Software Requirements Document

# 1. Project Overview

Project Name: Sparity  
Purpose: To connect content creators with businesses for performance-based or fixed-rate promotions, enabling trustworthy, data-driven collaborations.  
Technology Stack: MERN (MongoDB, Express.js, React.js, Node.js)

# 2. User Roles

1. Content Creator  
2. Business (or Individual Client)  
3. Admin  
  
Each user can register under one role and add another later (role switching).

# 3. Core Features & Requirements

## 3.1 User Registration & Authentication

- Sign up with role selection (creator or business)  
- Secure login/logout with JWT-based authentication  
- Email verification (optional)  
- Role-switching capability after signup  
- Password encryption (bcrypt)

## 3.2 Profile Management

Business Profiles:  
Required: Full name, email, username, password, business name, industry, location, target audience, business type (online/physical), preferred content niches.  
Optional: Profile picture, bio, website link, budget range, expectations.  
  
Content Creator Profiles:  
Required: Full name, email, username, password, platforms used, follower count per platform, niches, pricing model (fixed/success-based), spoken languages, links to sample videos, estimated delivery time.  
Optional: Profile picture, bio, website link, past collaborations, editing tools used.

## 3.3 Creator Discovery & Search

- Search bar with filters: niche, platform, follower count, price range  
- Creator cards with compact stats and links

## 3.4 Messaging System

- One-on-one conversations between creators and businesses  
- Real-time or near-real-time chat via Poling   
- Offer and negotiation tracking in chat

## 3.5 Collaboration & Offer System

- Business sends a campaign proposal with offer type, campaign duration, target views, payment cap  
- Creator can accept or modify terms  
- Agreement is logged and stored  
- Admin view for contract oversight

## 3.6 Payment & Escrow

- Business pays upfront  
- Payment held in escrow via Stripe or PayPal  
- Admin verifies performance and releases funds  
- Supports success-based pricing and bonuses

## 3.7 Admin Dashboard

- Manage users and profiles  
- Approve/reject campaigns or payments  
- Moderate messages and disputes

## 3.8 Campaign & Performance Tracking

- Creators report video performance (screenshots/links)  
- Admin reviews before releasing payments  
- Support for time-bound success-based bonuses

## 3.9 Public Showcase

- Homepage with featured creators and successful campaigns  
- Visitor access to creator profiles  
- Testimonials or success stories

# 4. Non-Functional Requirements

- Security: HTTPS, JWT, password encryption, input validation  
- Scalability: Designed for horizontal scaling (MongoDB Atlas, Node clusters)  
- Performance: Optimized APIs, efficient queries, lazy loading  
- Responsive Design: Mobile-first UI using TailwindCSS + DaisyUI  
- Maintainability: Modular folder structure, reusable components, services layer

# 5. Tech Stack Breakdown

|  |  |
| --- | --- |
| Layer | Tech |
| Frontend | React.js + TailwindCSS + DaisyUI |
| State Mgmt | Zustand |
| Backend | Express.js + Node.js |
| Database | MongoDB (Mongoose ORM) |
| Auth | JWT, bcrypt |
| Realtime Chat | Poling |
| Payments | PayPal |
| Media Upload | Cloudinary / AWS S3 |
| Hosting | Render (Not in Deal / Optional) |