Store Admin & Student Roles in Firebase

1. Go to **Firebase Firestore**.
2. Create a **users collection** and manually add admins & students.
3. Add a **role field** (`admin` or `student`).

Example User Data in Firestore

```
nginx
CopyEdit
users (Collection)
 ├── UserID123 (Document)
 │     ├── email: "admin@email.com"
 │     ├── role: "admin"
 ├── UserID456 (Document)
 │     ├── email: "student@email.com"
 │     ├── role: "student"
```

✅ How It Works (Navigation)

User Type	Login Process	Redirects To
Admin	Logs in with admin email	`admin.html`
Student	Logs in with student email	`home.html`
Unauthorized User	Tries to access `admin.html`	Redirected to `index.html`

🎯 Final File Structure

```
pgsql
CopyEdit
NCW_Event_Organizer/
│── index.html  (Login Page)
│── home.html  (Student Dashboard)
│── admin.html  (Admin Panel)
│── event-details.html  (Event Details)
│── register.html  (Event Registration)
│── styles/
│   ├── style.css  (Main Styles)
│── js/
│   ├── script.js  (Main JS)
│   ├── auth.js  (Login System)
│   ├── firebase-config.js  (Firebase Configuration)
│── README.md  (Documentation)
```

✅ Features Implemented

✔ **Login System** for Students & Admins
✔ **Redirect Admins to `admin.html` & Students to `home.html`**
✔ **Block Students from `admin.html`**
✔ **Prevent Unauthorized Users from Accessing the Website**

🔥 **Now your authentication system is fully functional!** 🚀 Let me know if you need further modifications.

Final File Structure

📁 **NCW_Event_Organizer/**
│── 📄 **index.html** *(Login Page)*
│── 📄 **home.html** *(Home Page with Event Calendar)*
│── 📄 **event-details.html** *(Event Details Page)*
│── 📄 **register.html** *(Event Registration Page)*
│── 📄 **admin.html** *(Admin Panel - Only for Admins)*
│── **styles/** *(CSS Files for UI Styling)*
│ ├── 📄 **style.css** *(Main UI Styling for Users)*
│ ├── 📄 **responsive.css** *(For Mobile Responsiveness - Optional)*
│ ├── 📄 **admin.css** *(Admin Panel Styling)*
│── **js/** *(JavaScript Files for Interactivity & Backend Integration)*
│ ├── 📄 **script.js** *(General JavaScript for User Pages)*
│ ├── 📄 **admin.js** *(JavaScript for Admin Panel - Firebase Integration)*
│ ├── 📄 **auth.js** *(Handles User & Admin Authentication - Firebase)*
│── **images/** *(All Media & Icons)*
│ ├── 📄 **background.jpg** *(Background Image)*
│ ├── 📄 **logo.png** *(Event Logo)*
│── **assets/** *(Other Assets - Icons, Fonts, etc.)*
│ ├── **icons/** *(Icons Used in UI)*
│── **firebase/** *(Firebase Configuration & Database Handling)*
│ ├── 📄 **firebase-config.js** *(Firebase Initialization File)*
│── 📄 **README.md** *(Project Documentation & Setup Guide)*
│── 📄 **package.json** *(If Using Node.js for Hosting - Optional)*

📌 Explanation of the Structure

✓ **Admin Panel (`admin.html`)** is separated for security
✓ **`styles/` folder** for all CSS styles
✓ **`js/` folder** for scripts, including user interactions & Firebase backend
✓ **`firebase/` folder** for Firebase database and authentication setup
✓ **Media (`images/` & `assets/`)** is structured properly
✓ **README.md** provides setup instructions

📌 How Everything Works

1️⃣ **Users & Admin Login**:

- `index.html` allows both users and admins to log in.
- `auth.js` manages authentication using Firebase.

2️⃣ **Navigation & Role-Based Access**:

- After login, **users go to `home.html`** (event calendar).
- **Admins go to `admin.html`**, which is protected via Firebase Authentication.

3️⃣ **Admin Panel Functionalities (`admin.js`)**:

- **Add/Edit/Delete Events** (Stored in Firebase Database).
- **View Registered Students**.
- **Upload Event Results & Winners**.

4️⃣ **Firebase (`firebase-config.js`)**:

- Connects the app to Firebase for database & authentication.