```
//TODO: App: zrobic logowanie uzytkownika oraz wykorzystanie do tego localStorage


import { BrowserRouter, Route, Routes, useNavigate } from "react-router-dom";
import MovieSearch from "./Components/movieSearch/MovieSearch";
import "./App.css";
import WelcomePage from "./Components/welcomePage/WelcomePage";
import WatchListLibrary from "./Components/watchlist/WatchListLibrary";
import { UserContext, UserProvider } from "./Components/contex/UserProvider";


import { CssBaseline } from "@mui/material";
import Test from "./Components/template/test";
import SignIn from "./Components/signIn/SignIn";
import { useContext } from "react";
const App = () => {
  return (
    <UserProvider>
      <CssBaseline />
      <BrowserRouter>
        <Routes>
          <Route exact path="/*" element={<WelcomePage />} />
          <Route path="/" element={<WelcomePage />} />
          <Route path="/movies" element={<MovieSearch />} />
          <Route path="/watchlist" element={<WatchListLibrary />} />
          <Route path="/test" element={<Test />} />
          <Route path="/signin" element={<SignIn />} />
        </Routes>
      </BrowserRouter>
    </UserProvider>
  );
};
export default App;
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";
import "./index.css";
import App from "./App.jsx";
import "../node_modules/bootstrap/dist/css/bootstrap.min.css";
createRoot(document.getElementById("root")).render(
  <StrictMode>
    <App />
  </StrictMode>
);
//TODO: UserProvider: walidowanie zalogowanie uzytkownika do zapisywania filmow do watchlisty oraz wyswietlanie, samej watchlisty

import { useState, createContext, useEffect } from "react";

export const UserContext = createContext();

export const UserProvider = ({ children }) => {
  const getLocalStorageUser = () => {
    const userLS = localStorage.getItem("user");
    return userLS
      ? JSON.parse(userLS)
      : {
          name: "User1",
          password: "password123",
          watchlist: [],
          rentedMovies: [],
        };
  };
  const [user, setUser] = useState(getLocalStorageUser);
  const addMovie = (movie) => {
    setUser({
      ...user,
      watchlist: [...user.watchlist, movie],
    });
  };
  const removeMovie = (movie) => {
    setUser({
      ...user,
      watchlist: user.watchlist.filter((e) => e.imdbID != movie.imdbID),
    });
  };
  const addToRentedMovies = (movie) => {
    setUser({
      ...user,
      rentedMovies: [...user.rentedMovies, movie],
    });
  };
  useEffect(() => {
    localStorage.setItem("user", JSON.stringify(user));
  }, [user]);
  return (
    <UserContext.Provider
```

```
      value={{ user, addMovie, removeMovie, addToRentedMovies, setUser }}
    >
      {children}
    </UserContext.Provider>
  );
};
import "../../Styles/MovieCard.css";
import NoImageAvailable from "../../assets/NoImageAvailable.png";
import Plus from "../../assets/plus.svg?react";
import { useContext, useEffect, useRef, useState } from "react";
import { useInView } from "framer-motion";
import { UserContext } from "../contex/UserProvider";

const MovieCard = ({ movieInfo }) => {
  const [isImageLoaded, setIsImageLoaded] = useState(false);
  const [error, setError] = useState(false);
  const cardRef = useRef(null);
  const isCardInView = useInView(cardRef, { once: true });
  const { addMovie, removeMovie, user } = useContext(UserContext);
  const isMovieOnWatchlist = (movie) => {
    if (
      user.watchlist.filter((e) => e.imdbID === movie.imdbID).length > 0
    ) {
      return true;
    }
    return false;
  };
```

```
const [isAddedToWatchList, setisAddedToWatchList] = useState(false);

const onAddMovieClick = (movie) => {
  const isRented = user.rentedMovies.some(
    (m) => m.imdbID === movie.imdbID
  );

  if (isRented) return;

  if (isAddedToWatchList) {
    removeMovie(movie);
  } else {
    addMovie(movie);
  }

  setisAddedToWatchList(!isAddedToWatchList);
};
user
  ? useEffect(() => {
      user
        ? setisAddedToWatchList(isMovieOnWatchlist(movieInfo))
        : null;
    }, [user.watchlist])
  : null;
return (
  <div
    className={"movieCard"}
    style={{ opacity: isCardInView ? 1 : 0, transition: "3s" }}
