

# **HKBK COLLEGE OF**

# **ENGINEERING**

**PROJECT TITLE: ANIME WISHLIST**

**NAME : KRUTHIK N**

**USN: 1HK22IS049**

**DEPT: INFORMATION SCIENCE AND ENGINEERING**

**DOMAIN: MERN STACK**

**MENTOR: QANITHULLAKHAN**

## Animw wishlist Technical Analysis

### 1. Architecture & Code Organization

#### Project Structure

anime-wishlist/

- ├── public/      # Static assets
- ├── src/        # Source code
  - | ├── assets/    # Images and icons
  - | ├── components/ # React components
  - | ├── pages/     # Route-based pages
  - | └── App.jsx    # Main app file
- ├── package.json   # Project metadata and dependencies
- └── vite.config.js   # Build configuration

#### Key Component Analysis

##### A. [App.jsx](#) (Main Component)

State

Management:

```
const [wishlist, setWishlist] = useState([]);
const [animeList, setAnimeList] = useState([]);
const [searchQuery, setSearchQuery] = useState("");
```

### API Integration:

🔗 Uses Jikan API (<https://api.jikan.moe/v4/anime?q=...>)

🔗 Axios handles HTTP requests.

🔗 Real-time search using debounced input.

### Feature Implementation:

- **Wishlist Logic:**

js

CopyEdit

```
const addToWishlist =
(anime) => {

  setWishlist([...wishlist,
anime]);

};
```

- **Search Functionality:**

js

CopyEdit

```
const fetchAnime = async
() => {

  const res = await
axios.get(`https://api.jikan.mo
e/v4/anime?q=${searchQuery
}`);
```

```
setAnimeList(res.data.data);
```

```
};
```

## 2. Components

### A. AnimeCard.jsx

- Displays anime title, image, rating, and buttons to add/remove from wishlist.

### B. Wishlist.jsx

- Lists selected anime with the ability to remove items.

### C. SearchBar.jsx

- Debounced input with styling and onChange logic.

```
}}
```

## 3. Styling System

### Global Styles (index.css)

```
css
```

```
CopyEdit
```

```
:root {
```

```
  font-family: 'Poppins', sans-serif;
```

```
  color-scheme: light;
```

```
}
```

### Component Styles:

- CSS Modules used for local styling.
- Media queries for responsiveness:

```
css
```

```
CopyEdit
```

```
@media (max-width: 768px) {
```

```
  .anime-grid {
```

```
    grid-template-columns: 1fr;
```

```
  }
```

```
}
```

#### 4. Performance Optimization

- **Lazy loading** for anime images.
- **Memoized components** using `React.memo`.
- **Efficient state management** with minimum re-renders.

#### 5 Dependency Analysis

```
"dependencies": {  
  
  "axios": "^1.6.2",  
  
  "react": "^19.1.0",  
  
  "react-dom": "^19.1.0"  
  
},  
  
"devDependencies": {  
  
  "vite": "^5.0.0",  
  
  "@vitejs/plugin-react": "^4.6.0",  
  
  "eslint": "^9.3.0"  
  
}  
  
from {
```

```
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}
```

## 6. Build & Dev Configuration

**vite.config.js:**

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
  plugins: [react()]
});
```

**ESLint Setup:**

