Pixly Design Document

1. Introduction

Project Name: Pixly
Purpose: To develop a modern, vibrant social media platform for microblogging, where users can post short messages, images, and videos; interact with others; and participate in groups.

Goals: - Build a secure, user-friendly, and scalable platform - Ensure real-time interactions - Support rich content sharing

---

2. Background

Pixly is inspired by popular microblogging services like Twitter. It enhances traditional posting with visual content support and community features like group discussions and suggested user discovery. The platform is optimized for engagement and discoverability.

---

3. Requirements

Functional Requirements

- Email, Google, and GitHub-based registration
- Secure authentication via Clerk
- Core tweet features: create, like, retweet, comment, delete
- Main pages: Home, Search, Notifications, Tweet, Groups, Profile
- Sidebar navigation
- Group creation and invitations
- Suggested users and groups on the right sidebar
- Full responsive design

Non-Functional Requirements

- High scalability and availability
- Fast performance and responsiveness
- Secure data handling
- Intuitive and aesthetically pleasing interface

---

4. Architecture

Pixly follows a microservices architecture:

- **Authentication Service:** User login and registration
- **User Service:** Profile handling
- **Tweet Service:** Manage tweets and interactions
- **Notification Service:** Activity alerts
- **Search Service:** Search users and groups

- **Group Service:** Create and manage user groups
- **Frontend:** Responsive web and mobile UI

---

5. Detailed Design

Authentication

- POST `/auth/register`
- POST `/auth/login`

User Service

- Schema: `id, username, name, onboarded, image, bio, tweets, retweets, likedTweets, replies, groups`
- GET `/users/{id}`
- PUT `/users/{id}`

Tweet Service

- Schema: `id, text, author, retweetOf, group, createdAt, parentId, children, likes`
- Endpoints: create, like, comment, retweet, delete

Group Service

- Schema: `id, username, name, image, createdBy, tweets, members`
- Endpoints: GET `/groups/{id}`, POST `/groups`

Search & Notification

- GET `/search`
- GET `/notifications`

---

6. UI Overview

Pages

- **Home Feed**: User and group posts
- **Profile**: Info and tweets
- **Tweet Creation**: Compose modal/page
- **Notifications**: User alerts
- **Groups**: Explore and manage
- **Search**: Discover content

Interaction Flows

1. Register / Log in
2. Post / Like / Comment / Retweet / Delete a tweet
3. Navigate via sidebar
4. Create & invite to groups

5. Discover content via suggestions

---

7. Deployment

- **Dev:** Node.js & Next.js environment
- **Prod:** Deployed via Vercel

---

8. Security

- Authentication: Clerk
- Secure user data handling

---

9. Glossary

- **Tweet:** Short post
- **Like:** Approve post
- **Retweet:** Repost content
- **Comment:** Respond to post
- **Group:** Community of users

---

10. Technology Stack

- **Frontend:** Next.js, TypeScript, shadcn/ui
- **Backend:** Node.js, MongoDB
- **Authentication:** Clerk
- **Hosting:** Vercel

---

11. Style Guide Summary

- **Primary Font:** Poppins / Manrope
- **Primary Color:** `#007BFF` (blue)
- **Accent Color:** `#FFB74D` (orange)
- **Background:** `#0B0F1A` (dark)
- **Text:** `#F1F5F9` (light)
- **Secondary Text:** `#9CA3AF` (gray)