  ref={cardRef}
>
  <div className="DivPoster">
    {user ? (
      <button
        className={
          isAddedToWatchList
            ? "addButton activeButton"
            : "addButton"
        }
      >
        <Plus
          width={30}
          height={30}
          fill={"#EEE"}
          onClick={() => onAddMovieClick(movieInfo)}
        />
      </button>
    ) : null}

    <a
      target="_blank"
      href={`https://www.imdb.com/title/${movieInfo.imdbID}`}
    >
      <div className="movieInfo">
        <b>
          <span>{movieInfo.Title}</span>
        </b>
```

```
            <br />
            <span>{movieInfo.Year}</span>
          </div>
          <div className="DivPosterFlex">
            <img
              width={200}
              className="poster"
              src={
                movieInfo.Poster !== "N/A" &&
                isImageLoaded &&
                !error
                  ? movieInfo.Poster
                  : NoImageAvailable
              }
              onLoad={() => setIsImageLoaded(true)}
              onError={() => setError(true)}
            />
          </div>
        </a>
      </div>
    </div>
  );
};

export default MovieCard;
import { useContext, useEffect, useState } from "react";
import "../../Styles/MovieSearch.css";
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";

import Col from "react-bootstrap/Col";

import { options } from "../../config/config";

import MovieCard from "./MovieCard";

import NavigationBar from "../navbar/Navbar";

import "../../Styles/SideBar.css";

import "../../Styles/gridHeader.css";

import { UserContext } from "../contex/UserProvider";


const MovieSearch = () => {

  const apiKey = options.apikey;


  const [movieData, setMovieData] = useState([]);

  const [search, setSearch] = useState("");


  useEffect(() => {

    fetch(

      `http://www.omdbapi.com/?apikey=${apiKey}&s=${

        search != "" ? search : "godfather"

      }&plot=full`

    )

      .then((res) => res.json())

      .then((res) => {

        setMovieData([]);


        if (res.Response == "True") {

          setMovieData((data) => [...data, ...res.Search]);

        }
      })
      .catch((err) => console.error(err));
  }, [search]);

  return (
    <>
      <NavigationBar
        searchCallback={setSearch}
        searchValue={search}
        SearchBar={true}
      ></NavigationBar>
      <Container fluid className="body">
        <Row>
          <Col className="pt-3">
            {search == "" ? (
              <h1>Example results for phrase: "Godfather"</h1>
            ) : (
              <h1>Results for phrase: "{search}"</h1>
            )}
          </Col>
        </Row>
        <Row>
          <Col>
            <Container className="movieGrid">
              {movieData && movieData.length > 0 ? (
                movieData
                  .reduce((cols, e, i) => {
                    if (i % 3 === 0) {
                      cols.push([]);
                    }
                    cols[cols.length - 1].push(
                      <Col lg={3} xs={6} key={i}>
                        <MovieCard movieInfo={e} />
                      </Col>
                    );
                    return cols;
                  }, [])
                  .map((rowContent, i) => (
                    <Row
                      className="justify-content-center align-items-start"
                      key={i}
                    >
                      {rowContent}
                    </Row>
                  ))
              ) : search != "" ? (
                <h1 className="text-danger">No movies found</h1>
              ) : null}
            </Container>
          </Col>
        </Row>
        <div></div>
      </Container>
    </>
  );
};
```

```
export default MovieSearch;
//TODO: Navbar: Navbar nie jest w 100% responsywny, naprawic serachbar, najprawdopodobniej zrobic hamburgera

import Container from "react-bootstrap/Container";
import Navbar from "react-bootstrap/Navbar";
import Nav from "react-bootstrap/Nav";
import { Button, Form, NavDropdown, Offcanvas } from "react-bootstrap";
import { Row, Col } from "react-bootstrap";
