PRD Sample: AI-Powered Web Platform

What The Style (WTS): AI wardrobe styling and virtual try-on

2026

What The Style (WTS): AI Wardrobe Platform

Document type: Product Requirement Document
Version: 1.0
Scope: Web application + AI styling pipeline + credit-based billing

---

1. Overview

What The Style (WTS) is an AI-powered personal styling platform. Users photograph items from their physical wardrobe, and the platform organises the clothing using AI. When the user has an occasion to dress for, they describe it in plain language and an AI stylist assembles an outfit from their own clothes. Finally, the platform generates a photoreal image of the user wearing the completed outfit (before a single item leaves the hanger).

The platform is built around three user-facing actions: **Snap** (upload wardrobe items), **Ask** (describe an occasion; receive an AI-curated outfit), and **See** (generate a try-on render on the user's own photo). Storage defaults to the user's personal Google Drive; an optional WTS Cloud plan is also available.

This format illustrates a PRD for a web platform that integrates multiple AI systems: image classification, large language model (LLM) outfit reasoning, and image generation for virtual try-on.

---

2. User Personas

Everyday Styler A working professional who has a full wardrobe but struggles to put outfits together for specific occasions. Wants AI help without shopping for new items.

Occasioned Dresser Dresses carefully for specific events (dinner parties, dates, weddings). Uses WTS to plan outfit combinations and see the result before getting dressed.

Minimalist Shopper Wants to reduce impulse buying by knowing exactly what combinations their current wardrobe allows, before purchasing anything new.

---

3. AI Pipeline Overview

The platform relies on three distinct AI subsystems:

Step	Subsystem	Input	Output
1	Image classifier	Photo of a clothing item	Category, colour, tags, background-removed image
2	LLM stylist	Occasion description + wardrobe item list	Outfit (set of item IDs) with reasoning

Step	Subsystem	Input	Output
3	Image generator	User portrait photo + outfit item images	Photoreal try-on render

Each try-on render consumes one credit. Steps 1 and 2 (upload and styling) do not consume credits.

---

4. Data Models

4.1 User

Field	Type	Notes
`id`	uuid	
`email`	string	Primary identifier; login via OAuth
`display_name`	string	
`portrait_photo`	file	The user's own photo used for try-on renders
`storage_type`	enum	`google_drive` or `wts_cloud`
`google_drive_token`	string	OAuth token for Drive API; nullable
`credit_balance`	integer	Current credits; debited per try-on render
`created_at`	timestamp	

4.2 Wardrobe Item

Field	Type	Notes
`id`	uuid	
`user_id`	uuid	Foreign key to User
`original_photo_url`	string	URL of original uploaded photo
`processed_photo_url`	string	Background-removed version
`category`	enum	`top`, `bottom`, `footwear`, `outerwear`, `accessory`, `dress`, `suit`
`colour`	string	Primary colour name, AI-detected
`tags`	string array	AI-detected attributes: e.g. `casual`, `formal`, `striped`, `denim`
`label`	string	Optional user-defined name (e.g. "Blue Oxford shirt")
`is_active`	boolean	Hidden items are excluded from AI styling
`storage_path`	string	File path within Google Drive or WTS Cloud
`created_at`	timestamp	

4.3 Outfit

Field	Type	Notes
`id`	uuid	
`user_id`	uuid	
`occasion`	string	Free text occasion entered by user, e.g. "First date"

Field	Type	Notes
item_ids	uuid array	The wardrobe item IDs included in this outfit
ai_reasoning	text	Explanation from the LLM of why this combination was chosen
created_at	timestamp	

4.4 Try-On Render

Field	Type	Notes
id	uuid	
outfit_id	uuid	
user_id	uuid	
render_url	string	URL of the generated try-on image
credits_used	integer	Always 1 per render at v1
status	enum	pending, processing, complete, failed
created_at	timestamp	

