HKBK COLLEGE OF ENGINEERING

PROJECT TITLE: ANIME WISHLIST WEBSITE

TEAM NAME: NOOB CORDERS

KRUTHIK N   1HK22IS049

DEPT: INFORMATION SCIENCE AND ENGINEERING

DOMAIN: MERN STACK

MENTOR: QANITHULLAKHAN

About the Anime Wishlist Website

- The Anime Wishlist Website is a web-based application designed to help users track, organize, and manage a personalized list of anime they wish to watch. It allows users to add new anime titles, view their wishlist, update details like watch status or priority, and remove titles when desired. This ensures an organized and easily accessible way to manage anime preferences.

- The application uses a lightweight backend with JSON or a simple database for data storage, ensuring persistence of user data across sessions. It is designed to be intuitive, responsive, and user-friendly, allowing seamless usage across devices. The codebase is modular and structured to support scalability, maintainability, and reusability of components.

Purpose

The main objective of this project is to provide a simple yet functional anime wishlist management tool that can be accessed through a web interface. It also serves as a foundation for learning basic front-end and back-end integration, data handling, and modular web development using technologies like HTML, CSS, JavaScript Node.js.

Key Features

- **Add Anime** – Add a new anime title to the wishlist along with optional details such as genre, release year, priority, and notes.

- **View Wishlist** – Display all saved anime entries in a neat, organized format.

- **Search Anime** – Find anime by title with partial matching or filter based on genre or priority.

- **Update Entry** – Edit existing anime details such as watch status or notes without losing current data.

- **Delete Entry** – Remove anime titles that are no longer of interest or have already been watched.

- **Data Persistence** – All changes are saved automatically using JSON or database storage, ensuring data is retained across sessions.

Advantages

- Simple and lightweight
- No need for internet connection or database setup
- Modular structure for easy maintenance
- Can be extended full feature mobile or web-based app

Future Enhancements

- **graphical User Interface (GUI)** using frameworks like React, Tkinter, or Flask.
- Import/Export Contacts in CSV or Excel format
- Cloud Synchronization for multi-device access
- Contact Grouping (e.g., Family, Friends, Work)

Project Structure

>anime
>Backend
 └node modules
 └server.js
 └package.json

>Frotnend
 └index.html
 └app.js
 └style.css

WISHLIST.JS CODE

```js
import { useState, useEffect } from 'react';
import axios from 'axios';
import AnimeCard from '../components/AnimeCard';
import { useAuth } from '../context/AuthContext';

export default function Wishlist() {
  const [wishlist, setWishlist] = useState([]);
  const [loading, setLoading] = useState(true);
  const { currentUser } = useAuth();


  // Added states for search
  const [searchQuery, setSearchQuery] = useState('');
  const [searchResults, setSearchResults] = useState([]);
  const [searchError, setSearchError] = useState('');


  useEffect(() => {
    const fetchWishlist = async () => {
      if (!currentUser) return;
      try {
        const response = await axios.get(`http://localhost:5000/api/wishlist/${currentUser.uid}`);
        console.log('Fetched wishlist data:', response.data);
        const mappedWishlist = response.data.map(item => ({
          _id: item._id,
          mal_id: item.mal_id,
          title: item.title || item.name || 'Unknown Title',

const addToWishlist = async (anime) => {
  if (!currentUser) {
    setSearchError('Please login to add to wishlist');
    return;
  }
  try {
    const animeData = {
      mal_id: anime.mal_id,
      title: anime.title || 'Unknown Title',
      image_url: anime.image_url || '',
      episodes: anime.episodes !== undefined ? anime.episodes : 'N/A',
      score: anime.score !== undefined ? anime.score : 'N/A',
      status: anime.airing ? 'Airing' : 'Finished',
      userId: currentUser.uid,
    };
    console.log('Adding to wishlist:', animeData);
    const response = await axios.post('http://localhost:5000/api/wishlist', animeData);
    setWishlist([...wishlist, response.data]);
    setSearchError('');
  } catch (error) {
    console.error('Error adding to wishlist:', error);
    setSearchError('Failed to add anime to wishlist');
  }
};
```

```jsx
    return (
      <div className="text-center py--12">
        <p className="text-gray-500 text-lg">Please login to view your wish
      </div>
    );
  }

  if (loading) {
    return (
      <div className="text-center py--12">
        <p className="text-gray-500 text-lg">Loading your wishlist...</p>
      </div>
    );
  }

  return (
    <div className="wishlist-page">
      <h1 className="wishlist-header">Your Anime Wishlist</h1>
      <p className="wishlist-subheader">Search and add your favorite anime
      <div className="wishlist-search">
        <input
          type="text"
          placeholder="Search for anime..."
          className="wishlist-search-input"
          value={searchQuery}
          onChange={(e) => setSearchQuery(e.target.value)}
        />
      {searchResults.length > 0 && (
        <div className="wishlist-search-results">
          {searchResults.map((anime) => (
            <div key={anime.mal_id} className="wishlist-search-result-item">
              <AnimeCard anime={anime} />
              <button onClick={() => addToWishlist(anime)}>Add to Wishlist</button>
            </div>
          ))}
        </div>
      )}
      {wishlist.length > 0 ? (
        <div className="wishlist-anime-container">
          {wishlist.map((anime) => (
            <AnimeCard
              key={anime._id}
              anime={anime}
              isInWishlist
              onRemoveFromWishlist={() => removeFromWishlist(anime._id)}
            />
          ))}
        </div>
      ) : (
        <div className="wishlist-empty">
          <p>Your wishlist is empty</p>
        </div>
      )}
    </div>
  );
```

