Authentication Module (Firebase REST API)

📌 Newly Added Features (After Previous README)

This update adds **Login functionality** using Firebase Authentication REST API, along with secure token handling for authenticated users.

🔗 What Was Added

🔐 1. Login with Email & Password

- Implemented **user login** using Firebase REST API:
- `accounts:signInWithPassword`
- Login works with existing Firebase users
- Incorrect credentials show:
- **"Authentication failed!"** error message

🧾 2. JWT Token Handling

- On successful **Signup or Login**, Firebase returns an `idToken`
- This token is a **JWT (JSON Web Token)** used for authentication

```
console.log(data.idToken);
```

💾 3. Token Storage (Client-side)

- The JWT token is stored in the browser using `localStorage`

```
localStorage.setItem('token', data.idToken);
localStorage.setItem('email', data.email);
```

📍 Location in browser:

```
DevTools → Application → Local Storage
```

🕐 4. Auth Mode Toggle

- Users can switch between:
- **Login**
- **Sign Up**
- UI updates dynamically using React state

⏳ 5. Loading & Error States

- Loader shown while request is in progress
- Error message displayed on authentication failure

🔑 Firebase Setup Used

- Firebase **Authentication → Email/Password** enabled
- Firebase **Web API Key** used directly (REST API)
- Firebase SDK **NOT required**

Why Store the Token?

The token is required for: - Making authenticated API requests - Protecting private routes - Keeping user logged in after refresh

🔍 Security Note

- `localStorage` is acceptable for learning & demos
- For production apps:
- Prefer **httpOnly cookies**
- Protect against XSS attacks

🗜️ Next Possible Enhancements

- Auto-login on page refresh
- Logout functionality
- Protected routes
- Token expiration handling
- Context-based auth state

🔗 This update completes **basic authentication flow** using Firebase REST API.