Social Media App — Full Documentation

🧩 Tech Stack

Layer	Technology
Frontend	Angular
Backend	Express.js (Node.js)
Database	MongoDB (Mongoose)
Storage	Cloudinary (image uploads)
Auth	JWT (JSON Web Token)
Notification	Nodemailer (Email)

🔥 Features

- ✅ User Register/Login with JWT
- ✅ Create posts with text + image (Cloudinary)
- ✅ Like and **unlike** posts
- ✅ Add and **delete comments** on posts
- ✅ Follow/Unfollow users
- ✅ Feed to show posts from followed users
- ✅ **Delete your own posts**
- ✅ **Search users** by name or email
- ✅ Email notification when someone you follow posts

🏗️ Project Structure

```
project-root/
├── backend/
│   ├── controllers/
│   ├── models/
│   ├── routes/
│   ├── middlewares/
│   ├── utils/
│   ├── app.js
│   └── server.js
├── frontend/ (Angular app)
```

📁 Backend Details

✅ Models

User

```
{
  username: String,
  email: String,
  password: String,
  followers: [UserId],
  following: [UserId],
  notifications: [String]
}
```

Post

js

CopyEdit

```
{
  author: UserId,
  content: String,
  imageUrl: String,
  likes: [UserId],
  comments: [{
    user: UserId,
    text: String,
    createdAt: Date
  }],
  createdAt: Date
}
```

🔒 Auth

- JWT issued on login.
- Sent in Authorization: Bearer <token> header.
- Middleware auth.middleware.js protects all secured endpoints.

☁️ Cloudinary Integration

- Upload images from Angular.
- Backend uploads to Cloudinary using cloudinary.v2.
- Image URL is saved in MongoDB with the post.

📬 Email Notifications

- Sent to followers when a user creates a post.
- Handled with nodemailer.

🔗 Backend API Endpoints

Method	Endpoint	Description
POST	/api/auth/register	Register a new user
POST	/api/auth/login	Login and receive JWT
GET	/api/posts/feed	Get posts from followed users
POST	/api/posts	Create a new post
PUT	/api/posts/:id/like	Like a post
PUT	/api/posts/:id/unlike	Unlike a post
DELETE	/api/posts/:id	Delete a post
POST	/api/posts/:id/comments	Add comment to post
DELETE	/api/posts/:postId/comments/:id	Delete a specific comment
POST	/api/users/:id/follow	Follow a user
POST	/api/users/:id/unfollow	Unfollow a user
GET	/api/users/search?q=keyword	Search users by name or email

🌐 Frontend (Angular)

🔐 Route Guards (auth.guard.ts)

- Protects routes like /feed, /create-post, /profile, etc.
- If not authenticated, redirects to /login.

🧩 Angular Components

pgsql
CopyEdit

```
app/
├── auth/
│   ├── login.component.ts
│   ├── register.component.ts
├── feed/
│   ├── feed.component.ts        → View posts from followed users
│   ├── post-card.component.ts   → Show each post with likes/comments
├── post/
│   ├── create-post.component.ts → Create new post with image
├── profile/
│   ├── profile.component.ts     → View user profile and follow/unfollow
├── search/
│   ├── search.component.ts      → Search and follow users
├── services/
│   ├── auth.service.ts
│   ├── post.service.ts
│   └── user.service.ts
├── guards/
│   └── auth.guard.ts
│
```

📦 Frontend Features Summary

Feature	Component
Register/Login	auth/
View Feed	feed/feed.component.ts
Like/Unlike Post	post-card.component.ts
Comment on Post	post-card.component.ts
Create Post	post/create-post.component.ts
Delete Post	post-card.component.ts (only owner)
Follow/Unfollow Users	profile.component.ts
View & Search Users	search.component.ts

⚙️ Environment Variables

Backend .env:

```
MONGO_URI=mongodb://localhost:27017/linkedin-clone
CLOUD_NAME=dyt2h3pgq
CLOUD_API_KEY=387575578193138
CLOUD_API_SECRET=3m_ITFUhKqMuCYx4s0g3pi1AGxw
EMAIL_USER=your_email@gmail.com
EMAIL_PASS=your_email_password_or_app_password
```

🚀 Future Enhancements

- Real-time notifications
- Bookmark/save posts
- Infinite scroll
- Activity history
- Profile image support
- Full messaging system