IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

DrobeDex: An AI-Powered Smart Wardrobe and Outfit Planner

Kanish Rishab D[1], Manish V[2], Prajval Gowda[3], Dr. Abhilash C N[4]

Student, AIML, SJB Institute of Technology, Bengaluru, India[1]

Student, AIML, SJB Institute of Technology, Bengaluru, India[2]

Student, AIML, SJB Institute of Technology, Bengaluru, India[3]

Professor and HOD, AIML, SJB Institute of Technology, Bengaluru, India[4]

Abstract: In today's fast-paced digital world, individuals face increasing challenges in managing their personal wardrobes efficiently. Despite owning a variety of clothing, many struggle with repetitive outfit choices, underutilization of their wardrobe, and daily decision fatigue. DrobeDex is an AI-powered smart wardrobe and outfit planner Progressive Web Application (PWA) designed to address these challenges by combining user-centric design with intelligent automation. The application allows users to digitize their clothing collections through image uploads, which are then auto-tagged using a custom ResNet50 model. Users can create, manage, and log daily outfits using a drag-and drop interface. DrobeDex is built using React.js to ensure cross platform compatibility and smooth performance on modern web browsers. It employs efficient client-side image preprocessing and integrates a custom ResNet50 model for auto-tagging alongside Generative AI for personalized outfit recommendations. The project follows an Agile development methodology, enabling iterative improvement based on user feedback and real-world testing. Index Terms—AI-powered wardrobe, outfit planning, deep learning, React Native, sustainability, personalized recommendations.

Keywords: AI-powered wardrobe, outfit planning, deep learning, sustainability, personalized recommendations.

I. INTRODUCTION

A. Overview

The contemporary fashion landscape stands at a critical juncture, defined by a stark dichotomy between unprecedented technological sophistication in retail and a chaotic, unsustainable post-purchase reality for the consumer. Over the last decade, the digitization of commerce has revolutionized how clothing is marketed and sold. Algorithms predict trends, optimize supply chains, and personalize shopping feeds with uncanny accuracy. Yet, the moment a transaction is complete, this digital intelligence evaporates. DrobeDex represents a strategic technological intervention designed to resolve this paradox. It is a comprehensive, Intelli gent Wardrobe Management System (IWMS) built as a Single Page Application (SPA). Unlike traditional inventory tools that act as static databases, DrobeDex leverages the emergent capabilities of multimodal Generative Artificial Intelligence (GenAI)—specifically the Gemini 2.5 Flash model—to act as an active, cognitive agent.

B. Background of the Study

To understand the necessity of DrobeDex, one must first analyze the converging crises in sustainability, consumer psychology, and the limitations of legacy technology.

- *1) The Sustainability Imperative:* The fashion industry operates primarily on a linear model: "take-make-waste". This model is responsible for approximately 10% of global carbon emissions and nearly 20% of global wastewater. Research from the Ellen MacArthur Foundation indicates that the average number of times a garment is worn before it ceases to be used has decreased by 36% compared to 15 years ago.
- *2) The Psychology of Organization:* Beyond the environmental cost, the modern wardrobe imposes a significant cognitive tax on the individual. The average adult makes approximately 35,000 remotely conscious decisions each day. The cumulative effect of these choices depletes mental energy, a phenomenon known as decision fatigue.
- *3) The Technological Context:* The advent of Multimodal Large Language Models (like Gemini 1.5/2.5) represents a paradigm shift. These models possess "Zero-Shot" capabilities, allowing them to understand and classify images based on broad, pre-trained knowledge of the world

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

C. Problem Formulation

Despite the clear need for better wardrobe management, the market lacks a solution that effectively combines ease of use, deep personalization, and sustainability tracking. The core problem is the friction of digitization and the lack of contextual intelligence.

1) Critical Limitations of Current Solutions: Existing solutions generally fall into three categories:

- **Static Inventory Applications:** The most common ”closet apps” function as simple CRUD databases. To add an item, the user must take a photo, then manually type the brand, select the color from a dropdown, choose the category, and tag the season.
- **E-Commerce Driven ”Personal Stylists”:** These algorithms are optimized to recommend new purchases based on browsing history. They have no visibility into what the user already owns.
- **Generic Generative AI:** Users can ask standard chatbots for fashion advice, but these interactions suffer from a ”Grounding Problem”.

2) Primary Technical Objectives:

- **Develop a Frictionless Client-Side Ingestion Pipeline** using a custom ResNet50 model to auto-tag wardrobe items with > 90% accuracy without server latency.
- **Engineer a Context-Aware Retrieval Augmented Generation (RAG) System**.
- **Construct a Visual ”Outfit Canvas” Interface**.