import "../../Styles/Navbar.css";
import PlayerIcon from "../../assets/player.svg?react";
import Logo from "../template/Logo";
import { Link } from "react-router-dom";
import VideoBackground from "../template/VideoBackground";

const NavigationBar = ({ searchCallback, searchValue, SearchBar = false }) => {
  return (
    <>
      <Navbar
        expand="lg"
        className="bg-body-tertiary mb-3 NavigationBar"
        variant="dark"
        data-bs-theme="dark"
      >
        <Container fluid>
          <Navbar.Brand>
            <Logo />
```

```
</Navbar.Brand>
<Navbar.Toggle aria-controls={`offcanvasNavbar-expand`} />
<Navbar.Offcanvas
  id={`offcanvasNavbar-expand`}
  aria-labelledby={`offcanvasNavbarLabel-expand`}
  placement="end"
  data-bs-theme="dark"
  className="offCanvasNavigationBar"
>
  <Offcanvas.Header closeButton>
    <Offcanvas.Title id={`offcanvasNavbarLabel-expand`}>
      <Logo />
    </Offcanvas.Title>
  </Offcanvas.Header>
  <Offcanvas.Body>
    <Nav className="justify-content-start flex-grow-1 pe-3">
      <Link to={"/"} className="nav-link text-center">
        <span>Home</span>
      </Link>
    </Nav>
    <Nav className="justify-content-end flex-grow-1 pe-3">
      {SearchBar ? (
        <Form className="d-flex">
          <Navbar.Text className="px-3">
            Search:
          </Navbar.Text>
          <Form.Control
            value={searchValue}
      type="text"
      placeholder="Movie Title"
      className=" mr-sm-2 me-1 searchInput"
      onInput={(e) =>
        searchCallback(e.target.value)
      }
    />
  </Form>
) : (
  <Link
    to={"/movies"}
    className="nav-link text-center"
  >
    <span>Search for movies</span>
  </Link>
)}

<Link
  to={"/watchlist"}
  className="nav-link text-center"
>
  <span>Watchlist</span>{" "}
  <PlayerIcon
    width={15}
    height={15}
    fill={"#0A900A"}
  />
</Link>
```

```
        </Nav>
      </Offcanvas.Body>
    </Navbar.Offcanvas>
  </Container>
</Navbar>
{/* <Navbar className="NavigationBar">
  <Container>
    <Nav className="align-items-center">
      <Link
        to="/"
        className="navbar-brand"
        style={{ textDecoration: "none" }}
      >
        <Logo />
      </Link>
      <Link className="nav-item nav-link ms-5" to="/movies">
        <span className="btn btn-light text-dark">
          Search for a movie
        </span>
      </Link>
    </Nav>
    <Nav className="justify-content-end align-items-center">
      <Link
        className="nav-item nav-link ms-5"
        to="/watchlist"
      >
        <span>WatchList</span>{" "}
        <PlayerIcon
  );

};

                fill={"#0A900A"}

                width={20}

                height={20}

              />

            </Link>

            <Navbar.Text className="px-3">Search:</Navbar.Text>

            <Form>

              <Row>

                <Col xs="auto">

                  <Form.Control

                    value={searchValue}

                    type="text"

                    placeholder="Search"

                    className=" mr-sm-2 searchInput"

                    onInput={(e) =>

                      searchCallback(e.target.value)

                    }

                  />

                </Col>

              </Row>

            </Form>

          </Nav>

        </Container>

      </Navbar> */}

    </>

  );

};
```

```
export default NavigationBar;
//TODO: Dokonczyc strone logowania i rejestracji
import { Col, Container, Row } from "react-bootstrap";
import VideoBackground from "../template/VideoBackground";
import "../../Styles/WelcomePage.css";
import SignInForm from "./SignInForm";
import Logo from "../template/Logo";
import { Link, useNavigate } from "react-router-dom";
import { useContext, useEffect } from "react";
import { UserContext } from "../contex/UserProvider";
const SignIn = () => {
  const { user } = useContext(UserContext);
  const navigate = useNavigate();
  const isLogged = () => {
    user ? navigate("/") : null;
  };
  useEffect(() => {
    isLogged();
  }, []);
  return (
    <VideoBackground>
      <Container
        fluid
        style={{ position: "fixed", width: "100%", height: "100%" }}
