 C46.1454545,22.6133333
45.9318182,21.12 45.6113636,19.7333333 L23.7136364,19.7333333
L23.7136364,28.8 L36.3181818,28.8 C35.6879545,31.8912
33.9724545,34.2677333 31.5177727,35.8144 L39.0249545,41.6181333
C43.3393409,37.6138667 46.1454545,31.6490667 46.1454545,24" id="Fill-4"
fill="#4285F4">
            </path>
          </g>
        </g>
      </g>
    </svg>
    Sign In With Google
  </GoogleButton>
  <p>
    Not registered yet? <a href="/register">Register</a>
  </p>
</SignInContainer>

```

```

    );
  };

export default Login;

// ./client/src/pages/Sign/Register.jsx

import { useState } from "react";
import { getAuth, createUserWithEmailAndPassword, updateProfile,
signInWithPopup, GoogleAuthProvider } from "firebase/auth";
import styled from "styled-components";
import { useNavigate } from "react-router-dom";
import { auth } from "../../config/firebase";

const SignInContainer = styled.div`
  width: 40vw;
  padding: 2em 0;
  min-width: 300px;
  margin: 3em auto;
  border: 1px solid grey;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  box-shadow: 0 0 40px rgba(0, 0, 0, 0.5);
`;

const Input = styled.input`
  width: 70%;
  padding: 0.5em 1em;
  margin: 0.5em 0;
  box-sizing: border-box;
  height: 4vw;
  border-radius: 30px;
`;

const Button = styled.button`
  width: 70%;
  padding: 0.5em;
  margin: 0.5em 0;
  box-sizing: border-box;
  background-color: #004ea1;
  color: #fff;
  border: none;

```

```

    cursor: pointer;
    height: 4vw;
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: center;
    border-radius: 30px;

    &:hover {
      background-color: #2364a9;
    }
  `;

const GoogleButton = styled(Button)`
  background-color: #fff;
  color: #000;
  border: 1px solid #000;

  svg {
    height: 70%;
    padding-right: 1em;
  }
`;

const ErrorMessage = styled.p`
  margin-top: 1em;
`;

const Register = () => {
  const navigate = useNavigate();
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [nickname, setNickname] = useState("");
  const [errMsg, setErrMsg] = useState("");

  const register = () => {
    createUserWithEmailAndPassword(auth, email, password)
      .then((data) => {
        if (data) {
          if (nickname) {
            updateProfile(auth.currentUser, { displayName: nickname })
              .then(() => {
                console.log("Successfully updated profile!");
              })
          }
        }
      })
  }
}

```

```

        .catch((error) => {
            console.error("Error updating profile:", error);
        });
    }
    console.log("Successfully registered!");
    navigate('/');
}
})
.catch((error) => {
    console.log(error.code);
    switch (error.code) {
        case "auth/invalid-email":
            setErrMsg("Invalid email");
            break;
        case "auth/user-not-found":
            setErrMsg("No account with that email was found");
            break;
        case "auth/wrong-password":
            setErrMsg("Incorrect password");
            break;
        default:
            setErrMsg("Email or password was incorrect");
            break;
    }
});
};

const signInWithGoogle = () => {
    const provider = new GoogleAuthProvider();
    signInWithPopup(getAuth(), provider)
        .then((result) => {
            if (result) navigate("/");
        })
        .catch((error) => {
            console.log(error.code);
        });
};

return (
    <SignInContainer>
        <h1>Register</h1>
        <Input type="text" placeholder="Nickname" value={nickname}
onChange={(e) => setNickname(e.target.value)} />
        <Input type="text" placeholder="Email" value={email} onChange={(e) =>
setEmail(e.target.value)} />

```

```

    <Input type="password" placeholder="Password" value={password}
onChange={(e) => setPassword(e.target.value)} />
    <ErrorMessage>{errMsg}</ErrorMessage>
    <Button onClick={register}>Submit</Button>
    <GoogleButton onClick={signInWithGoogle}>
    <svg viewBox="-0.5 0 48 48" version="1.1">

        <title>Google-color</title>
        <desc>Created with Sketch.</desc>
        <defs>

            </defs>
            <g id="Icons" stroke="none" strokeWidth="1" fill="none"
fillRule="evenodd">
                <g id="Color-" transform="translate(-401.000000, -860.000000)">
                    <g id="Google" transform="translate(401.000000, 860.000000)">
                        <path d="M9.82727273,24 C9.82727273,22.4757333
10.0804318,21.0144 10.5322727,19.6437333 L2.62345455,13.6042667
C1.08206818,16.7338667 0.213636364,20.2602667 0.213636364,24
C0.213636364,27.7365333 1.081,31.2608 2.62025,34.3882667
L10.5247955,28.3370667 C10.0772273,26.9728 9.82727273,25.5168
9.82727273,24" id="Fill-1" fill="#FBBC05">
                            </path>

