

# TeamName: Git It Girls – StyleSense AI Supplementary Documentation

---

## Table of Contents

1. Executive Summary
2. Problem Statement
3. Stakeholder Analysis
4. Vision & Core Objectives
5. Solution Overview: StyleSense AI
6. Technical Architecture
  - System Diagrams
  - Component Breakdown
  - Data Flow & Lifecycle
  - Persistence & Scalability
7. Implementation Details
  - Backend
  - AI & ML Services
  - Frontend
  - Data Management
8. Samsung AI Ecosystem & Integration
  - Technical Approach
  - Device & Platform Leverage
  - Security/Privacy with Knox
  - Health, Lifestyle & Smart Home Fusion
9. Business Plan
  - Market Opportunity
  - Customer Segments
  - Go-to-Market
  - Monetization
  - Competitive Edge
  - Risks & Scalability

## 10. Evaluation & Impact Metrics

## 11. Appendix

- Team Bios
  - File Structure
  - References & Datasets
-

## 1. Executive Summary

StyleSense AI is an advanced GenAI-powered multimodal fashion assistant designed to make fashion recommendations hyper-personalized, emotion-aware, and contextually intelligent. Uniting persistent wardrobe memory, celebrity “twin” matching, contextual weather intelligence, and end-to-end AI-driven synthesis, our solution creates a superior user experience, seamless retailer integration, and robust business scalability.

---

## 2. Problem Statement

The AI fashion vertical, projected to surge to \$60.57B by 2034, is limited by:

- Emotional Blindness: Lacks user feeling/context, focuses only on functional filters.
- Static Experiences: One-off suggestions; no sense of evolving personal style.
- Poor Multimodal Fusion: Struggles to connect voice, image, text, and context at scale.
- Cultural Disconnect: Misses festival/regional trends, critical for global relevance.
- Memory Gap: No persistent fashion history or future stylistic prediction.

Our research highlights a 39.12% CAGR in personalized AI fashion, with 71% of users demanding deeper personalization.

---

## 3. Stakeholder Analysis

- End Users: Seek expressive, fulfilling styling guidance, contextually aware wardrobe choices, and stress-free shopping.
  - Retailers & Brands: Want improved conversions, trend intelligence, and direct paths to purchase.
  - Tech Ecosystem Partners (Samsung, API vendors): Need plug-and-play, privacy-respecting, high-value consumer AI.
  - Developers & Researchers: Value modularity, extensibility, and reproducible design.
- 

## 4. Vision & Core Objectives

- Democratize high-confidence fashion assistance, making every user feel “seen.”
  - Deliver live, evolving personal recommendations rooted in memory, emotion, and global trend awareness.
  - Leverage advancements in generative AI and edge-compute platforms (Samsung).
  - Bridge gaps between offline wardrobe, online shopping, and contextual living.
-