      >
        <Row style={{ height: "100%" }}>
          <Col
            xs={12}
            lg={6}
            className="d-flex align-items-center justify-content-center"
          >
            <Link to="/">
              <Logo size={80} />
            </Link>
          </Col>
          <Col
            style={{ minWidth: "500px" }}
            className="menuWelcomePage d-flex justify-content-center align-items-center"
            xs={12}
            lg={6}
          >
            <SignInForm />
          </Col>
        </Row>
      </Container>
    </VideoBackground>
  );
};

export default SignIn;
import Button from "react-bootstrap/Button";
import Form from "react-bootstrap/Form";
import "../../Styles/SignIn.css";
import { useContext, useState } from "react";
import { UserContext } from "../contex/UserProvider";
import { Navigate, useNavigate } from "react-router-dom";

function SignInForm() {
  const navigate = useNavigate();
  const { setUser } = useContext(UserContext);
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const sendForm = () => {
    if (username == "" || password == "") {
      alert("Inputs can't be empty");
      return;
    }
    const user = {
      name: username,
      password: password,
      watchlist: [],
      rentedMovies: [],
    };
    setUser(user);
    navigate("/");
  };
  return (
    <Form className="SignInForm">
      <Form.Group className="mb-3" controlId="formBasicEmail">
        <Form.Label>Username</Form.Label>
        <Form.Control
          type="text"
          placeholder="Enter username"
          value={username}
          onChange={(e) => setUsername(e.target.value)}
        />
      </Form.Group>

      <Form.Group className="mb-3" controlId="formBasicPassword">
        <Form.Label>Password</Form.Label>
        <Form.Control
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          type="password"
          placeholder="Password"
        />
      </Form.Group>

      <Button variant="primary" type="submit" onClick={() => sendForm()}>
        Submit
      </Button>
    </Form>
  );
}

export default SignInForm;
import "../../Styles/Logo.css";
const Logo = ({ size }) => {
  const fontSize = size ? size : 30;
  return (
    <div className="logo text-center" style={{ fontSize: fontSize + "px" }}>
```

```
      <span>
        Movie<span className="Rental">Rental</span>.com
      </span>
    </div>
  );
};
export default Logo;
import BackgroundWelcomePage from "../../assets/BackgroundWelcomePage.mp4";
import "../../Styles/WelcomePage.css";

const VideoBackground = ({ children }) => {
  return (
    <div className="backgroundWelcomePage">
      <video muted loop autoPlay className="backgroundVideo">
        <source src={BackgroundWelcomePage} type="video/mp4" />
      </video>
      {children}
    </div>
  );
};
export default VideoBackground;
import RentalOptions from "./RentalOptions";
import RentalStatus from "./RentalStatus";
import "../../Styles/watchlistMovie.css";

const MovieCardLayout = ({
  movie,
  movieDetails,
  rented,
  timeLeft,
  videoQuality,
  setVideoQuality,
  audioQuality,
  setAudioQuality,
  subtitles,
  setSubtitles,
  rentalTime,
  setRentalTime,
  getTotalPrice,
  startRental,
  imageUnavailable,
}) => {
  return (
    <div
      className="card text-white bg-dark mb-3 shadow"
      style={{ maxWidth: "540px" }}
    >
      <div className="row g-0">
        <div className="col-md-4">
          <img
            src={movie.Poster || imageUnavailable}
            alt={movie.Title}
            className="img-fluid rounded-start"
            style={{ objectFit: "cover", height: "100%" }}
          />
        </div>
```

```
<div className="col-md-8">
  <div className="card-body">
    <h5 className="card-title">{movie.Title}</h5>
    <p className="card-text text-light-emphasis">
      Time:{" "}
      {movieDetails ? movieDetails.Runtime : "Loading..."}
    </p>
    <RentalOptions
      rented={rented}
      videoQuality={videoQuality}
      setVideoQuality={setVideoQuality}
      audioQuality={audioQuality}
      setAudioQuality={setAudioQuality}
      subtitles={subtitles}
