YouTube Clone — Fullstack App

This is a YouTube clone built using Next.js 13+, Tailwind CSS, MongoDB, Prisma, and Cloudinary. Users can log in with Google, create a channel, upload videos, and interact with content — just like on YouTube.

Features

✅ Google Authentication (NextAuth.js)

✅ Upload & stream videos with Cloudinary

✅ Create, update, and view user channels

✅ Like, comment, view counts, and subscribe

✅ TailwindCSS for UI

✅ Fully responsive design

Tech Stack

Frontend: Next.js 13 App Router, Tailwind CSS

Backend: Prisma + MongoDB

Auth: NextAuth.js with Google Provider

Cloud Storage: Cloudinary (for video uploads)

Environment Variables

Create a .env.local file in the root and include:

MongoDB

DATABASE_URL=mongodb+srv://...

NextAuth

GOOGLE_CLIENT_ID=your_google_client_id
GOOGLE_CLIENT_SECRET=your_google_client_secret

```
NEXTAUTH_SECRET=your_random_secret
NEXTAUTH_URL=http://localhost:3000
```

Cloudinary

```
NEXT_PUBLIC_CLOUDINARY_CLOUD_NAME=your_cloud_name
CLOUDINARY_URL=cloudinary://API_KEY:API_SECRET@your_cloud_name
```

Ensure you also create an unsigned upload preset in Cloudinary

Project Structure

/app → App router pages
/components → Reusable UI and modal components
/context → React context providers (auth, sidebar, etc.)
/actions → Server-side logic (getCurrentUser, etc.)
/lib → Utility functions
/prisma → Prisma schema and DB client
/public → Static assets

Getting Started (Local Development)

1. Install dependencies:
   ```
   npm install
   ```
2. Generate Prisma client:
   ```
   npx prisma generate
   ```
3. Push Prisma schema to MongoDB:
   ```
   npx prisma db push
   ```
4. Run development server:
   ```
   npm run dev
   ```

Visit http://localhost:3000 to view the app.

If port 3000 is busy, Next.js will switch to 3001.

Optional: Seed Dummy Data
If you want to add sample users/channels/videos for development:

```
npx prisma db seed
```

Make sure prisma/seed.ts is set up properly.

Cloudinary Setup

Go to https://cloudinary.com/console

Create an unsigned upload preset called: my_unsigned_preset

Add the following to .env.local:

NEXT_PUBLIC_CLOUDINARY_CLOUD_NAME=your_cloud_name

In the MediaUpload.tsx component:

< CldUploadWidget uploadPreset="my_unsigned_preset" />

Google Auth Setup

1. Visit Google Cloud Console
2. Create OAuth 2.0 Credentials
3. Add Authorized redirect URI:
   http://localhost:3000/api/auth/callback/google
4. Paste client ID & secret into .env.local

Deploying to Vercel

1. Push your project to GitHub
2. Go to vercel.com
3. Import the repo
4. Set the same .env values in the Vercel dashboard
5. Deploy

Resources

1. Next.js Docs
2. NextAuth.js Docs
3. Tailwind CSS Docs
4. Cloudinary Docs
5. Prisma Docs

Contributing

Feel free to fork, clone, and make PRs. Suggestions are welcome.

License

MIT License — use freely, but give credit where due.