3) Research Questions:

- **RQ1:** Can multi-modal LLMs sufficiently automate the tagging of fashion items?
- **RQ2:** How effective is a context-window injection strategy in grounding an LLM?
- **RQ3:** Does the visualization of ”Wear Count” and ”Cost Per Wear” metrics influence user behavior?

D. Significance of the Study

This project contributes to the growing body of research on Applied Generative AI. It demonstrates practical techniques for ”Grounding” LLMs to small, dynamic private datasets and provides a scalable technological framework for reducing textile waste.

E. Scope and Limitations

The scope includes authentication, inventory digitization, visual studio, conversational stylist, and local persistence. Exclusions include social networking, e-commerce functionality, advanced computer vision, and virtual try-on features.

II. LITERATURE REVIEW

A. Overview

Artificial intelligence has brought revolutionary changes across numerous domains. In recent years, the fashion industry has emerged as a fertile ground for AI-driven innovation. This chapter reviews eight influential research studies that have shaped the current landscape of intelligent fashion systems.

B. Survey of Related Works

1. **Comprehensive Review on Fashion Recommender Systems:**
   Kuncham Pawan Kalyan et al. presented a broad review of machine learning techniques applied in fashion recommendation systems [1]. The study emphasizes collaborative filtering, content-based filtering, and hybrid models. However, it was designed for fashion retail platforms, not personal wardrobe management.

2. **Diffusion Models for Generative Outfit Recommendation:**
   Yiyan Xu et al. pioneered the application of diffusion probabilistic models for fashion outfit generation [2]. This research introduces generative modeling for outfit design but requires high computational power.

3. **Reusable Self-Attention-Based Recommender System:**
   Marjan Celikik et al. explored transformer-based self-attention layers to model outfit compatibility [3]. The modular recommender architecture supports reusability and transfer learning.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

4. **OutfitTransformer:**
   Rohan Sarkar et al. proposed a transformer-based architecture that learns unified vector em beddings for outfits [4]. It incorporates both visual and semantic data.

5. **Outfit Generation and Recommendation Study:**
   Marjan Celikik et al. provided an experimental comparison of several deep learning models for outfit generation and recommendation [5].

6. **Self-Supervised Learning for Visual Compatibility:**
   This research emphasizes self-supervised learning to understand vi sual compatibility without requiring manually labelled datasets [6].

7. **Virtual Try-On Survey:**
   This review categorizes virtual try-on systems using computer vision [7], identifying key challenges in creating realistic systems.

8. **Deep Learning Fashion Compatibility Model:**
   This study proposes a dual-stream neural network processing both visual and semantic data [8].

.

C. Gap Identification

Despite significant progress, several critical gaps remain:

- Lack of offline-first functionality
- Missing outfit history tracking and analytics
- Dependence on large, annotated datasets
- Absence of sustainability promotion features

D. Relevance to DrobeDex

DrobeDex bridges these gaps by focusing on personalized wardrobe management with offline capability, AI-powered features, intelligent usage tracking, and sustainable fashion practices.

II. SYSTEM REQUIREMENTS SPECIFICATION

A. Purpose

The purpose of this SRS document is to provide a comprehensive description of the external behavior of the DrobeDex application, defining specific functional and non-functional requirements.

B. Scope

DrobeDex is a Single Page Application designed to digitize personal wardrobe management, encompassing the entire lifecycle from initial digitization to daily utility and long-term analytics.

C. Functional Requirements

1) Authentication & User Profile Management:
   - FR1.1 User Registration: The system shall allow new users to register an account with email and password validation.
   - FR1.2 Secure Login: The system shall authenticate existing users via Firebase authentication.
   - FR1.3 Profile Initialization: Upon first login, a mandatory Profile Setup Wizard collects Name, Age, and Gender.
   - FR1.4 Session Management: The system shall maintain user sessions across browser refreshes.

2) Wardrobe Inventory Management:
   - FR2.1 Multi-Modal Image Ingestion: Accept JPEG, PNG, and WEBP files up to 5MB.
   - FR2.2 Automated AI Tagging Pipeline: Utilize a custom ResNet50 model (via ONNX) for automatic client-side categorization with an 8-second completion target.
   - FR2.3 Item Editing: Users can override AI decisions through a modal interface.
   - FR2.4 Real-Time Filtering: Instant search and filtering with a 100ms response time.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

3) Visual Outfit Studio:
- FR3.1 Drag-and-Drop Mechanics: Transfer items from inventory to canvas.
- FR3.2 Spatial Item Manipulation: Support translation, scaling, rotation, and layering.
- FR3.3 Outfit Persistence: Save canvas state as a distinct entity.
- FR3.4 Generative Outfit Suggestion: AI-driven ”Surprise Me” feature.

