

Project Code Files

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\globals.css

```
@import "tailwindcss";

:root {
  --background: #ffffff;
  --foreground: #171717;
}

@theme inline {
  --color-background: var(--background);
  --color-foreground: var(--foreground);
  --font-sans: var(--font-geist-sans);
  --font-mono: var(--font-geist-mono);
}

@media (prefers-color-scheme: dark) {
  :root {
    --background: #0a0a0a;
    --foreground: #ededed;
  }
}

body {
  background: var(--background);
  color: var(--foreground);
  font-family: Arial, Helvetica, sans-serif;
}
```

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\layout.js

```
import Navbar from "../Components/Navbar";
import { SearchProvider } from "../contexts/SearchContext";
import { MovieBookProvider } from "../contexts/MovieBookContext"; // Import the MovieBookProvider
import "../globals.css";

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body>
        <SearchProvider>
          <MovieBookProvider> { /* Wrap your app with MovieBookProvider */}
            <Navbar />
            {children}
          </MovieBookProvider>
        </SearchProvider>
      </body>
    </html>
  );
}
```

Project Code Files

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\page.js

```
"use client";

import { useState, useEffect } from "react";
import MovieCard from "../Components/MovieCard";

const API_KEY = "38c0eda6"; // Replace with your real API key

export default function Home() {
  const [movies, setMovies] = useState([]);
  const [search, setSearch] = useState('');

  useEffect(() => {
    async function fetchMovies() {
      try {
        const response = await fetch(
          `http://www.omdbapi.com/?apikey=${API_KEY}&s=spiderman&page=1`
        );
        const data = await response.json();

        if (data.Response === "True" && Array.isArray(data.Search)) {
          setMovies(data.Search);
        } else {
          console.error("Error fetching data:", data.Error);
        }
      } catch (error) {
        console.error("Error fetching movies:", error);
      }
    }

    fetchMovies();
  }, []);

  const filteredMovies = movies.filter((movie) =>
    movie.Title.toLowerCase().includes(search.toLowerCase())
  );

  return (
    <div className="container mx-auto p-4">
      <div className="max-w-7xl mx-auto">
        <div className="grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-5 xl:grid-cols-6 gap-4">
          {filteredMovies.length > 0 ? (
            filteredMovies.map((movie) => (
              <MovieCard key={movie.imdbID} movie={movie} />
            ))
          ) : (
            <p>No movies found matching your search.</p>
          )}
        </div>
      </div>
    </div>
  );
}
```

Project Code Files

```
};  
}
```

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\Components\MovieCard.js

```
"use client";  
  
import { useRouter } from "next/navigation"; // Correct for Next.js 13  
import Image from "next/image";  
import { useFavorites } from "../contexts/MovieBookContext";  
  
export default function MovieCard({ movie }) {  
  const { favorites, toggleFavorite } = useFavorites();  
  const router = useRouter(); // Initialize the useRouter hook  
  
  const handleFavoriteClick = () => {  
    toggleFavorite(movie); // Toggle favorite movie  
    // Adding a small delay to ensure toggleFavorite state is updated before navigating  
    setTimeout(() => {  
      router.push('/favourite'); // Navigate to the "favorites" page after toggling  
    }, 100);  
  };  
  
  return (  
    <div className="w-[220px] h-[380px] bg-gray-100 rounded-lg shadow-lg p-4 flex flex-col space-y-4 overflow-hidden">  
      <div className="relative w-full h-[200px]">  
        <Image  
          src={movie.Poster !== "N/A" ? movie.Poster : "/fallback-poster.jpg"} // Check if poster exists  
          alt={movie.Title}  
          width={500}  
          height={750}  
          style={{ objectFit: "cover" }} // Add objectFit through inline style for Next.js  
          className="rounded-md"  
        />  
      </div>  
      <div className="flex flex-col space-y-2">  
        <p className="font-bold text-lg truncate">{movie.Title}</p>  
        <p className="text-sm text-gray-600 line-clamp-2">{movie.Plot}</p>  
        <button  
          className={`p-2 w-full rounded ${favorites.some(fav => fav.imdbID === movie.imdbID) ? "bg-red-500" :  
"bg-blue-500"} text-white`}  
          onClick={handleFavoriteClick}  
        >  
          {favorites.some(fav => fav.imdbID === movie.imdbID) ? "Remove" : "Favorite"}  
        </button>  
      </div>  
    </div>  
  );  
}
```

Project Code Files

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\Components\Navbar.js

```
"use client";
import React, { useState } from "react";
import Link from "next/link";
import { useSearch } from "../contexts/SearchContext";

export default function Navbar() {
  const { setSearch } = useSearch();
  const [input, setInput] = useState('');

  const handleSearchChange = (e) => {
    setInput(e.target.value);
    setSearch(e.target.value);
  };

  return (
    <nav className="flex justify-between items-center p-4 bg-gray-800 text-white shadow-md">
      

      { /* Search Bar */ }
      <input
        className="p-2 rounded bg-gray-700 text-white border border-gray-500 focus:ring-2 focus:ring-red-400"
        type="text"
        value={input}
        onChange={handleSearchChange}
        placeholder="Search for Movies..."
      />

      { /* Navigation Links */ }
      <div className="flex gap-4">
        <Link href="/" className="px-4 py-2 bg-red-500 rounded hover:bg-red-600">Home</Link>
        <Link href="/favourites" className="px-4 py-2 bg-green-500 rounded hover:bg-green-600">Favorites</Link>
        <Link href="/login" className="px-4 py-2 bg-yellow-400 rounded hover:bg-yellow-500">Login</Link>
      </div>
    </nav>
  );
}
```