## 5. Solution Overview: StyleSense AI

### Key Innovations

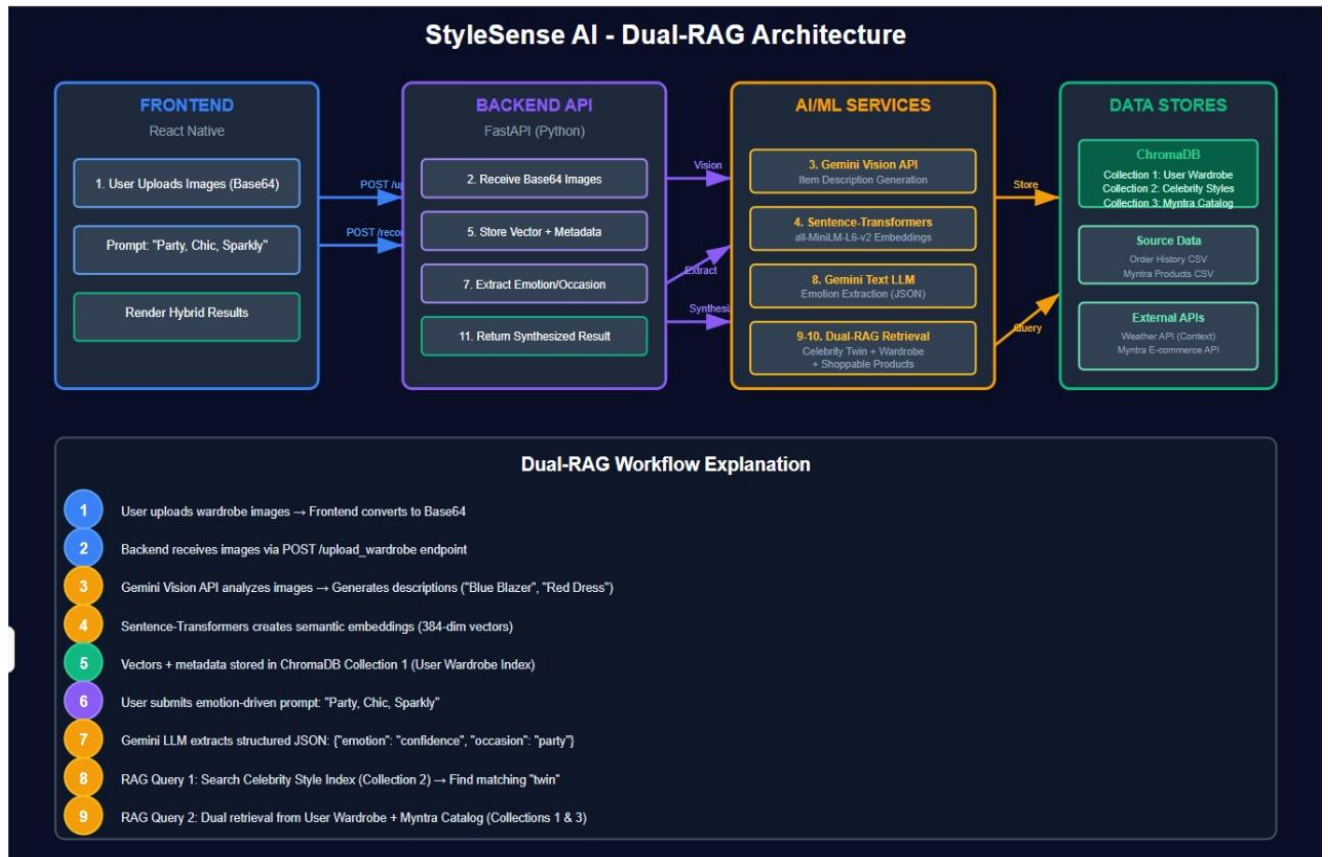
- Emotion-Memory Fusion Stylist: AI that merges real-time emotion recognition with persistent wardrobe evolution, generating “outfit signatures.”
- Hybrid Dual-RAG Retrieval: Multimodal search from user wardrobe, celebrity twins, and shopping catalogs.
- Real-World Feedback/Social Fusion: Anticipates event/weather/social response before final recommendations.
- Integrated AR/VR Try-On: Augmented reality for instant feedback and delight.
- Dynamic Cultural Intelligence: Celebrates local festivals and traditions, not just Western-centric fashion.

### User Flow Summary

1. First-time user uploads wardrobe images; ChromaDB vectors persist user “memory.”
  2. User submits a prompt (e.g., “Internship, confident, cool weather”).
  3. System checks weather API for live context; Gemini LLM extracts required emotion/occasion.
  4. Hybrid RAG retrieves:
    - “Wardrobe twin” celebrity look
    - Outfit matches from user’s collection
    - Shoppable items from Myntra catalog (with links)
  5. AI synthesizes a contextually-aware, Markdown-formatted, attractive result.
  6. User gets persistent benefits, never needing to re-upload.
-

