# AI Prompt for Digital Credentials Platform

You are an expert software architect and full-stack developer. Help me design and develop a scalable, secure, and customizable Digital Credentials Platform like Certifier.io and Credly’s Acclaim. This platform should issue, manage, verify, and analyze digital certificates and badges for educational institutions, corporates, NGOs, and professional bodies.  
  
Core Functional Requirements:  
  
1. Credential Issuance:  
 - Generate secure, verifiable certificates & badges (PDF/PNG + JSON-LD for optional blockchain)  
 - Credential metadata (issuer, issue date, expiry, description, recipient data)  
 - Support for single & bulk issuance via UI & API  
  
2. Verification:  
 - Public/private verification pages with unique credential URLs  
 - Optional blockchain-based verification for authenticity  
  
3. Customization:  
 - Drag-and-drop design studio for building certificate/badge templates  
 - Upload custom logos, signatures, background images  
 - Embed QR codes for verification links on the certificate  
  
4. Earners Directory:  
 - Directory of all credential holders (with privacy controls)  
 - Searchable by credential, name, or issuing body  
  
5. Analytics Dashboard:  
 - Track credential views, verification events, social shares, click-throughs  
 - Export analytics reports in CSV/Excel format  
  
6. User Roles:  
 - Super Admin: Full platform access  
 - Issuer Admin: Manage credential templates, issue credentials  
 - Verifier: Verify credentials  
 - Recipient/Earner: View, download, share credentials  
  
7. Integrations:  
 - Email (SMTP, SendGrid, AWS SES) for credential delivery  
 - WhatsApp API (Twilio) integration for direct delivery to users  
 - OAuth2-based SSO (Google, Microsoft, LinkedIn)  
 - Social sharing buttons (LinkedIn, Twitter, Facebook)  
 - API/Webhooks for integration with external LMS or HR systems  
  
8. Security & Compliance:  
 - GDPR-compliant user data handling  
 - Secure credential URLs (non-guessable UUIDs)  
 - HTTPS, JWT-based authentication, OAuth2 for integrations  
 - Role-based access control (RBAC)  
  
9. Scalability & Reliability:  
 - Microservices-compatible architecture  
 - Prepared for cloud-native deployment (Docker/Kubernetes ready)  
 - Support for bulk credential import (CSV) and issuance APIs  
 - Database backup & migration utilities  
  
10. Technology Stack Recommendation:  
 - Backend: Node.js + Express or Python (FastAPI)  
 - Frontend: React.js or Next.js  
 - Database: PostgreSQL for relational data  
 - Authentication: OAuth2, JWT  
 - Optional: Blockchain or decentralized identity (DID) integration  
  
Required Output from AI:  
1. Folder structure for a monorepo or modular architecture  
2. Example REST API endpoints: Issue credential, verify credential  
3. Database schema: Tables/entities for Users, Credentials, Templates, Analytics  
4. Frontend structure: Main components, routing, state management  
5. Suggestions for scaling this platform in production  
6. Any recommended libraries or open-source tools for:  
 - QR Code generation  
 - PDF/PNG rendering  
 - Analytics visualization (e.g., Chart.js, Recharts)  
  
Generate production-grade, modular code that can serve as an MVP foundation.