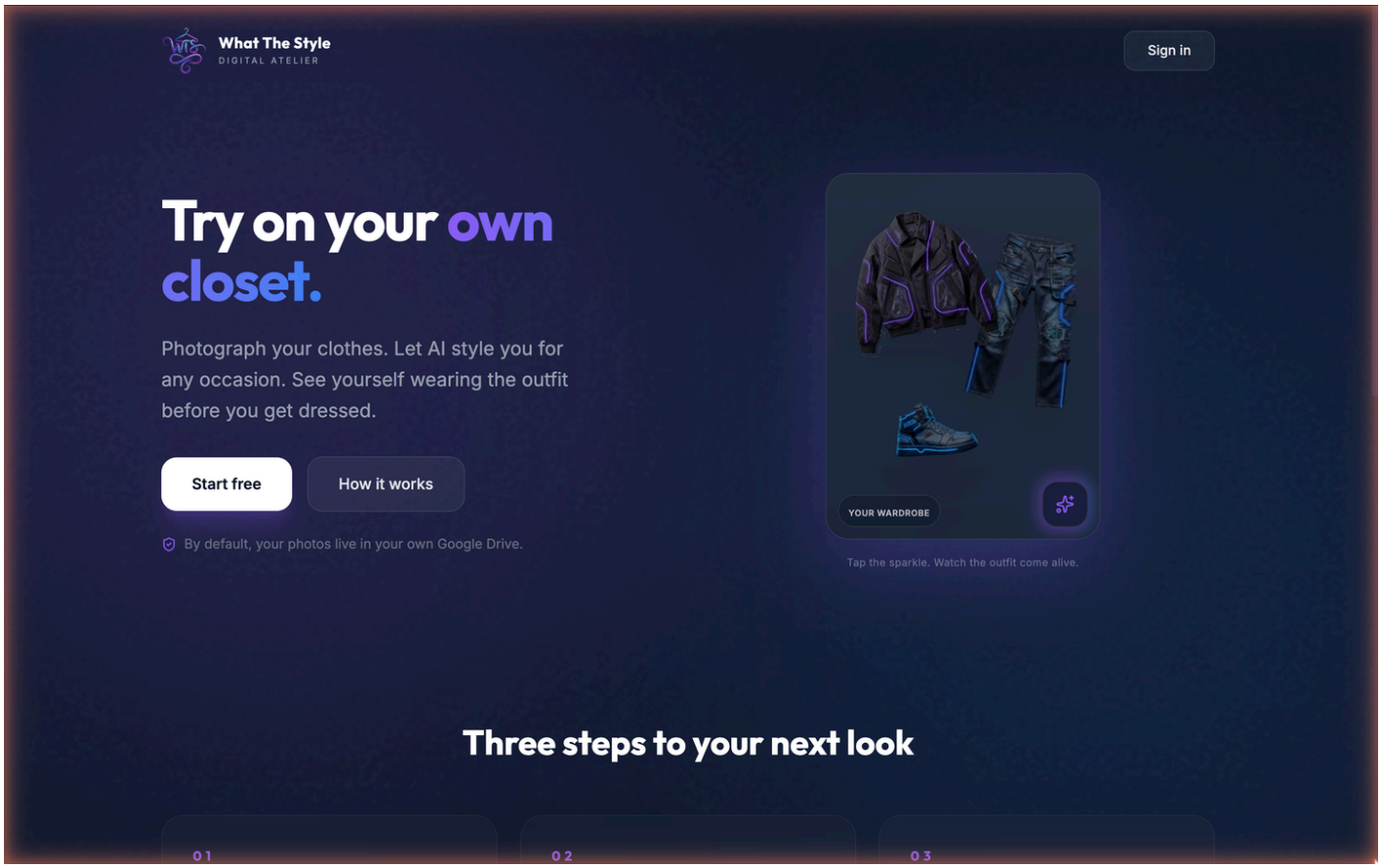
4.5 Credit Transaction

Field	Type	Notes
id	uuid	
user_id	uuid	
type	enum	purchase, debit, refund, bonus
amount	integer	Positive for purchases/bonuses; negative for debits
description	string	e.g. “Try-on render”, “Credit top-up: 20 credits”
reference	string	Payment gateway transaction ID for purchases; nullable
created_at	timestamp	

