B2B Fashion SaaS — Image-to-Video Catalogue + Trend Analyzer

Role

You are an expert full-stack application architect and UI/UX designer specializing in B2B fashion technology platforms. This document defines a complete product design, UI/UX, and technical specification for a SaaS product that enables fashion retailers and bulk buyers to upload dress images and receive AI-generated model showcases and videos, paired with real-time trend analytics.

Table of contents

---

1. Product overview

A premium B2B SaaS that shortens the gap between design → visualization → market validation. Retailers and bulk buyers upload dress images (single or batch) and in return receive: - Photorealistic 3D model stills (3–5 per design) wearing the dress in multiple poses. - 15-second product clips from different angles for listings or social content. - 30-second runway videos with multi-camera cuts and walk sequences for pitching/bulk purchasing. - A Trend Analyzer displaying trending dress types across age demographics (18–25, 26–35, 36–50, 50+), seasons, regions, and YoY growth.

Primary goals: - Reduce cost & time of physical photoshoots. - Provide actionable trend insights to inform buying and inventory. - Enable bulk catalogue generation and easy exports.

Target users: fashion retailers, bulk buyers, category managers, merchandisers, wholesalers.

---

2. Core user journeys

1. **Quick catalogue generation (single design):** Upload image → choose model templates → generate → review → export.
2. **Bulk catalogue creation:** Batch upload 10–200 images → queue jobs → monitor progress → bulk export as ZIP with metadata CSV.
3. **Trend exploration:** Open Trend Analyzer → filter by season/age/category → click an item → view regional & historical insights → pin trends to a watchlist.
4. **Purchase decision:** Use generated videos + trend scores to create a purchase shortlist and share with procurement.

---

3. Section designs

A. Image Upload & Catalogue Generator

Layout: left column upload controls + presets → center workspace (preview grid) → right metadata & actions panel.

Upload interface features: - Drag & drop zone with fallback file picker. - Supported formats: JPEG, PNG, WEBP, HEIC (auto-convert), up to 50 MB per file by default. - Batch support with thumbnails and per-file remove/edit. - Visual feedback: instant thumbnail, client-side validation, file size and aspect ratio hints. - Progress indicators for uploads (progress bar + percent) and resumable uploads (tus protocol or multipart resume).

Generation workflow: 1. Select images → choose model presets (gender mix, body diversity presets, skin tone variety, height presets) → choose output package: Stills (3–5), Clips (15s), Runway (30s) or All. 2. Optional: add style tags (casual, evening, cocktail, maxi, midi), desired backgrounds (studio, street, runway), and music style (for runway video). 3. Submit job → job queued and shown in activity panel with ETA and progress stages (uploaded → preprocessing → 3D mapping → rendering → postprocessing).

Output presentation: - Organized grid tabs: Pictures | 15s Clips | Runway Videos. Each item card contains thumbnail, duration (for videos), resolution, size, and three metadata badges: Estimated Market Appeal (0–100), Style Compatibility Tags, Body Diversity Indicators (percent coverage). - Click to preview in a modal with playback and multi-angle toggle. Provide download and share buttons; bulk select for zip export. - Metadata export: CSV/JSON per item with fields: original_filename, model_preset, tags, appeal_score, generated_assets_urls, timestamp.

Metadata & AI insights: - Each generated model has computed attributes: fit estimation, recommended size range, color accuracy score, suggested product description and SEO title, and predicted conversion uplift (based on trend model).

Accessibility & Mobile: - Mobile responsive: single column with collapsible metadata panel and bottom action sheet for generation presets.

B. Trend Analyzer Dashboard

Landing: large search bar to query dress categories + quick filters (season, age range, region).

Top visual elements: - Trending Today carousel (top 10 dress types by search volume) - Age demographic stacked bar chart (18–25, 26–35, 36–50, 50+) - Seasonal trend heatmap (month vs category) - Time-series trend velocity chart (30/60/90 day) with smoothing and derivative overlay (velocity) - YoY comparison small multiples for selected categories

Interactivity: - Click any trend tile to open a detail panel with: - Regional breakdown (choropleth or choropleth-lite for B2B regions), - Historical search volume & social mentions timeline, - Related search terms and rising queries, - Predicted short-term momentum and confidence score. - Filters: Age group, Season (Spring/Summer, Autumn/Winter), Category (maxi, midi, wrap, bodycon, A-line), Region, Date range.

Predictive insights: - A small machine learning module forecasts 14–30 day trend direction with probability bands. - Each insight card displays confidence and recommended action (e.g., “Increase order quantity by 15% for A-line dresses in 26–35 in APAC”).

Data sources: - Google Trends (for search volume), social listening APIs (public Twitter/X, Instagram hashtags via official partners), internal platform signals (how many generated catalogues contain a category), and third-party fashion datasets.

Export & Alerts: - Export charts / data CSV / PNG. - Alerting: save watchlists and configure email/Slack alerts when trend velocity for a category crosses a threshold.

C. Navigation & Layout

Global layout: Persistent left sidebar on desktop with app logo, primary nav entries: Catalogue Generator, Trend Analyzer, Watchlist, Exports, Team & Settings. - Secondary topbar contains workspace search, notifications, user menu, and quick upload button. - Mobile: bottom tab bar for quick switching between Generator and Trends; hamburger menu for other items.

UX patterns: - Skeleton screens for network loads, animated micro-interactions on upload and generation states, hover and focus states for all actionable elements.