App.js

```javascript
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './pages/Home';
import Wishlist from './pages/Wishlist';
import Navbar from './components/Navbar';

function App() {
  return (
    <Router>
      <Navbar />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/wishlist" element={<Wishlist />} />
      </Routes>
    </Router>
  );
}

export default App;
```

INDEX.JS

```javascript
// Either remove the reportWebVitals import or use it
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import { AuthProvider } from './context/AuthContext';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <AuthProvider>
      <App />
    </AuthProvider>
  </React.StrictMode>
);
```

Server.js

```javascript
require('dotenv').config();
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');        import axios
const axios = require('axios');
const rateLimit = require('express-rate-limit');
const helmet = require('helmet');
const app = express();

// Security middleware
app.use(helmet());

app.use(cors({
  origin: '*',
  credentials: true
}));


// Rate limiting
const limiter = rateLimit({
  windowMs: 15 * 60 * 1000, // 15 minutes
  max: 100, // limit each IP to 100 requests per windowMs
  message: 'Too many requests from this IP, please try again later.'
});
app.use('/api/', limiter);

// Body parser middleware
app.use(express.json({ limit: '10mb' }));
app.use(express.urlencoded({ extended: true, limit: '10mb' }));
```

```javascript
const limiter = rateLimit({
  windowMs: 15 * 60 * 1000, // 15 minutes
  max: 100, // limit each IP to 100 requests per windowMs
  message: 'Too many requests from this IP, please try again later.'
});
app.use('/api/', limiter);

// Body parser middleware
app.use(express.json({ limit: '10mb' }));
app.use(express.urlencoded({ extended: true, limit: '10mb' }));

// MongoDB Connection
mongoose.connect(process.env.MONGODB_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
.then(() => console.log('Connected to MongoDB'))
.catch(err => console.error('MongoDB connection error:', err));

// Anime Schema
const animeSchema = new mongoose.Schema({
  mal_id: Number,
  title: String,
  image_url: String,
  episodes: Number,
  score: Number,
  status: String,
  userId: String,
});
```

```javascript
// Map the response to match frontend expected fields
const results = response.data.data.map(anime => ({
  mal_id: anime.mal_id,
  title: anime.title,
  image_url: anime.images.jpg.image_url,
  episodes: anime.episodes || 0,
  score: anime.score || 0,
  airing: anime.airing || false,
}));

res.json(results);
} catch (error) {
console.error('Jikan API error:', error.response?.data || error.message);

if (error.response?.status === 429) {
  res.status(429).json({ error: 'Too many requests to anime database. Please try again later.' });
} else if (error.response?.status === 404) {
  res.status(404).json({ error: 'No anime found for your search' });
} else {
  res.status(500).json({ error: 'Failed to fetch anime data. Please try again.' });
}
}
```
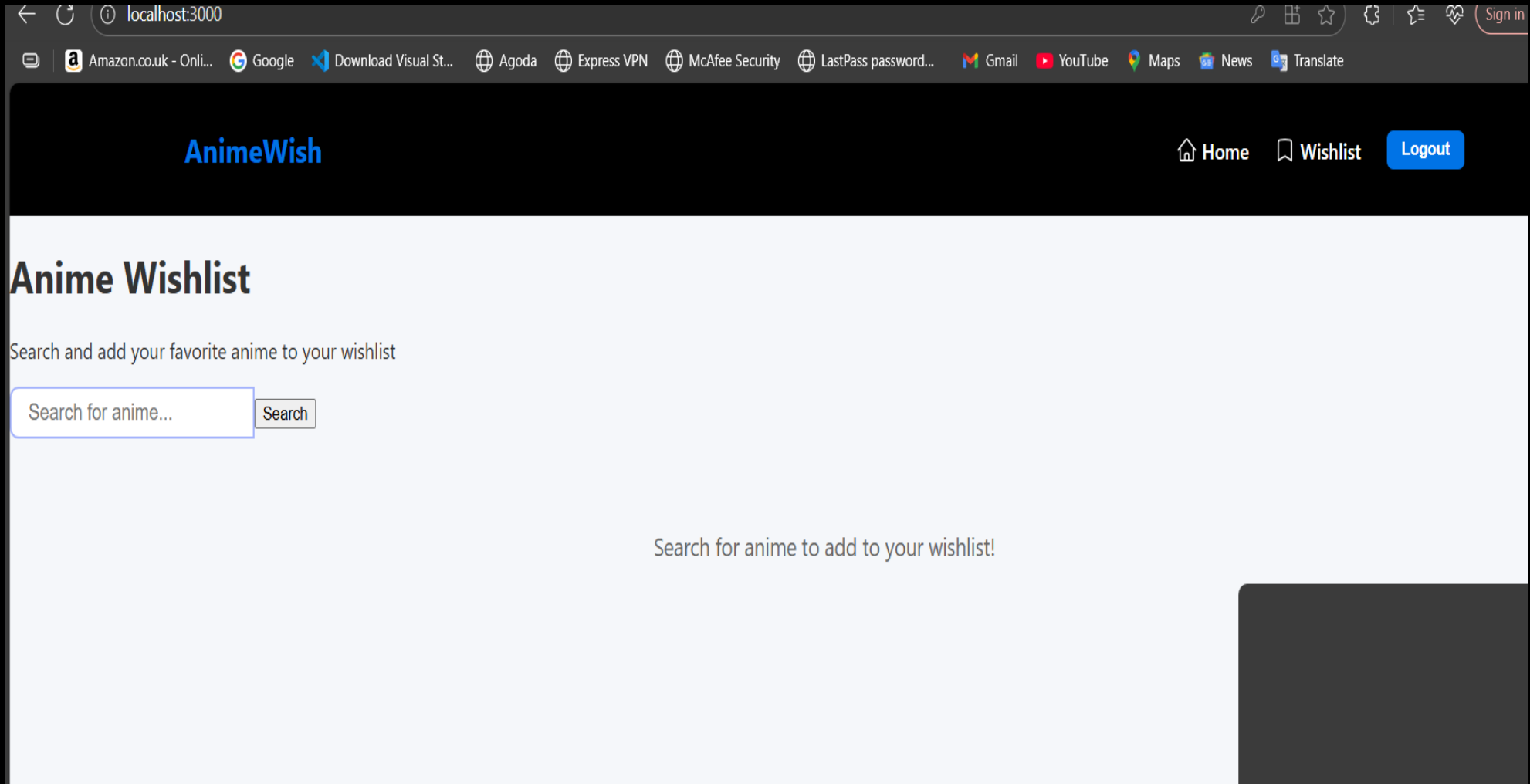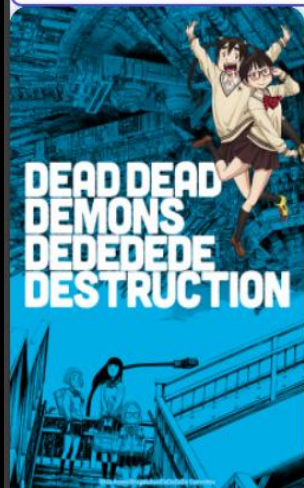
Anime Wishlist