---

5. Feature Requirements

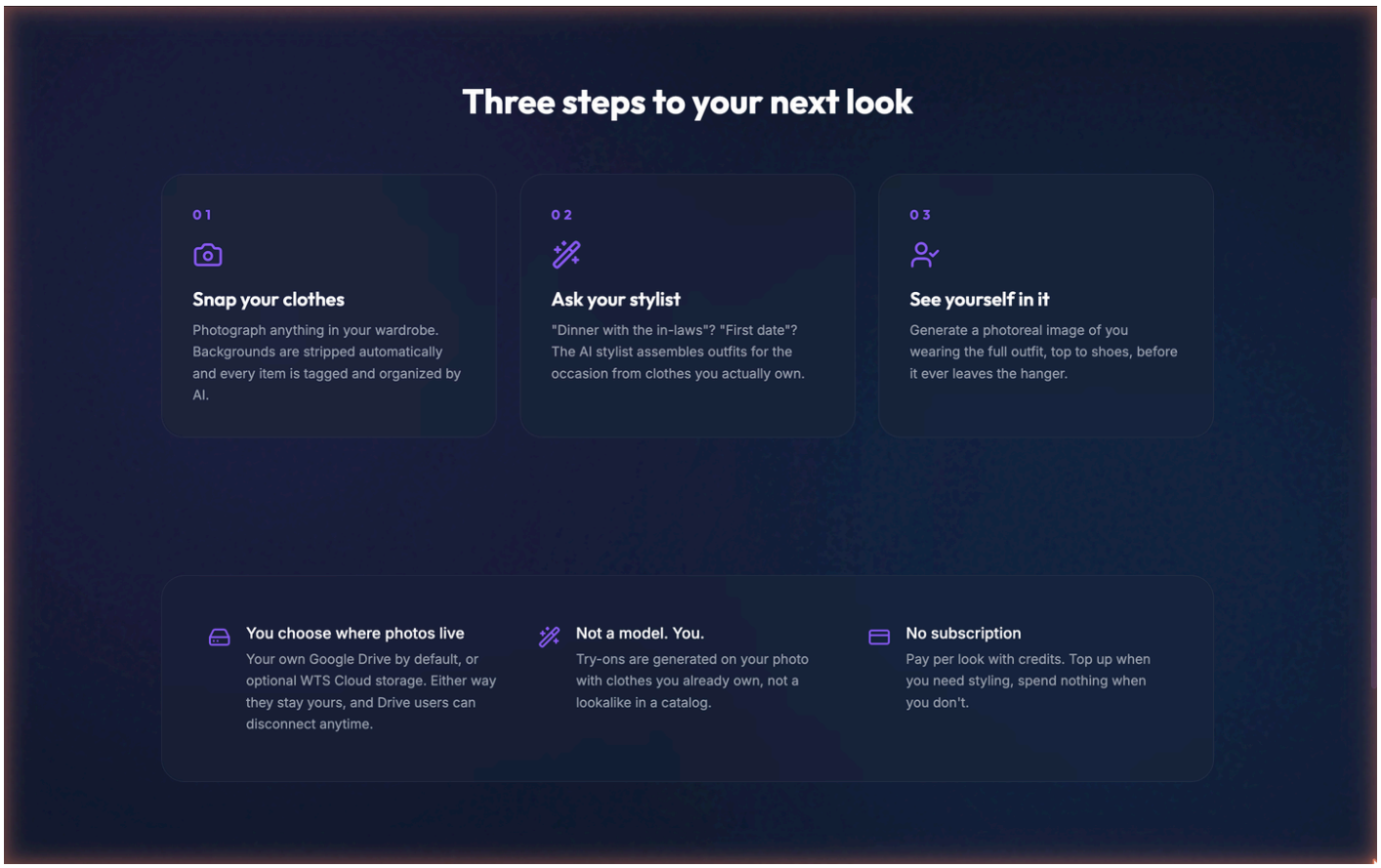
Feature 1: Landing Page & Onboarding



Route: /

Requirements:

1. Hero section communicates the core value proposition: “Try on your own closet.” with a supporting description and two CTAs: **Start free** and **How it works**.
2. “By default, your photos live in your own Google Drive.” trust badge is displayed beneath the primary CTA.
3. “How it works” scrolls to the three-step section on the same page.
4. “Start free” navigates to the sign-up / OAuth flow.
5. A “Sign in” button in the top navigation links to the login page for returning users.