      setSubtitles={setSubtitles}
      rentalTime={rentalTime}
      setRentalTime={setRentalTime}
    />
    <p
      className={` ${
        rented ? " total-price-rented " : " text-white "
      }card-text` }
    >
      <strong>Total price: </strong> {getTotalPrice()} zł
    </p>
    <RentalStatus
      rented={rented}
      timeLeft={timeLeft}
```

```
              startRental={startRental}
              rentalTime={rentalTime}
            />
          </div>
        </div>
      </div>
    </div>
  );
};

export default MovieCardLayout;
import prices from "../../assets/priceData/prices.json";
import "../../Styles/watchlistMovie.css";
const RentalOptions = ({
  rented,
  videoQuality,
  setVideoQuality,
  audioQuality,
  setAudioQuality,
  subtitles,
  setSubtitles,
  rentalTime,
  setRentalTime,
}) => {
  return (
    <>
      <div className="mb-2">
        <label className="form-label">Video quality</label>
  <select
    disabled={rented}
    className="form-select bg-dark text-white border-light"
    value={videoQuality}
    onChange={(e) => setVideoQuality(e.target.value)}
  >
    {Object.entries(prices["video-quality"]).map(
      ([key, val]) => (
        <option key={key} value={key}>
          {val.name} ({val.price} zł)
        </option>
      )
    )}
  </select>
</div>
<div className="mb-2">
  <label className="form-label">Audio</label>
  <select
    disabled={rented}
    className="form-select bg-dark text-white border-light"
    value={audioQuality}
    onChange={(e) => setAudioQuality(e.target.value)}
  >
    {Object.entries(prices["audio"]).map(([key, val]) => (
      <option key={key} value={key}>
        {val.name} ({val.price} zł)
      </option>
    ))}
  </select>
</div>
<div className="mb-3">
  <label className="form-label">Subtitles</label>
  <select
    disabled={rented}
    className="form-select bg-dark text-white border-light"
    value={subtitles}
    onChange={(e) => setSubtitles(e.target.value)}
  >
    {Object.keys(prices["subtitles"]).map((key) => (
      <option key={key} value={key}>
        {key.charAt(0).toUpperCase() + key.slice(1)} (
        {prices["subtitles"][key]} zł)
      </option>
    ))}
  </select>
</div>
<div className="mb-3">
  <label className="form-label">Rent for</label>
  <select
    disabled={rented}
    className="form-select bg-dark text-white border-light"
    value={rentalTime}
    onChange={(e) => setRentalTime(e.target.value)}
  >
    {Object.entries(prices["rental-time"]).map(([key, val]) => (
      <option key={key} value={key}>
```

```
              {val.name} ({val.price} zł)
            </option>
          ))}
        </select>
      </div>
    </>
  );
};

export default RentalOptions;
import prices from "../../assets/priceData/prices.json";

const RentalStatus = ({ rented, timeLeft, startRental, rentalTime }) => {
  const formatTime = (seconds) => {
    const d = Math.floor(seconds / (3600 * 24));
    const h = Math.floor((seconds % (3600 * 24)) / 3600);
    const m = Math.floor((seconds % 3600) / 60);
    const s = seconds % 60;
    return `${d}d ${h}h ${m}m ${s}s`;
  };

  return (
    <div style={{ height: "30px" }}>
      {rented ? (
        <p className="card-text text-success">
          Time left: <strong>{formatTime(timeLeft)}</strong>
        </p>
      ) : (
        <button
          className="btn btn-outline-light w-100"
          onClick={startRental}
        >
          Rent for {prices["rental-time"][rentalTime]?.name}
        </button>
      )}
    </div>
  );
};

export default RentalStatus;
import { useContext } from "react";
import NavigationBar from "../navbar/Navbar";
import { UserContext } from "../contex/UserProvider";
import { Link, Navigate } from "react-router-dom";
import { Alert, Col, Container, Row } from "react-bootstrap";
import "../../Styles/MovieSearch.css";
import WatchlistMovie from "./WatchlistMovie";

const watchListLibrary = () => {
  const { user } = useContext(UserContext);
  const UserNotLoggedIn = () => {};
  return (
    <>
      <NavigationBar />
      <Container fluid className="body">
        {
```

```
      <Row>
        {user ? (
          user.watchlist.map((e, i) => (