                            <path d="M23.7136364,10.1333333 C27.025,10.1333333
30.0159091,11.3066667 32.3659091,13.2266667 L39.2022727,6.4
C35.0363636,2.77333333 29.6954545,0.533333333 23.7136364,0.533333333
C14.4268636,0.533333333 6.44540909,5.84426667 2.62345455,13.6042667
L10.5322727,19.6437333 C12.3545909,14.112 17.5491591,10.1333333
23.7136364,10.1333333" id="Fill-2" fill="#EB4335">
                                </path>

                                <path d="M23.7136364,37.8666667 C17.5491591,37.8666667
12.3545909,33.888 10.5322727,28.3562667 L2.62345455,34.3946667
C6.44540909,42.1557333 14.4268636,47.4666667 23.7136364,47.4666667
C29.4455,47.4666667 34.9177955,45.4314667 39.0249545,41.6181333
L31.5177727,35.8144 C29.3995682,37.1488 26.7323182,37.8666667
23.7136364,37.8666667" id="Fill-3" fill="#34A853">
                                    </path>

                                    <path d="M46.1454545,24 C46.1454545,22.6133333
45.9318182,21.12 45.6113636,19.7333333 L23.7136364,19.7333333
L23.7136364,28.8 L36.3181818,28.8 C35.6879545,31.8912
33.9724545,34.2677333 31.5177727,35.8144 L39.0249545,41.6181333

```

```

C43.3393409,37.6138667 46.1454545,31.6490667 46.1454545,24" id="Fill-4"
fill="#4285F4">
    </path>
  </g>
</g>
</g>
</svg>
  Sign In With Google
</GoogleButton>
<p>
  Already have an account? <a href="/login">Login</a>
</p>
</SignInContainer>
);
};

export default Register;

// ./server/server.js

const http = require('node:http');
const sequelize = require('./config/db');
const router = require('./routes/router');

require('dotenv').config();

const port = process.env.PORT || 8000;
const hostname = process.env.HOSTNAME || '127.0.0.1'

const server = http.createServer((req, res) => {
  router(req, res);
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`)
});

server.on('error', err => {
  if (err.code === 'EACCES') {
    console.log(`No access to port: ${port}`);
  }
});

sequelize.authenticate().then(() => {

```

```

    console.log('Connection has been established successfully.');
```

```

  }).catch((error) => {
    console.error('Unable to connect to the database: ', error);
  });
```

```
// ./server/config/db.js
```

```

const Sequelize = require("sequelize");
require('dotenv').config();
```

```

const username = process.env.DATABASE_USERNAME;
const password = process.env.DATABASE_PASSWORD;
const host = process.env.DATABASE_HOST;
const port = process.env.DATABASE_PORT;
```

```

const sequelize = new Sequelize(
  'SuperCoffee',
  username,
  password,
  {
    host,
    port,
    dialect: 'mysql'
  }
);
```

```
module.exports = sequelize;
```

```
// ./server/utills/recommendation.functions.js
```

```

const formatData = data => {
  let formatted = [];

  for (const product of data) {
    let tmpObj = {};

    let content = product.name.toLowerCase() + " " +
product.categories[0].name.toLowerCase();

    tmpObj = {
      id: product.id,
      content,
    };
  }
};
```

```

        formatted.push(tmpObj);
    }

    return formatted;
}

const getRatingForAction = (action) => {
    switch (action) {
        case 'click':
            return 1;
        case 'cart':
            return 3;
        case 'payment':
            return 5;
        default:
            return 0;
    }
};

module.exports = { formatData, getRatingForAction };

// ./server/models/action.model.js

const { DataTypes } = require('sequelize');
const sequelize = require('../config/db');

const Action = sequelize.define('actions', {
    action_id: {
        type: DataTypes.INTEGER,
        autoIncrement: true,
        primaryKey: true,
    },
    user_id: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    product_id: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    action_type: {
        type: DataTypes.STRING,
        allowNull: false,
    }
});

```



```

    },
    timestamp: {
      type: DataTypes.DATE,
      allowNull: false,
      defaultValue: DataTypes.NOW,
    },
  });

```

```

module.exports = Action;

```

```

// ./server/controllers/action.controller.js

```

```

const sequelize = require('../config/db');
const Action = require('../models/action.model');
const { respondWithJSON, respondWithError } = require('../routes/responses');

```

```

async function postAction(req, res) {
  let data = '';

  req.on('data', (chunk) => {
    data += chunk;
  });

  req.on('end', () => {
    try {
      const { user_id, product_id, action_type } = JSON.parse(data);

      if (!user_id || !product_id || !action_type) {
        return respondWithError(res, 400, 'Invalid request. Missing required
fields.');
```

```

      }

      sequelize.sync()
        .then(() => {
          Action.create({
            user_id,
            product_id,
            action_type,
          }).then(newAction => {
            respondWithJSON(res, 201, newAction);
          }).catch((error) => {
            console.error('Failed to create a new record : ', error);
          });

```