Three Steps section:

1. **Snap your clothes** — Photograph anything in your wardrobe. Backgrounds are stripped automatically and every item is tagged and organised by AI.
2. **Ask your stylist** — Describe any occasion in plain language. The AI stylist assembles an outfit from clothes you actually own.
3. **See yourself in it** — Generate a photoreal image of you wearing the full outfit, top to shoes, before it ever leaves the hanger.

Differentiation points displayed below:

- You choose where photos live (Google Drive or WTS Cloud)
- Not a model. You. (renders use the user's own portrait)
- No subscription (pay per look with credits)

Feature 2: Authentication

Acceptance criteria:

1. Sign-up and sign-in use OAuth only (Google). No email/password accounts in v1.
2. On first sign-in, the user is prompted to connect their Google Drive (for Drive storage) or opt for WTS Cloud storage.
3. If the user chooses Google Drive, the app requests the minimum required OAuth scope: access to a single designated WTS folder within Drive. The user can revoke this access at any time from their Google account settings.
4. After completing setup, the user lands on their Wardrobe dashboard.
5. New users receive a welcome credit bonus of 3 free try-on renders.

Feature 3: Wardrobe Management (Snap)

Acceptance criteria:

1. The Wardrobe screen shows all active wardrobe items as a grid of background-removed images.
2. The user can upload one or more photos by dragging files into the upload zone, or tapping a camera icon to use the device camera (mobile).
3. After upload, each item is processed by the AI classifier within 30 seconds. The item card shows a spinner and "Processing…" label during this time.
4. Once processed, the item card shows: background-removed image, detected category badge, and colour chip. The AI-generated tags are shown on hover/tap.
5. The user can edit the item label, category, and tags at any time. Edited fields override AI-detected values for styling purposes.
6. The user can hide an item (toggle `is_active = false`). Hidden items are excluded from future AI styling suggestions but remain in the wardrobe.
7. The user can delete an item permanently. A confirmation dialog is shown: "Remove this item from your wardrobe? This cannot be undone." The item photo is deleted from storage.
8. Wardrobe items are filterable by category: All, Tops, Bottoms, Footwear, Outerwear, Accessories, Dresses, Suits.

Feature 4: AI Styling (Ask)

Acceptance criteria:

1. An "Ask your stylist" input is accessible from the Wardrobe screen and the Outfits screen.
2. The user types a free-text occasion description, e.g. "First date at a rooftop restaurant", "Board meeting", "Beach day", or "Wedding guest".
3. The LLM stylist processes the occasion description alongside the user's wardrobe item list (categories, colours, and tags). It selects a set of items (top, bottom, footwear, and optional outerwear/accessory) and returns:
    - The selected item IDs
    - A brief explanation of why these items work for the occasion
4. The proposed outfit is displayed as a flat-lay preview: each selected item is shown as a card in a row, labelled by category.
5. The user can swap any item in the outfit: clicking "Change" on a slot opens a picker showing only items of the same category. The user selects an alternative and the outfit is updated.
6. The user can request a completely new suggestion for the same occasion by clicking "Try another look". This generates a new outfit from different items. Up to 3 suggestions are available per occasion query.
7. Generating outfit suggestions (steps 1–6) does not consume credits.
8. An outfit can be saved to the user's Saved Outfits list at any time.

Feature 5: Virtual Try-On (See)

Acceptance criteria:

1. From a proposed or saved outfit, the user can click "Try it on" to generate a photoreal render.
2. Before the first render, if no portrait photo has been uploaded, the user is prompted to upload one. Requirements: face clearly visible, neutral background preferred, full-body or upper-body shot. Accepted formats: JPEG, PNG, WebP; max 10 MB.
3. Clicking "Try it on" debits 1 credit and starts a render job (`status = processing`). A progress indicator is shown on screen.
4. If `credit_balance = 0`, the user is shown the credit purchase screen instead. Credits must be purchased before proceeding.
5. Renders are typically available within 60–120 seconds. The user can navigate away; the render appears in their Renders gallery when complete.
6. On completion, the user is notified (in-app banner and, if permitted, browser push notification): "Your look is ready. See yourself in it."
7. The render shows a full-body image of the user wearing all selected outfit items.
8. The user can download the render as a JPEG or share it directly (Web Share API on supported browsers).

9. If a render fails (`status = failed`), 1 credit is automatically refunded and the user is notified: "Something went wrong with your render. Your credit has been returned."

