User Management System

# Features

Manage Users Page:  
- Role Management: Admins can change the role of a user between Admin and User.  
- Status Management: Admins can change the status of a user between Active and Inactive.  
- Permissions: Admins can grant the following permissions to users:  
 - Read (Green Button)  
 - Write (Yellow Button)  
 - Delete (Red Button)  
- Changes are saved to the browser's localStorage and persist even when the page is refreshed.  
  
View Users Page:  
- Dynamic Data: The View Users page dynamically loads user data, including roles and statuses, from localStorage.  
- It displays user information for multiple users, and reflects any changes made on the Manage Users page.

# Installation and Usage

1. Clone or Download the Repository:  
 - Clone the repository or download the source code to your local machine.  
 git clone <repository\_url>  
  
2. Open the Files:  
 - Open the `index.html` in your browser to view the user interface.  
  
3. Browser Support:  
 - The project supports modern browsers (Chrome, Firefox, Safari, Edge).  
  
4. Working Demo:  
 - Open the Manage Users page and make changes (role, status, permissions).  
 - Then navigate to the View Users page to see the updated data.

# File Structure

/project-root  
 ├── manageusers.html # Page for managing users (roles, status, permissions)  
 ├── viewusers.html # Page for viewing user details (roles, status)  
 └── README.md # Project documentation

# File Descriptions

1. manageusers.html:  
 - This page allows admins to manage users' roles, statuses, and permissions. It uses a dropdown for roles and statuses and buttons for permissions.  
 - User data (changes made) are stored in localStorage.  
  
2. viewusers.html:  
 - Displays a list of users with their current roles and statuses.  
 - It retrieves data from localStorage and updates dynamically based on any changes made in the Manage Users page.

# Local Storage Usage

- Roles and Status: The roles and statuses for users are saved in localStorage with keys like:  
 - `John Doe-role` for the role of John Doe.  
 - `John Doe-status` for the status of John Doe.  
  
- Permissions: Permissions granted to each user are saved in localStorage with keys like:  
 - `John Doe-Read-permission`  
 - `John Doe-Write-permission`  
 - `John Doe-Delete-permission`  
  
Example of localStorage keys:  
  
localStorage.setItem('John Doe-role', 'Admin');  
localStorage.setItem('John Doe-status', 'Active');  
localStorage.setItem('John Doe-Read-permission', 'granted');

# Technologies Used

- HTML: Structure of the pages.  
- CSS: Styling and layout of the pages, including custom sidebar and table design.  
- Bootstrap: Used for styling buttons, dropdowns, and layout responsiveness.  
- JavaScript: Handling logic for updating user roles, statuses, permissions, and saving data to localStorage.

# Customization

You can easily customize this project to add more user fields, additional actions, or modify the UI styling.

# Contributing

1. Fork the repository  
2. Create a new branch  
3. Make your changes  
4. Submit a pull request

# License

This project is open-source and available under the MIT License.