

EmployWise - User Management App

Overview

EmployWise ek **React application** hai jo **Reqres API** ka use karke basic **user management functions** perform karta hai. Is app me **authentication, user listing, pagination, edit, aur delete functionalities** available hain.

Installation & Setup

1.Clone Repository

sh

CopyEdit

```
git clone https://github.com/YOUR_GITHUB_USERNAME/employwise-app.git
```

```
cd employwise-app
```

2.Install Dependencies

sh

CopyEdit

```
npm install
```

3.Run the Project

sh

CopyEdit

```
npm start
```

◆ Isse **development server** start hoga jo <http://localhost:3000/> par accessible hoga.

Features Implemented

✓ Level 1: Authentication Screen

- **Login Page** jisme user **email aur password** daal kar login kar sakta hai.
- **Successful login** hone par **token** store hota hai aur **User List** page par navigate hota hai.
- API Endpoint: POST /api/login

✓ Level 2: User List & Pagination

- **Paginated list** of users using GET /api/users?page=1
- **First Name, Last Name, Avatar** show kiya gaya hai ek **table format** me.
- **Pagination or Lazy Loading** dono implement kiya ja sakta hai.

✓ Level 3: Edit & Delete Users

- **Edit Functionality:**
 - User list ke har user ke saath **Edit button** diya gaya hai.
 - Edit button click karne par **pre-filled form** open hota hai.
 - Update hone ke baad UI bina page refresh kiye update hota hai.
 - API Endpoint: PUT /api/users/{id}
- **Delete Functionality:**
 - User list me **Delete button** bhi diya gaya hai.
 - Delete hone par **UI se turant remove hota hai** bina refresh kiye.
 - API Endpoint: DELETE /api/users/{id}

📁 Project Structure

pgsql

CopyEdit

employwise-app/

|— src/

| |— components/

| | |— Login.js

| | |— UserList.js

| | |— EditUser.js

| |— App.js

| |— index.js

|— package.json

|— README.md

🛡️ Assumptions & Considerations

- 1.**Token storage:** Login hone ke baad token ko **localStorage** me save kiya gaya hai.
- 2.**Fake API (Reqres.in):** Ye API **backend changes save nahi karti**, isliye page refresh hone par data reset ho jata hai.
- 3.**Pagination & Performance:** Server-side pagination API support karta hai, but **lazy loading** bhi implement kiya ja sakta hai.
- 4.**Error Handling:** API errors handle kiye gaye hain, jaise invalid login credentials ya API failure.
- 5.**React Router:** Login aur User List ke beech **navigation** ke liye React Router use kiya gaya hai.