            <Col
              key={i}
              className="justify-content-center d-flex mb-5"
              xs="6"
            >
              <WatchlistMovie movie={e} />
            </Col>
          ))
        ) : (
          <Col>
            <Alert variant="danger">
              <Link
                to={"/signin"}
                className="alert-link sign-in-link"
              >
                Sign in
              </Link>{" "}
              to add to watchlist and rent movies
            </Alert>
          </Col>
        )}
      </Row>
    }
  </Container>
</>
```

```
  );
};
export default watchListLibrary;
import React, { useState, useEffect, useContext } from "react";
import { UserContext } from "../contex/UserProvider";
import { options } from "../../config/config";
import prices from "../../assets/priceData/prices.json";
import imageUnavailable from "../../assets/NoImageAvailable.png";
import MovieCardLayout from "./MovieCardLayout";

const WatchlistMovie = ({ movie }) => {
  const { user, addToRentedMovies } = useContext(UserContext);
  const apiKey = options.apikey;

  const [movieDetails, setMovieDetails] = useState(null);
  const [videoQuality, setVideoQuality] = useState("720p");
  const [audioQuality, setAudioQuality] = useState("mono");
  const [subtitles, setSubtitles] = useState("none");
  const [rentalTime, setRentalTime] = useState("1day");
  const [rented, setRented] = useState(false);
  const [timeLeft, setTimeLeft] = useState(0);

  const fetchMovieDetails = async () => {
    try {
      const response = await fetch(
        `http://www.omdbapi.com/?apikey=${apiKey}&i=${movie.imdbID}`
      );
      const data = await response.json();
    setMovieDetails(data);
  } catch (error) {
    console.error(error);
  }
};

const startRental = () => {
  const durationInSeconds =
    24 * 60 * 60 * prices["rental-time"][rentalTime]?.durationInDays;
  const endOfRent = new Date(Date.now() + durationInSeconds * 1000);
  const options = {
    video: videoQuality,
    audio: audioQuality,
    subtitles: subtitles,
    rentalTime: rentalTime,
  };
  setRented(true);
  setTimeLeft(durationInSeconds);
  addToRentedMovies({ imdbID: movie.imdbID, endOfRent, options });
};

const checkIfAlreadyRented = () => {
  const rentedMovie = user.rentedMovies.find(
    (m) => m.imdbID === movie.imdbID
  );
  if (rentedMovie) {
    const now = new Date();
    const end = new Date(rentedMovie.endOfRent);
    if (end > now) {
      setTimeLeft(Math.floor((end - now) / 1000));
      setVideoQuality(rentedMovie.options.video);
      setAudioQuality(rentedMovie.options.audio);
      setSubtitles(rentedMovie.options.subtitles);
      setRentalTime(rentedMovie.options.rentalTime);
      return true;
    }
  }
  return false;
};

useEffect(() => {
  fetchMovieDetails();
}, []);

useEffect(() => {
  setRented(checkIfAlreadyRented());
}, [user.rentedMovies, movie.imdbID]);

useEffect(() => {
  if (!rented || timeLeft <= 0) return;

  const timer = setInterval(() => {
    setTimeLeft((prev) => {
      if (prev <= 1) {
        clearInterval(timer);
        setRented(false);
        return 0;
      }
      return prev - 1;
    });
  }, 1000);

  return () => clearInterval(timer);
}, [rented, timeLeft]);

const getTotalPrice = () =>
  (
    prices["video-quality"][videoQuality]?.price +
    prices["audio"][audioQuality]?.price +
    (prices["subtitles"][subtitles] || 0) +
    prices["rental-time"][rentalTime]?.price
  ).toFixed(2);

return (
  <MovieCardLayout
    movie={movie}
    movieDetails={movieDetails}
    rented={rented}
    timeLeft={timeLeft}
    videoQuality={videoQuality}
    setVideoQuality={setVideoQuality}
    audioQuality={audioQuality}
    setAudioQuality={setAudioQuality}
    subtitles={subtitles}
```

```
      setSubtitles={setSubtitles}
      rentalTime={rentalTime}
      setRentalTime={setRentalTime}
      getTotalPrice={getTotalPrice}
      startRental={startRental}
      imageUnavailable={imageUnavailable}
    />
  );
};

export default WatchlistMovie;
import { Button, Col, Container, Row } from "react-bootstrap";

import "../../Styles/WelcomePage.css";
