

Tools & Implementation Plan

2.4. Tools & Implementation Plan (condensed)

This section presents the core implementation subtopics required to build Gemellery. Each subtopic lists the primary tools or technologies and explains *why* it is essential for the project.

2.4.1. Data Collection & Sources

Reasoning: Accurate, verifiable product and certification data are the foundation of trust for Gemellery. Consolidating authoritative sources early reduces downstream verification effort and enables features such as NGJA cross-checks and the blockchain Gem Passport.

What to collect / sources

- **NGJA certification records** (structured certificate fields: certificate id, gem type, carat, cut, color, clarity, issue date, issuing officer).
- **Seller records & export licenses** (EDB registration, business info, KYC documents).
- **High-resolution images** of gems and finished jewellery (multiple angles, standard lighting).
- **Product metadata** from sellers (price, availability, origin, weight, description).
- **User interaction logs** (searches, wishlist events, design requests).
- **External references** for provenance where available (mining location data).

Format & retrieval

- Image files: JPEG/PNG + stored in object storage (S3).
- Metadata/certificates: JSON/CSV and PDF for certificate documents.
- Retrieval: secure API endpoints or staged CSV/Payload ingestion from verified sellers/NGJA.

2.4.2. Data Acquisition & Preprocessing

Reasoning: Raw certificates, metadata and images arrive in heterogeneous formats. A repeatable pipeline ensures consistent data quality for listing, search, AI input, and blockchain anchoring.

Key tools & steps

- **Ingestion:** Node.js or Python microservice to accept uploads (multipart/form for images, JSON/CSV for metadata).
- **Storage:** AWS S3 (or equivalent) for images; PostgreSQL for structured metadata.
- **Preprocessing:** Python (Pandas, Pillow/OpenCV) or Node.js (sharp) to:
 - Validate certificates (field checks, NGJA id patterns).
 - Normalize images (resize, strip EXIF, standard color profile).
 - Extract/parse PDFs (pdfminer or Tika) to harvest certificate fields.
 - Sanitize and map seller-provided fields to canonical schema.
- **Output:** Cleaned records (JSON) saved to DB; thumbnails and preview assets stored in S3.

2.4.3. Core System Architecture & Tech Stack

Reasoning: The platform must be secure, scalable and support modular addition of AI and blockchain features. Choose mature, well-supported technologies that fit your team and hosting constraints.

Suggested stack

- **Frontend:** React (TypeScript) — component-based UI for marketplace and design studio.
- **Backend / API:** Node.js (Express/NestJS) or Python (FastAPI) for REST/GraphQL endpoints.
- **Database:** PostgreSQL (relational storage for listings, users, orders).
- **Object Storage & CDN:** AWS S3 + CloudFront (image hosting & fast delivery).

- **Workflow Engine:** n8n for low-code orchestration of multi-step flows (webhook → prompt build → Gemini call → store results → notify).
- **Authentication:** OAuth2 / JWT + role-based access (buyer, seller, admin).
- **DevOps:** Docker for containerization; CI/CD via GitHub Actions / GitLab CI; Kubernetes or managed ECS/EKS for production orchestration.

2.4.4. AI & Image Generation Implementation

Reasoning: AI powers the B2C design studio; using a robust model and a workflow engine reduces development complexity and enables iterative refinement by users.

Components & tools

- **Model:** Google Gemini image model (Gemini-2.5-flash-image / Nano-Banana) or a trusted proxy/wrapper if required.
- **Prompt Orchestration:** Backend services + n8n nodes build structured prompts using gem metadata + user preferences.
- **n8n Workflow Example:** Webhook → Fetch gem metadata → Build prompt → HTTP Request → Poll/Fetch images → Post-process → Upload to S3 → DB update → Notify user.
- **Post-processing:** Image tooling (sharp, Pillow) for resizing, watermarking (disclosure), and adding gem metadata overlays.
- **Storage of AI outputs:** Store generated images and associated prompt/metadata for provenance and reproducibility.

2.4.5. Blockchain, Verification & Compliance

Reasoning: Trust and provenance are core selling points. A lightweight, auditable Gem Passport anchored on a blockchain provides immutable proof of certification and ownership transfer.

Design & tools

- **Blockchain choice:** Permissioned chain (Hyperledger Fabric) or public layer-2 anchoring (Ethereum/Polygon) depending on cost and audit requirements.

- **Data anchored:** Certificate hash, NGJA ID, seller ID, timestamp, and transaction reference. Store full metadata off-chain (DB + S3) and only anchor hashes on-chain.
- **Integration:** Backend service to generate hashes and create transactions; UI to retrieve and display Gem Passport (link/transaction id).
- **Legal / Compliance:** Store raw certificates and seller KYC securely; ensure disclosures about AI-generated imagery; follow export regulations via EDB integration.

2.4.6. Deployment, Monitoring & Maintenance

Reasoning: Production stability, observability and the ability to iterate are essential for adoption and scaling.

Operational tools & practices

- **CI/CD:** Automated pipelines for tests, builds and deployments (GitHub Actions/GitLab CI).
- **Container orchestration:** Kubernetes or managed services for scalability.
- **Monitoring & Logging:** Prometheus + Grafana for metrics; ELK/EFK stack (Elasticsearch/Fluentd/Kibana) or cloud logging for traces.
- **Security:** Routine vulnerability scans, secrets management (Vault / AWS Secrets Manager), HTTPS everywhere, WAF for public endpoints.
- **Backups & DR:** Regular DB backups, S3 versioning, and tested disaster recovery plan.
- **Analytics:** Mixpanel/Amplitude or self-hosted analytics for user behaviour, plus event logs for design interactions and purchases.

Implementation Roadmap (high level steps)

1. **MVP Prep:** Implement data ingestion, product listing CRUD, NGJA verification manual flow, and minimal frontend listings.

2. **MVP AI:** Add simple n8n workflow to accept user design requests → call Gemini wrapper → return 1 preview image; store prompt & image.
 3. **Trust Features:** Integrate blockchain anchoring for certificates and add visible NGJA / seller badges.
 4. **Export & Compliance:** Integrate EDB export handling and logistics partners for shipping workflows.
 5. **Scale & Optimization:** Add advanced AI features (multi-variant generation), fraud detection, and performance scaling.
-

Summary — Why this condensed structure helps

- **Fewer subtopics** improve readability while covering all essential implementation areas.
 - Each subtopic groups related technical tasks so development teams can run parallel workstreams (data, AI, backend, blockchain, ops).
 - The plan balances **trust (NGJA + blockchain)** and **innovation (AI design)**—the two core differentiators of Gemellery—while remaining achievable for an MVP.
-