Feature 6: Credits & Billing

Acceptance criteria:

1. Credit balance is shown in the navigation header on all authenticated screens.
2. The Credits page shows: current balance, full transaction history (type, amount, description, date), and credit purchase options.
3. Credit packs available in v1:

Pack	Credits	Price
Starter	5	$4.99
Standard	20	$14.99
Pro	50	$29.99

4. Payment is handled by Stripe Checkout. After a successful payment, the credit balance is updated immediately and a `purchase` transaction is recorded.
5. Credits do not expire.
6. After a successful purchase, the user receives an email receipt (via Stripe).

Feature 7: Saved Outfits & Renders Gallery

Acceptance criteria:

1. The Outfits screen lists all saved outfits as cards showing the flat-lay item grid, occasion label, and creation date. Sorted newest first.
2. Clicking an outfit card reopens the outfit for review, item swapping, or a new try-on render.
3. The Renders screen (gallery) shows all completed try-on renders as a photo grid, sorted newest first.
4. Each render card shows the occasion label, date, and a download/share icon.
5. The user can delete a render. The render image is removed from storage.

Feature 8: Storage Settings

Acceptance criteria:

1. The Settings screen shows the current storage type (Google Drive or WTS Cloud) and, for Drive users, the connection status (connected / disconnected).
2. Drive users can disconnect their Google Drive. On disconnect, the app stops syncing new uploads to Drive; existing files remain in Drive and are not deleted by WTS.
3. Drive users can switch to WTS Cloud storage. This migrates future uploads only; existing Drive-hosted files remain accessible via their stored URLs.
4. WTS Cloud users can connect a Google Drive at any time to switch storage.

---

6. AI System Constraints & Guardrails

Constraint	Requirement
Wardrobe item classifier	Must return a category for 95%+ of wardrobe photos within 30 seconds

Constraint	Requirement
Background removal	Must produce a clean cutout for standard clothing photos on flat or moderately busy backgrounds
LLM stylist	Must not suggest items not in the user's wardrobe. Must return at least one complete outfit (top + bottom + footwear or dress + footwear)
LLM guardrail	If occasion input is offensive or off-topic, respond: `I can only help with outfit ideas. Try describing an occasion or event.'' | | Try-on renderer | Must render all selected items on the user's portrait. Must not alter the user's face, body shape, or skin tone | | Failed render policy | If the portrait photo is unsuitable (face not detected, image too blurry), return a user-facing error before debiting credits:` We couldn't process your portrait. Please upload a clearer photo."

---

7. Scope Boundaries

In scope:

- Landing page with value proposition and onboarding
- Google OAuth sign-in
- Google Drive integration for photo storage (default) + WTS Cloud option
- Wardrobe upload, AI classification, background removal, tagging
- AI outfit suggestion from natural language occasion input (LLM)
- Outfit flat-lay preview with item swap
- Virtual try-on render (image generation) on user portrait, credit-gated
- Credit purchase via Stripe Checkout (3 packs)
- Saved outfits and renders gallery
- In-app credit balance and transaction history

Out of scope for v1:

- Shopping / e-commerce integration (buying new items)
- Social features (sharing looks publicly, following other users)
- Style recommendations based on weather or calendar
- Recurring subscription plan (credits-only in v1)
- Mobile app (iOS / Android)
- Multi-user accounts or family plans
- Outfit scheduling or calendar sync
- Retailer affiliate links

---

8. Prerequisites

1. AI image classifier: background removal and clothing categorisation (e.g. Remove.bg, custom fine-tuned model).
2. LLM API for outfit reasoning (e.g. OpenAI GPT-4o, Anthropic Claude).
3. Image generation API for virtual try-on (e.g. Replicate, fal.ai, or custom diffusion model with garment conditioning).
4. Google OAuth credentials and Drive API access.
5. Stripe account and Checkout integration.
6. File storage: Google Drive API (user's Drive) + cloud object storage for WTS Cloud option.
7. Email delivery for render completion notifications and receipts.
8. Portrait photo quality validation before rendering (face detection check).

This is a sample PRD for an AI-powered web platform. It illustrates how to specify AI subsystems alongside standard web features, document credit-based billing flows, and define guardrails for generative AI outputs.