import { Link } from "react-router-dom";
import Logo from "../template/Logo";
import { useContext, useRef } from "react";
import { useInView } from "framer-motion";
import VideoBackground from "../template/VideoBackground";
import { UserContext } from "../contex/UserProvider";

const WelcomePage = () => {
  const { user, setUser } = useContext(UserContext);
  const logoRef = useRef(null);
  const isLogoInView = useInView(logoRef);
  const buttonsRef = useRef(null);
  const areButtonsInView = useInView(buttonsRef);
  return (
<VideoBackground>
  <Container
    fluid
    style={{ position: "fixed", width: "100%", height: "100%" }}
  >
    <Row style={{ height: "100%" }}>
      <Col className="menuWelcomePage d-flex align-items-center">
        <div className="container-fluid-xs container-lg">
          <Row>
            <Col
              xs={12}
              className="d-flex"
              style={{ cursor: "default" }}
            >
              <div
                ref={logoRef}
                style={{
                  opacity: isLogoInView ? 1 : 0,
                  transition: "2s",
                }}
              >
                <Logo size={90} />
              </div>
            </Col>
            <Col className="d-flex align-items-center justify-content-start">
              <div
                ref={buttonsRef}
                style={{
```

```
      transform: areButtonsInView
        ? "translateY(0)"
        : "translateY(100px)",
      opacity: areButtonsInView ? 1 : 0,
      transition: "1s",
    }}
  >
    <Link to="/movies">
      <Button className=" rounded-5">
        Go Search a Movie
      </Button>
    </Link>
    {user == null ? (
      <Link to="/signin">
        <Button className="rounded-5 signInButton">
          Sign in
        </Button>
      </Link>
    ) : (
      <Button
        className="rounded-5 signOutButton"
        onClick={() => setUser(null)}
      >
        Sign out
      </Button>
    )}
  </div>
</Col>
```

```
              </Row>
            </div>
          </Col>
        </Row>
      </Container>
    </VideoBackground>
  );
};

export default WelcomePage;
```

prices.json

```
{
  "video-quality": {
    "360p": { "name": "360p", "price": 2.5 },
    "720p": { "name": "720p", "price": 5 },
    "1080p": { "name": "1080p", "price": 6.25 },
    "4k": { "name": "4k", "price": 8 }
  },
  "audio": {
    "mono": { "name": "Mono", "price": 0 },
    "stereo": { "name": "Stereo", "price": 0.5 },
    "5_1-surround": { "name": "5.1 Surround", "price": 1 }
  },
  "subtitles": {
    "none": 0,
    "english": 0,
    "polish": 0
  },
    "rental-time": {
        "1day": { "name": "1 day", "durationInDays": 1, "price": 0.5 },
        "7days": { "name": "7 days", "durationInDays": 7, "price": 2 },
        "1month": { "name": "1 month", "durationInDays": 30, "price": 5 }
    }
}
```

package.json

```
{
  "name": "movierentalproject",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint .",
    "preview": "vite preview --host"
  },
  "dependencies": {
    "@emotion/react": "^11.14.0",
    "@emotion/styled": "^11.14.0",
    "@fontsource/inter": "^5.2.5",
    "@mui/joy": "^5.0.0-beta.52",
    "@mui/material": "^7.0.2",
    "bootstrap": "^5.3.3",
    "framer-motion": "^12.6.3",
    "motion": "^12.6.3",
    "react": "^19.0.0",
    "react-bootstrap": "^2.10.9",
    "react-dom": "^19.0.0"
  },
  "devDependencies": {
    "@eslint/js": "^9.21.0",
    "@types/react": "^19.0.10",
    "@types/react-dom": "^19.0.4",
    "@vitejs/plugin-react": "^4.3.4",
    "eslint": "^9.21.0",
    "eslint-plugin-react-hooks": "^5.1.0",
    "eslint-plugin-react-refresh": "^0.4.19",
    "globals": "^15.15.0",
    "react-router-dom": "^7.5.0",
    "vite": "^6.2.0",
    "vite-plugin-svgr": "^4.3.0"
  }
}
```

vite.config.js

```
import { defineConfig } from "vite";
import react from "@vitejs/plugin-react";
import svgr from "vite-plugin-svgr";
// https://vite.dev/config/
export default defineConfig({
    plugins: [react(), svgr()],
    preview: {
        host: true,
        port: parseInt(process.env.PORT) || 4173,
        allowedHosts: ["reactmovierental.onrender.com"],
  },
});
```