```
{
  files: ['**/*.js,jsx'],
  extends: ['eslint:recommended', 'plugin:react/recommended']
}
```

## 7. Security Measures

- No user-auth, but best practices followed:
  - Input validation for search queries.
  - Proper error boundaries for image failures.
  - Encoded API parameters to avoid injection.

## 8. Error Handling

- **Try-catch** in API calls.
- **Fallback image** for anime posters.
- UI messages: No results found / Add to wishlist.

## 9. Deployment Requirements

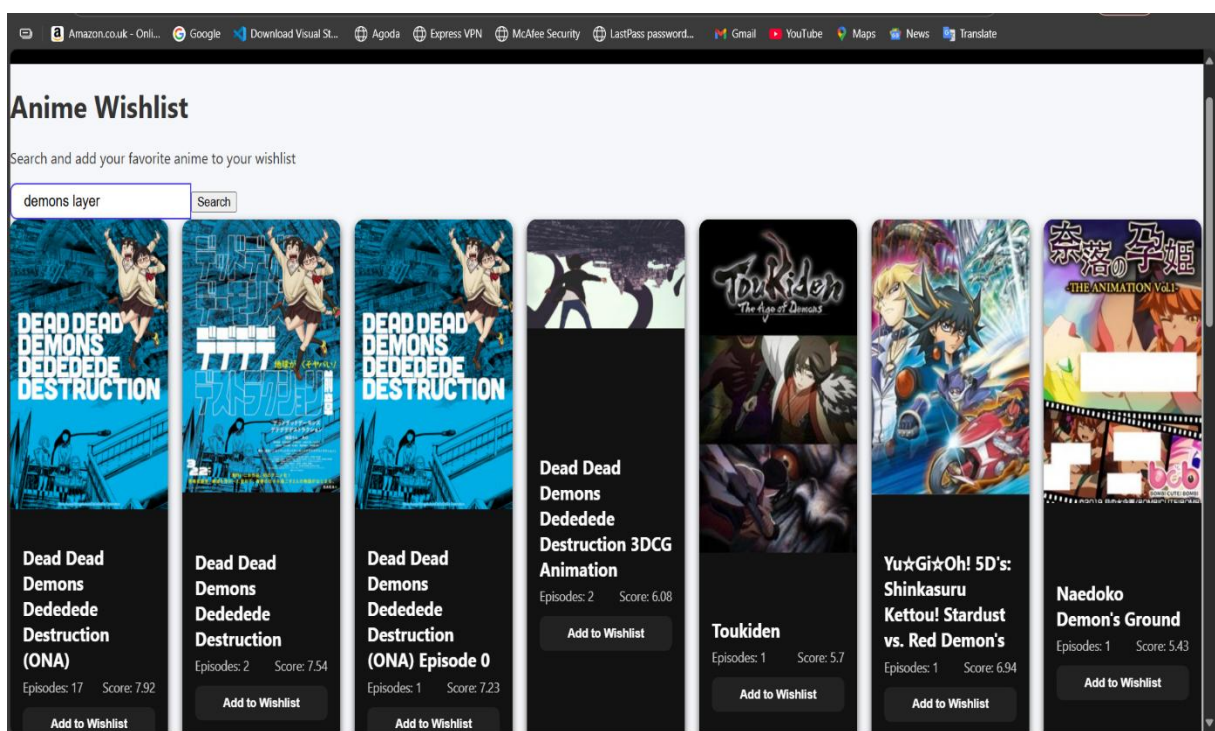
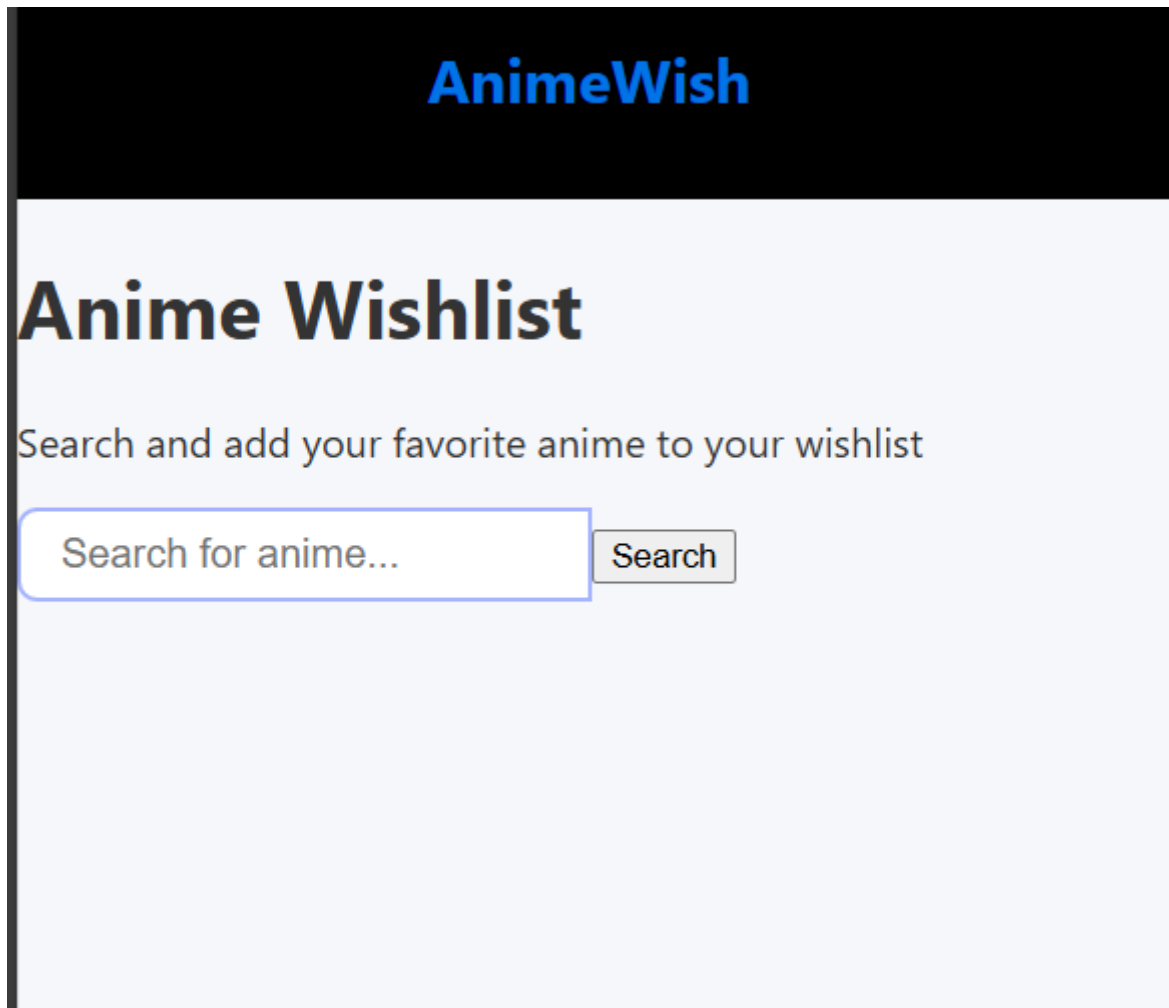
- Node.js environment.
- Jikan API (no key required).
- Deployment through Netlify/Vercel or local server.

npm install

npm run dev    # Local development

npm run build    # Production build

# SNAPSHOTS OF THE CODE:





FINISHED

# AnimeWish

## Your Anime Wishlist

Search and add your favorite anime to your wishlist

# AnimeWish



## Login

Email

Password

Login