## 6. Technical Architecture

### 6.1 System Architecture Diagrams



- The diagram shows:
  - Frontend Layer: React Native/App, AR SDK
  - Backend Layer: FastAPI, Security/Auth
  - AI/ML Layer: Gemini API, Vision Models, Embedding Models, Orchestrator
  - Persistence: ChromaDB, Caching
  - Integrations: Weather, Myntra, Social

Component	Technology	Details
Frontend	React Native	Attractive onboarding, wardrobe image upload, prompt submit, receives final rec markdown + links
Backend (API)	FastAPI (Python)	Orchestrates entire flow, API routing, middleware, persistent DB connector
LLM/AI Engine	Gemini API (Vision+Text)	Multimodal extraction: prompt understanding, emotion, occasion, style cue synthesis
Embedding Engine	Sentence-Transformers	Vectors for wardrobe/celebrity/catalog for rapid, relevant retrieval
Persistent Vector DB	ChromaDB (server-client)	Stores wardrobe, celebrity, catalog as logical “collections”
Datasets	Myntra, Kaggle, Celeb	.csv, .jpg used with loader scripts for quick DB population
External APIs	Weather, Instagram, Calendar	Used for contextual retrieval and social validation
Orchestration & Security	LangChain, Python	Workflow logic, failover handling, authentication via secrets/environment

### 6.2 Component Breakdown

### 6.3 Data Flow & Lifecycle

- Image Upload:
  - User uploads images (Base64 or direct file).
  - Backend encodes and generates vectors using embedding engine.
  - Vectors stored in wardrobe collection (ChromaDB).
- Prompt Submission:
  - User uses prompt (text/voice).
  - Gemini API processes for emotion/context.

- Weather/calendar/social APIs queried.
3. Multi-Hop Retrieval:
- Hybrid RAG searches all collections:
    - Finds “twin” in celeb DB.
    - Finds matches in wardrobe DB.
    - Locates buyable items in Myntra DB.
4. Synthesis & Output:
- LLM crafts final result: Markdown block of recommendations, clickable links, and explanation.
  - Returned as JSON to frontend; rendered as rich cards.

## 6.4 Persistence, Privacy, and Scalability

- ChromaDB enables persistent, server-client wardrobe memory (users don’t need to re-upload).
  - All sensitive info is saved securely; only vector representations are held long-term, not raw images.
  - Modular backend can be horizontally scaled (more API nodes, FastAPI, ChromaDB cluster).
  - Built for cloud deployment: AWS/Google Cloud; can extend to on-device vector cache in future.
  - Privacy is enforced: API keys in .env, user data anonymized, Knocks integration (see Samsung section).
- 

## 7. Implementation Details

### 7.1 Backend (FastAPI)

- main.py starts server and registers routers.
- app/api.py: Core endpoints (/upload\_wardrobe, /recommend, /status).
- app/database.py: ChromaDB server-client interface and collection setup logic.
- app/data\_loader.py: One-off scripts for bulk ingest from data/ to ChromaDB.
- app/prompts.py: Stores all prompts for DRY code, easy re-tuning.
- app/recommender.py: Main orchestration logic (Dual-RAG, weather, LLM, return synthesis).
- .env: Secures API keys/tokens from code.

### 7.2 AI & ML Services

- Gemini API for all prompt parsing, occasion/emotion/context extraction, and stylish text synthesis.
- Vision Models: Extract fashion elements from wardrobe images (color, cut, type).
- Sentence-Transformers: Same engine vectorizes all catalog, celebrity, and user data for semantic retrieval.
- End-to-End Orchestration: Calls modules in sequence, fallbacks in case of API unavailability.

### **7.3 Frontend (React Native)**

- index.tsx: Main home screen with upload, AI chat, and navigation.
- \_layout.tsx: Root navigation and context provider setup.
- api.ts: Backend API client with TypeScript interfaces.
- contexts/UserContext.tsx: Global user state management and persistence.
- components/: Reusable UI components (Splash, UploadProgressView).
- app/screens/: Feature screens (recommend, profile, model/closet).