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\contexts\FavouritesContext.js

```
"use client";
import { createContext, useContext, useState } from "react";

const FavoritesContext = createContext();

export function useFavorites() {
  const context = useContext(FavoritesContext);
  if (!context) {
    throw new Error("useFavorites must be used within a FavoritesProvider");
  }
}
```

Project Code Files

```
    return context;
  }

export function FavoritesProvider({ children }) {
  const [favorites, setFavorites] = useState([]);

  const addFavorite = (movie) => {
    setFavorites((prevFavorites) => [...prevFavorites, movie]);
  };

  const removeFavorite = (movieID) => {
    setFavorites((prevFavorites) =>
      prevFavorites.filter((movie) => movie.imdbID !== movieID)
    );
  };

  return (
    <FavoritesContext.Provider value={{ favorites, addFavorite, removeFavorite }}>
      {children}
    </FavoritesContext.Provider>
  );
}
```

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\contexts\MovieBookContext.js

```
"use client";
import { createContext, useContext, useState } from 'react';

const MovieBookContext = createContext();

export function useFavorites() {
  const context = useContext(MovieBookContext);
  if (!context) {
    throw new Error("useFavorites must be used within a MovieBookProvider");
  }
  return context;
}

export function MovieBookProvider({ children }) {
  const [favorites, setFavorites] = useState([]);

  const toggleFavorite = (movie) => {
    setFavorites((prevFavorites) => {
      if (prevFavorites.some((fav) => fav.id === movie.id)) {
        return prevFavorites.filter((fav) => fav.id !== movie.id);
      } else {
        return [...prevFavorites, movie];
      }
    });
  };
}
```

Project Code Files

```
return (  
  <MovieBookContext.Provider value={{ favorites, toggleFavorite }}>  
    {children}  
  </MovieBookContext.Provider>  
);  
}
```

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\contexts\SearchContext.js

```
"use client";  
import { createContext, useContext, useState } from "react";  
  
// Create the Search context  
const SearchContext = createContext();  
  
export function useSearch() {  
  const context = useContext(SearchContext);  
  if (!context) {  
    throw new Error("useSearch must be used within a SearchProvider");  
  }  
  return context;  
}  
  
// Search provider to wrap the app  
export function SearchProvider({ children }) {  
  const [search, setSearch] = useState(""); // State to store search query  
  
  return (  
    <SearchContext.Provider value={{ search, setSearch }}>  
      {children}  
    </SearchContext.Provider>  
  );  
}
```

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\avourite\page.js

```
"use client";  
import { useFavorites } from "../contexts/MovieBookContext";  
import MovieCard from "../Components/MovieCard";  
import { useSearch } from "../contexts/SearchContext"; // Import the search context  
  
export default function Favorites() {  
  const { favorites } = useFavorites(); // Get the list of favorite movies  
  const { search } = useSearch(); // Access the global search term  
  
  // Filter favorites based on the global search term  
  const filteredFavorites = favorites.filter((movie) => {  
    // Check if movie title and description are defined before calling toLowerCase  
    const title = movie.title || ''; // Default to empty string if title is undefined  
    const desc = movie.desc || ''; // Default to empty string if description is undefined  
    const query = search || ''; // Default to empty string if search term is undefined
```

Project Code Files

```
return (
  title.toLowerCase().includes(query.toLowerCase()) ||
  desc.toLowerCase().includes(query.toLowerCase())
);
});

return (
  <div className="p-4">
    <h1 className="text-2xl font-bold mb-4">Favorites</h1>
    <div className="flex flex-wrap gap-4">
      {filteredFavorites.length > 0 ? (
        filteredFavorites.map((movie) => (
          // Ensure each movie has a unique key
          <MovieCard key={movie.id} movie={movie} />
        ))
      ) : (
        <p>No favorite movies found matching your search.</p>
      )}
    </div>
  </div>
);
}
```

File: C:\Users\rashm\OneDrive\Desktop\NextAssign\movie_booking\app\pages\index.js

```
import React from "react";
import Navbar from "../Components/Navbar";
import MovieCard from "../Components/MovieCard";

export async function getServerSideProps() {
  const response = await fetch("/api/Movies");
  const movies = await response.json();
  return { props: { movies } };
}

export default function Home({ movies }) {
  return (
    <div>
      <Navbar />
      <div className="grid grid-cols-2 md:grid-cols-4 gap-4 p-4">
        {movies.map((movie) => (
          <MovieCard key={movie.id} movie={movie} />
        ))}
      </div>
    </div>
  );
}
```