**AI Packaging Design Platform – Technical Documentation**

**1. Objective**

Build a web-based platform that enables businesses to generate professional, print-ready packaging designs (e.g., soap boxes, labels) using AI. Users can create designs from text prompts, and (in future phases) from hand-drawn sketches or reference images.

**2. Problem Statement**

Small businesses and startups often face high costs and long turnaround times for custom packaging design. This platform democratizes access to high-quality, unique packaging designs by leveraging AI, reducing both cost and time.

**3. Feature List & Milestones**

**MVP (Milestone 1)**

- User authentication (email/password or social login)

- Onboarding flow

- Text prompt input for design generation

- AI-powered packaging design generation (mockup preview)

- Download print-ready design (high-res PNG, PDF, or SVG)

- User dashboard (view/download previous designs)

- Basic error handling and status feedback

**Milestone 2**

- Upload hand-drawn sketches or reference images

- Colour palette picker

- Advanced design customisation (fonts, layout tweaks)

- Template selection for common packaging types

- Job queue for long-running AI tasks (with progress/status)

- Enhanced user profile management

**Milestone 3**

- Team/collaborator support

- Payment integration for premium features

- Analytics/dashboard for users

- Admin panel for moderation and analytics

**4. User Workflow**

1. Onboarding: User signs up/logs in.
2. Input: The user enters a text prompt (and, in the future, uploads an image/sketch or selects a template).
3. Generation: The AI model generates a packaging design mockup.
4. Review: Users can preview the design, regenerate it, or tweak it.
5. Download: User downloads print-ready files.
6. Dashboard: The User can view/download previous designs.

**5. Tech Stack**

**Frontend**

* **Framework:** Next.js (React + TypeScript)
* **Styling:** Tailwind CSS
* **Deployment:** Vercel

**Backend/API**

* **Framework:** FastAPI (Python 3.11+)
* **Server:** Uvicorn (for development and production)
* **Deployment:** Railway (or Render/Heroku if you prefer)

**AI Model Integration**

* **Provider:** Hugging Face Inference API (or Replicate)

- No self-hosting or GPU management for MVP.

- Use their hosted text-to-image models (e.g., Stable Diffusion).

**Data & Storage**

* **File Storage:** Supabase Storage

- Handles user uploads and generated images.

* **Database:** Supabase PostgreSQL

- Stores user info, design metadata, etc.

**Authentication**

* **Provider:** Supabase Auth
* - Email/password and social login support.

**Monitoring & Logging**

* **Error Tracking:** Sentry (frontend and backend)

**CI/CD**

* **Frontend:** Vercel (auto-deploys from GitHub)
* **Backend:** Railway/Render/Heroku (auto-deploys from GitHub)

**DevOps**

* **Containerization:** Not required for MVP (use managed services)
* **Infrastructure as Code:** Not required for MVP

**6. Print-Ready Output**

* **Formats:** High-resolution PNG, PDF, and SVG (for vector support)
* **Resolution:** Minimum 300 DPI for print
* **Colour Profile:** CMYK support for print (if possible)
* **Templates:** Predefined die-lines for common packaging types (soap box, label, etc.)

**7. Security & Compliance**

* Use HTTPS for all endpoints.
* Store secrets securely (AWS Secrets Manager)
* Validate and sanitise all user inputs
* Scan uploaded files for malware
* Comply with GDPR for user data

**8. Development Plan**

**Phase 1: MVP**

* Set up monorepo (frontend, backend)
* Implement authentication (Firebase)
* Build onboarding and dashboard UI (Next.js + Tailwind)
* Integrate text-to-image AI model (hosted)
* Implement design generation and download
* Store user data and designs (PostgreSQL, S3)
* Set up CI/CD (GitHub Actions, Vercel, managed backend)
* Add Sentry for error monitoring

**Phase 2: Advanced Features**

* Add image/sketch upload and processing
* Integrate the colour palette picker and templates
* Implement job queue (Celery + Redis)
* Add advanced customisation options

**Phase 3: Scale & Monetise**

* Add team/collaborator features
* Integrate payments
* Build an admin panel and analytics
* Migrate backend to AWS ECS/EKS if needed

**9. References & Resources**

* **[Next.js Documentation](https://nextjs.org/docs)**
* **[FastAPI Documentation](https://fastapi.tiangolo.com/)**
* **[Hugging Face Inference API](https://huggingface.co/inference-api)**
* **[Replicate API](https://replicate.com/docs)**
* **[Firebase Auth](https://firebase.google.com/docs/auth)**
* **[AWS S3](https://aws.amazon.com/s3/)**
* **[Supabase](https://supabase.com/)**
* **[Sentry](https://sentry.io/welcome/)**

**10. Future Considerations**

* Support for more packaging types and custom die-lines
* Integration with print-on-demand services
* Mobile app version
* AI-powered design suggestions and A/B testing