---

4. Visual design system

- **Typography:** Sans-serif system (Inter or Helvetica Neue). Use 2–3 weights: Regular (400), Medium (500), Bold (700).
- **Color palette:** Neutral base (off-white #F7F7F7, charcoal #1F2937) + accent teal (#00A99D) and accent coral (#FF6B6B) for statuses. Use greys for UI surfaces.
- **Spacing:** 8px baseline grid. Generous whitespace, card shadows with subtle elevation.

- **Components:** Buttons (primary/secondary/ghost), Cards, Data table, Tag chips, Avatars, Modal, Toasts, Progress skeletons.
- **Micro-interactions:** subtle scale + shadow on hover for cards, progress shimmer while rendering, confirmation toasts for exports.

Accessibility: WCAG AA contrast, keyboard navigation, aria labels for uploads and player controls.

---

5. Detailed UI components and interactions

1. **Upload Dropzone:** large dashed container with icon, accepts drag & drop; shows thumbnails grid once files are added.
2. **Presets Modal:** choose model packs and diversity presets.
3. **Job Activity Panel:** shows per-job stage, logs, and retry options.
4. **Preview Modal:** high-res playback with scrubber, camera angle selector, and frame capture button.
5. **Metadata Editor:** quick inline editor to modify tags, titles, and pricing suggestions.
6. **Bulk Export Wizard:** select assets → choose export template (eCommerce images, video snippets, CSV metadata) → output as ZIP.

---

6. Technical architecture & recommended tech stack

Frontend: React (Next.js for SSR if SEO/data sharing pages needed) or Vue 3 + Vite. **Styling:** Tailwind CSS + component library (Radix UI / shadcn for primitives). **State management:** React Query for server state + Zustand or Redux for local UI state. **Media player:** hls.js or HTML5 for clips; use adaptive bitrate if serving large videos.

Backend: Node.js (Fastify / Express) or Python (FastAPI). Use REST + GraphQL hybrid where GraphQL is used for dashboard queries. **Storage:** PostgreSQL (primary relational) + Redis (caching + rate limiting) + S3 (or S3 compatible object store) for media artifacts. **Job queue & workers:** Redis + BullMQ (Node) or RQ/Celery (Python) for queuing AI generation jobs and postprocessing. **Streaming & real-time:** WebSockets via Socket.IO or Pusher for real-time job progress updates.

Authentication: OAuth2 + JWT, SSO for enterprise customers (SAML optional). **Monitoring:** Prometheus + Grafana; log aggregation via ELK or Datadog.

---

7. Integrations & third-party services

- **AI video generation:** RunwayML, Synthesia, or bespoke model via AWS Sagemaker/Roboflow. Implement an adapter layer to support multiple providers.
- **Trend data:** Google Trends API (or pytrends), social listening (brandwatch, Pulsar), optional fashion data vendors.
- **Storage & CDNs:** AWS S3 + CloudFront (or DigitalOcean Spaces + CDN).
- **Transcoding:** FFMPEG worker pool for format conversion and HLS packaging.
- **Payments & Billing:** Stripe (billing, invoices, metered billing for generated minutes).

8. Data, caching, scaling, and rate limiting

- Cache trend queries (Redis) with short TTLs (5–30 minutes) and stale-while-revalidate pattern for dashboard responsiveness.
- Cache generated asset fingerprints and their metadata to avoid duplicate generation across users.
- API rate limits: per-user and per-org limits; soft limits with throttling and informative headers (X-RateLimit-Remaining).
- Autoscaling workers for rendering jobs; scale down during off-peak.

---

9. Job processing, queuing and concurrency

- Jobs: preprocess (mask extraction, fabric pattern extraction) → model application (texture mapping) → render stills → render clips → runway composition → postprocess (color grading, watermarking).
- Use job priorities and concurrency pools. Long jobs spawn child subjobs for parallelism (multiple camera angles rendered concurrently).
- Implement idempotency keys and deduplication by hashing the input (image + presets) to reuse prior results.

---

10. Security, privacy & compliance

- S3 objects should default to private with signed URLs for access.
- GDPR/CCPA: provide data deletion endpoints and retention policies. Store minimal PII.
- Secure file scanning for malware on upload.
- Role-based access control: org admin, editor, viewer.
- Audit logging for asset access and generation.

---

11. Edge cases & error handling

- **Upload fails:** show file-level error with reason (size, format), retry button, and fallback to direct SFTP for large bulk.
- **Generation timeouts:** set a worker timeout with retry policy; show estimated completion; allow user to continue and receive notification on completion.
- **Trend data missing:** present cached data with timestamp and a warning badge saying "Data may be outdated".
- **Concurrent identical requests:** detect duplicate hash and return cached assets instead of reprocessing.
- **Network latency/offline:** allow resumable uploads and local queueing when offline; sync when back online.

---

12. Exporting & bulk operations

- Bulk ZIP export with structure: `/images/{sku}/`, `/videos/{sku}/`, `metadata.csv`.
- Filtering on export (age group targeting, region tags).
- API for partners to pull generated assets programmatically (signed URL rotation).

---

13. Appendix: API design samples and wireframe notes

(Include sample REST endpoints, core DB tables and wireframe sketches — see next version for full API spec and ERD.)

Next steps

- Prototype key flows: Upload → Generate → Preview (MVP) then integrate trends.
- Create sample set of model presets and 10 seed categories for initial training/fine-tuning of the generation pipeline.
- Run a pilot with 3 retail partners to collect performance and conversion signal to improve the appeal scoring model.

Document created for Md Faizan Ashrafi — detailed product and UI/UX + technical design for a B2B fashion SaaS.