### **7.4 Data Management**

- Datasets include:
  - Order History.csv: User “memory.”
  - myntra202305041052.csv: Product search.
  - Celeb-FBI Dataset/: All celebrity images.
- Data flow ensures GDPR-like standards: vectors only, no raw photos stored in database.

---

## **8. Samsung AI Ecosystem & Integration**

### **8.1 Galaxy AI & Device Integration**

- Bixby Voice: Multi-turn styling commands. “Suggest an outfit for tomorrow’s meeting.”
- Circle to Search (Edge Devices/Phones): Instantly get fashion recs from images/context.
- Samsung 8th Gen AI chip: On-device emotion recognition; rapid, private edge inference.
- SmartThings:
  - Smart mirrors show outfit picks in real time.
  - Lighting/climate from home reflected in style suggestions.



- Multi-user (family) fashion plans via AI Home.

## **8.2 Knox Security & User Privacy**

- Defense-grade privacy: Wardrobe vectors and biometric fashion “signatures” protected via Knox.
- Secure Folder: Ultimate privacy; wardrobe & recs always user-owned.
- No cloud data leaks; all API secrets in secured .env files.

## **8.3 Lifestyle & Health Fusion**

- Leverage Samsung Health data for fitness/stress-aware outfits (post-gym, workday, festivals).
- Supports motivation & well-being through mood-boosting style choices.

## **8.4 Business/Ecosystem Extension**

- White-label ready: Copy-paste infrastructure to other Samsung/SmartThings retail partners.
  - All major APIs, datasets, and orchestrator modules built for modular swapping.
- 

# **9. Business Plan**

## **9.1 Market Opportunity**

- Addressing a \$60.5B total addressable market in AI/retail fashion by 2034.
- 39.1% CAGR; 71% of users crave personalization.
- Business model built for retail partnerships, affiliate, in-app upsell, and white-label B2B options.

## **9.2 Customer Segments**

- Mobile-first Gen Z and Millennial shoppers, especially in growing e-commerce regions.
- Luxury and mass-market retailers looking for plug-and-play AI upskilling.
- Families and working professionals (via SmartThings integration).

## **9.3 Go-to-Market Plan**

- Phase 1: Launch Android (Samsung Galaxy Store), demo at hackathons, integrate FAQs/demo scripts.
- Phase 2: Direct B2B outreach to e-commerce, fashion, and AR/VR platforms.
- Phase 3: Family+ home, workplace, and smart home verticals.

## 9.4 Monetization & Growth

- Affiliate Revenue: Clickout sales from Myntra/catalog integrations.
- Premium Features: Save more wardrobe slots, enhanced analytics, AR try-on, exclusive celeb “twins.”
- B2B: Licensing SDK to retail partners, white-labeling for Samsung & SmartThings.

## 9.5 Competitive Advantage

- Only GenAI solution natively combining emotion, persistent memory, and robust multimodal retrieval.
- Pre-built for Samsung integration—no lag on device/AI/UX sync.

## 9.6 Risks & Scalability

- Mitigates data privacy risk via on-device compute.
- Designed for global scaling (multi-cloud, mobile-optimized, modular APIs).
- Built-in fallback routines for external API downtime (offline mode supported).

---

## 10. Evaluation & Impact Metrics

- Customer Satisfaction: NPS, retention, repeat styling sessions, referral rates.
- Engagement: Outfits recommended per session, save/share rate, average session time.
- Conversion: Affiliate conversion; links clicked, purchases tracked via UTM analytics.
- Tech Reliability: Uptime, median API latency, error rates, persistent data recovery.
- Ecosystem Impact: Samsung device adoption, integration NPS, merchant partner feedback.

