Account Linking & Duplicate Prevention Features

Overview

This document describes the improvements made to prevent duplicate accounts and ensure proper linking between form-based and OAuth accounts.

Features Implemented

1. Form-Based Signup Validation

File: `Backend/api/serializers.py`

The `UserSerializer` now includes two custom validators:

Username Validation

- Checks if username already exists (case-insensitive)
- Returns error: "This username is already taken. Please choose a different username."

Email Validation

- Checks if email already exists (case-insensitive)
- Detects if the email is registered via Google OAuth
- Returns specific error messages:
  - If email has Google account: "This email is already registered with Google. Please sign in using 'Continue with Google' instead."
  - If email has form account: "This email is already registered. Please use a different email or log in."

2. Enhanced Login Flow

File: `Backend/api/views.py`

The `CookieTokenObtainPairView` now:

- Checks if user exists but only has OAuth (no password)
- Returns helpful error: "This account was created with Google. Please sign in using 'Continue with Google' instead."
- Provides better error messages for failed login attempts

3. OAuth Account Linking

File: `Backend/glogin/adapter.py`

The `SocialAdapter` automatically:

- Links Google OAuth to existing form-based accounts with matching email
- Links Google OAuth to existing Google accounts
- Logs whether user has form-based or OAuth-only account

- Handles duplicate email scenarios gracefully

4. Frontend Error Display

Files:

- `frontend/src/components/Signup.jsx`
- `frontend/src/components/Login.jsx`

Both components now:

- Call backend APIs with proper error handling
- Display backend validation errors in the correct fields
- Show loading states during API calls
- Display general error messages prominently
- Fetch and include CSRF tokens automatically

User Experience Flows

Scenario 1: User signs up with form, then tries OAuth

1. User creates account: `user@example.com` with form
2. Later, user clicks "Continue with Google" using `user@example.com`
3. System automatically links Google account to existing user
4. User can now login either way

Scenario 2: User signs up with OAuth, then tries form

1. User clicks "Continue with Google" with `user@example.com`
2. Account created via OAuth (no password)
3. Later, user tries to create account with form using `user@example.com`
4. System shows error: "This email is already registered with Google. Please sign in using 'Continue with Google' instead."

Scenario 3: User tries to login with wrong method

1. User has OAuth account (no password)
2. User tries to login with form using email/password
3. System shows error: "This account was created with Google. Please sign in using 'Continue with Google' instead."

Scenario 4: Duplicate username attempt

1. User tries to signup with username already taken
2. System shows error: "This username is already taken. Please choose a different username."

API Endpoints

Registration: `POST /api/user/register/`

Request Body:

```
{
  "username": "string",
  "email": "string",
  "first_name": "string",
  "last_name": "string",
  "password": "string"
}
```

Success Response (201):

```
{
  "message": "User created successfully",
  "user": {
    "id": 1,
    "username": "string",
    "email": "string",
    "first_name": "string",
    "last_name": "string"
  }
}
```

Error Response (400):

```
{
  "errors": {
    "username": ["This username is already taken. Please choose a different
username."],
    "email": ["This email is already registered with Google. Please sign in using
'Continue with Google' instead."]
  }
}
```

Login: `POST /api/auth/login/`

Request Body:

```
{
  "username": "string",  // or email
  "password": "string"
}
```

Success Response (200):

```
{
  "detail": "login successful"
}
```

Error Response (400/401):

```
{
  "detail": "This account was created with Google. Please sign in using 'Continue with Google' instead."
}
```

Testing Checklist

- [ ] Form signup with new email works
- [ ] Form signup with existing email shows error
- [ ] Form signup with existing Google email shows specific error
- [ ] Form signup with existing username shows error
- [ ] OAuth login creates new account if email doesn't exist
- [ ] OAuth login links to existing form account with same email
- [ ] Form login works for form accounts
- [ ] Form login fails for OAuth-only accounts with helpful message
- [ ] Error messages display correctly in UI
- [ ] Loading states work during API calls

Logging

All account linking operations are logged with the `glogin` logger at DEBUG level in development:

- When social login already exists
- When checking for existing users by email
- When linking social account to existing user
- When user has form-based vs OAuth-only account

View logs in Django console when running `python manage.py runserver`.