```

    }).catch((error) => {
      console.error('Unable to create table : ', error);
    });

    } catch (error) {
      console.error('Error creating action:', error);
      respondWithError(res, 500, 'Internal Server Error');
    }
  });
}

module.exports = { postAction };

// ./server/controllers/recommendation.controller.js

const { respondWithJSON, respondWithError } = require('../routes/responses');
const { createRecommendations } = require('../services/recommendation.service');

async function postRecs(req, res) {
  let data = "";

  req.on('data', (chunk) => {
    data += chunk;
  });

  req.on('end', async () => {
    try {

      const { user_id, products } = JSON.parse(data);

      if (!products) {
        return respondWithError(res, 400, 'Invalid request. Missing required
fields.');
```

```

        console.error('Error creating action:', error);
        respondWithError(res, 500, 'Internal Server Error');
    }
});
}

module.exports = { postRecs };

// ./server/routes/responses.js

function respondWithJSON(res, statusCode, data) {
    res.writeHead(statusCode, getCommonHeaders());
    res.end(JSON.stringify(data));
}

function respondWithError(res, statusCode, errorMessage) {
    res.writeHead(statusCode, getCommonHeaders());
    res.end(JSON.stringify({ error: errorMessage }));
}

function responseWithText(res, statusCode, text) {
    res.writeHead(statusCode, getCommonHeaders());
    res.end(text);
}

function getCommonHeaders() {
    return {
        'Content-Type': 'text/plain',
        'Access-Control-Allow-Origin': '*',
        "Access-Control-Allow-Methods": "GET, POST, PATCH, PUT, DELETE,
OPTIONS",
        "Access-Control-Allow-Headers": "Content-Type, Access-Control-Allow-
Headers, Authorization, X-Requested-With, Origin",

    };
}

module.exports = {
    respondWithJSON,
    respondWithError,
    responseWithText,
    getCommonHeaders,
};

```

```

// ./server/routes/router.js

const actionController = require('../controllers/action.controller');
const recommendationController =
require('../controllers/recommendation.controller');
const { responseWithText, respondWithError } = require('../responses');

function router(req, res) {
  const path = req.url;

  switch (path) {
    case '/action':
      if (req.method === 'POST') {
        actionController.postAction(req, res);
      } else {
        respondWithError(res, 405, 'Invalid method for /action');
      }
      break;

    case '/recommendation':
      if (req.method === 'POST') {
        recommendationController.postRecs(req, res);
      } else {
        respondWithError(res, 405, 'Invalid method for /action');
      }
      break;

    default:
      responseWithText(res, 200, 'Not Found');
  }
}

module.exports = router;

// ./server/services/recommendation.service.js

const sequelize = require('../config/db');
const Action = require('../models/action.model');
const { formatData, getRatingForAction } =
require('../utils/recommendation.functions');
const natural = require('natural');
const TfIdf = natural.TfIdf;

```

```

const tokenizer = new natural.WordTokenizer();

const createRecommendations = async (id, products) => {
  const formatProducts = formatData(products);
  let recommendations = [];
  let userActions;
  let result = [];

  try {
    await sequelize.sync();
    let actions;

    if (id) {
      actions = await Action.findAll({
        where: {
          user_id: id,
        },
      });
    } else {
      actions = await Action.findAll();
    }

    userActions = actions;

    const productActionsMatrix = {};

    userActions.forEach((action) => {
      const { product_id, action_type } = action;
      if (!productActionsMatrix[product_id]) {
        productActionsMatrix[product_id] = [];
      }

      productActionsMatrix[product_id].push(action_type);
    });

    const tfidf = new TfIdf();
    formatProducts.forEach((product) => {
      if (productActionsMatrix[product.id]) {
        const actions = productActionsMatrix[product.id].join(' ');
        tfidf.addDocument(tokenizer.tokenize(actions), { product });
      }
    });
  }
};

```

```

    const userPreferences = userActions.map((action) =>
action.action_type).join(' ');
    const userVector = tokenizer.tokenize(userPreferences);

    tfidf.tfidfjs(userVector, (index, measure, { product }) => {
    recommendations.push({ product, score: measure });
    });

    recommendations.sort((a, b) => b.score - a.score);

    let recs = recommendations.slice(0, 3);
    let recProducts = [];

    if (recs.length && recs.length > 0) {
    recs.map((rec) => {
    recProducts.push(products.find((product) => product.id ===
rec.product.id));
    });
    }
    while (recProducts.length < 3) {
    const sortedProducts = products.sort((a, b) => b.updated - a.updated);
    const remainingProducts = sortedProducts.filter((product) =>
!recProducts.includes(product.id));
    if (remainingProducts.length > 0) {
    recProducts.push(remainingProducts[0]);
    } else {
    break;
    }
    }

    result = recProducts;
  } catch (error) {
    console.error('Error in createRecommendations:', error);
  }

  return result;
};

module.exports = { createRecommendations };

```