---

## 11. Appendix

### 11.1 Team Bios

#### Arundhati Chaturvedi

- Role: Backend Lead, Technical Integrator
- Expertise: Python backend development, API orchestration, RAG systems, ChromaDB, deployment
- Key Contributions: Designed server-client persistence, dual-RAG logic, security protocols

#### Priyasha Mohanty

- Role: Data/ML Lead and Code Walkthrough

- Expertise: AI/ML engineering, prompt writing, LLM/Vision API integration, data transformation
- Key Contributions: Built Gemini and Vision Model pipelines, prompt/embedding logic, model evaluation

### **Dhriti Parikh**

- Role: UI/UX, Frontend and Demo Lead
- Expertise: React Native/Flutter, modern/mobile UI des, user onboarding, user demo scripting
- Key Contributions: Created compelling onboarding, API integration on client, demo/storyline, usability scripts

## **11.2 File Structure**

Stylesense\_ai/

```

├── .gitattributes
├── .gitignore
├── LICENSE
├── README.md
├── datasets.md
├── run_local.sh
|
├── .expo/
|   ├── README.md
|   └── settings.json
|
├── backend/
|   ├── .env
|   ├── complete_sys_test.py
|   ├── debug_celebrity.py
|   ├── main.py
|   ├── requirements.txt
|   ├── test_gemini_api.py
|   └── test_results.json

```

```
| |
| |─ app/
| |   |─ _init_.py
| |   |─ api.py
| |   |─ data_loader.py
| |   |─ database.py
| |   |─ recommender.py
| |   └─ _pycache_/
| |       |─ _init_.cpython-XX.pyc
| |       |─ api.cpython-XX.pyc
| |       |─ data_loader.cpython-XX.pyc
| |       |─ database.cpython-XX.pyc
| |       └─ recommender.cpython-XX.pyc
| |
| |─ chroma/
| |   └─ chroma.sqlite3
| |
| |─ data/
| |   |─ Celeb_FBI_Dataset/
| |   |   |─ celebrity_images/
| |   |   |   |─ actor1.jpg
| |   |   |   |─ actor2.jpg
| |   |   |   └─ ...
| |   |   └─ celebrity_metadata.json
| |   |─ myntra202305041052.csv
| |   └─ Order_History.csv
|
|─ frontend/
|   |─ .gitignore
|   |─ app.json
```

```
| |— eslint.config.js
| |— expo-env.d.ts
| |— package.json
| |— README.md
| |— tsconfig.json
| |
| |— .expo/
| | |— devices.json
| | |— README.md
| | |— settings.json
| |
| |— .vscode/
| | |— extensions.json
| |
| |— app/
| | |— index.tsx
| | |— _layout.tsx
| | |
| | |— services/
| | | |— api.ts
| | |
| | |— screens/
| | | |— model.tsx
| | | |— profile.tsx
| | | |— recommend.tsx
| | |
| |— assets/
| | |— fonts/
| | | |— BubbleBobble-rg3rx.ttf
| | |
```

```
| | └─ images/
| |   └─ splash_bg.png
| |   └─ profile_pic.png
| |   └─ margo.jpg
| |   └─ models.png
| |
| └─ components/
|   └─ Splash.tsx
|   └─ uploadProgressView.tsx
|
| └─ contexts/
|   └─ UserContext.tsx
|
| └─ utils/
|   └─ helpers.ts
```

### 11.3 References & Datasets

- Datasets
  - Myntra Catalog — <https://www.kaggle.com/datasets/ronakbokaria/myntra-products-dataset>
  - Order History — <https://www.kaggle.com/datasets/archit9406/customer-transaction-dataset>
  - Celeb-FBI Dataset — <https://www.kaggle.com/datasets/pronaydebnath1/celeb-fbi>
- APIs
  - Gemini Vision/Text API (Google)
  - Weather API (OpenWeatherMap or similar)
  - Instagram API (trends)
  - Calendar (Google/O365)
- Frameworks/Libraries
  - FastAPI
  - ChromaDB

- LangChain
- Sentence-Transformers
- React Native
- Hackathon Submission
  - Demo links —