Search and add your favorite anime to your wishlist

demons layer | Search

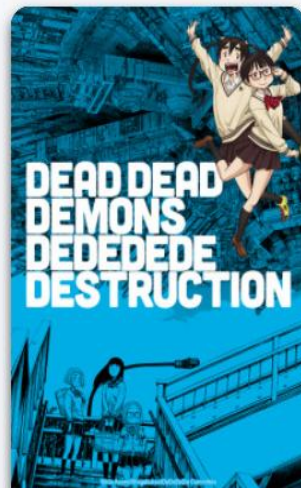**Dead Dead Demons Dededede Destruction (ONA)**

Episodes: 17 | Score: 7.92

Add to Wishlist

Dead Dead Demons Dededede Destruction

Episodes: 2 | Score: 7.54

Add to Wishlist

Dead Dead Demons Dededede Destruction (ONA) Episode 0

Episodes: 1 | Score: 7.23

Add to Wishlist

Dead Dead Demons Dededede Destruction 3DCG Animation

Episodes: 2 | Score: 6.08

Add to Wishlist

Toukiden

Episodes: 1 | Score: 5.7

Add to Wishlist

Yu☆Gi☆Oh! 5D's: Shinkasuru Kettou! Stardust vs. Red Demon's

Episodes: 1 | Score: 6.94

Add to Wishlist

Naedoko Demon's Ground

Episodes: 1 | Score: 5.43

Add to Wishlist

AnimeWish

Your Anime Wishlist

Search and add your favorite anime to your wishlist

AnimeWish

Login

Email

Password

Login

THANK YOU