

# Product Requirements Document (PRD)

## Twitter-Like Social Media Application

### 1. Product Overview

This document outlines the requirements for building an end-to-end Twitter-like social media application. The platform enables users to create short posts, interact with other users, follow accounts, and engage in real-time conversations. The product is designed to be scalable, secure, and production-ready.

### 2. Goals & Objectives

- 1 Enable users to share short-form content in real time
- 2 Facilitate social interaction through likes, replies, reposts, and follows
- 3 Provide a scalable backend suitable for millions of users
- 4 Ensure data security, moderation, and high availability

### 3. Target Users

- 1 General users consuming and posting content
- 2 Content creators and influencers
- 3 Moderators and administrators
- 4 Developers integrating via APIs

### 4. Core Features

#### 4.1 Authentication & User Management

- 1 User registration and login
- 2 JWT-based authentication
- 3 Password reset and email verification
- 4 Profile creation and updates

#### 4.2 Posting (Tweets)

- 1 Create short text posts (tweets)
- 2 Attach images, videos, or links
- 3 Edit and delete posts
- 4 Threaded conversations

#### 4.3 Social Interactions

- 1 Like and unlike posts
- 2 Reply to posts
- 3 Repost (retweet)

- 4 Follow and unfollow users

## 4.4 Feed & Timeline

- 1 Personalized home timeline
- 2 User profile timelines
- 3 Trending topics
- 4 Chronological and algorithmic feeds

## 4.5 Search & Discovery

- 1 Search users and posts
- 2 Hashtags and topics
- 3 Suggested accounts

## 4.6 Notifications

- 1 Likes, replies, and mentions notifications
- 2 Follow notifications
- 3 Real-time delivery (WebSockets or push)

## 5. Non-Functional Requirements

- 1 Scalability to handle high traffic
- 2 High availability and fault tolerance
- 3 Secure data storage and transmission
- 4 Low-latency API responses

## 6. Technical Architecture

Backend: Flask (REST API), SQLAlchemy, PostgreSQL

Frontend: Web or Mobile client (React, Flutter, etc.)

Caching: Redis

Async Tasks: Celery + Redis/RabbitMQ

Storage: Object storage for media

Deployment: Docker, CI/CD, Cloud hosting

## 7. Security & Moderation

- 1 Rate limiting and abuse prevention
- 2 Content moderation and reporting
- 3 Role-based access control (admin/moderator)
- 4 Audit logging

## 8. Analytics & Admin Features

- 1 User management dashboard

- 2 Content moderation tools
- 3 Usage analytics and reporting

## 9. Future Enhancements

- 1 Direct messaging
- 2 Spaces / live audio
- 3 Monetization (ads, subscriptions)
- 4 AI-powered content recommendations