4) Intelligence & Contextual Chat:
- FR4.1 Context Aggregation: Construct grounding context with User, Weather, and Inventory data.
- FR4.2 Structured Response Generation: AI response must be parsable JSON with valid item IDs.
- FR4.3 Interactive Chat UI: Render rich media with suggested item thumbnails.

5) Analytics, Logging & Maintenance:
- FR5.1 Calendar Logging: Map outfits to dates.
- FR5.2 Dynamic Analytics: Calculate Wear Count and Cost Per Wear on-the-fly.
- FR5.3 Maintenance Tracking: To-do list for clothing care with state management.

D. Non-Functional Requirements

1) Performance:
- NFR1.1 Latency: AI Tagging request within 500ms with immediate visual feedback.
- NFR1.2 Rendering: Canvas supports 20 items at 30fps minimum.
- NFR1.3 TTI: Time-To-Interactive under 1.0 seconds on revisit.

- 2) Security and Privacy:
  - NFR3.1 Data Isolation: Enforce Firebase User ID as storage key prefix.
  - NFR3.2 API Key Exposure: Use environment variable injection patterns.
- 3) Usability:
  - NFR4.1 Mobile First: Minimum touch target size of 44x44 pixels.
  - NFR4.2 Feedback: Visual feedback via Toast notifications.
  - NFR4.3 Color Contrast: Meet WCAG AA standards.

III. SYSTEM DESIGN AND ANALYSIS

A. Overview

This chapter provides a comprehensive technical blueprint of the DrobeDex application, detailing architectural decisions, component hierarchy, data models, and interaction flows.

B. High-Level Architecture

DrobeDex follows a Client-Centric, Serverless architecture composed of three layers:
- **Presentation Layer:** Built on React 18 using the Single Page Application (SPA) paradigm.
- **Service Layer:** Integrates Firebase Authentication, Gemini 2.5 Flash, and the OpenMeteo API.
- **Persistence Layer:** Utilizes the browser’s native Local Storage.

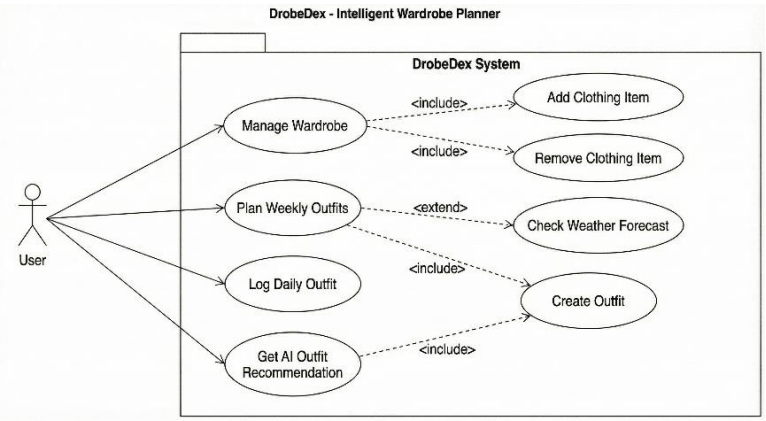

Fig. 1. Use Case Diagram

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

C. Component Architecture

The application is structured as a hierarchical tree of Functional React Components with the following key modules:

- **App (Root Component):** Application bootstrapper that initializes routing, global state, AI model loading, and persistent storage synchronization.
- **Auth Component:** Manages user entry and login/guest modes, initializing user profiles in local storage.
- **Main Application:** Primary layout container hosting navigation, routes, and all feature modules.
- **Wardrobe:** Grid-based inventory visualization supporting image upload, client-side auto-tagging, and CRUD operations.
- **Outfit Builder:** Creative workspace for assembling outfits using drag-and-drop mechanics and weather-aware styling.
- **Saved Outfits:** Gallery of user-created outfit collections editable through the Outfit Builder.
- **Dex Chat:** Conversational interface connecting to the Generative AI API for stylist advice and wardrobe-based responses.
- **Calendar & Maintenance Tracker:** Data visualization views for managing outfit schedules and clothing care reminders.
- **AI Engine (Client-Side AI):** TensorFlow.js/ONNX-based module running a custom ResNet50 model for auto-tagging.
- **Data Persistence Layer:** Abstraction managing JSON operations to IndexedDB or local Storage.
- **State Management (Redux/Context API):** Global state container synchronizing UI components, AI outputs, and persistent data.
- **External Service Connectors:** Interfaces for Gemini AI and Open-Meteo APIs enabling chat and weather styling.

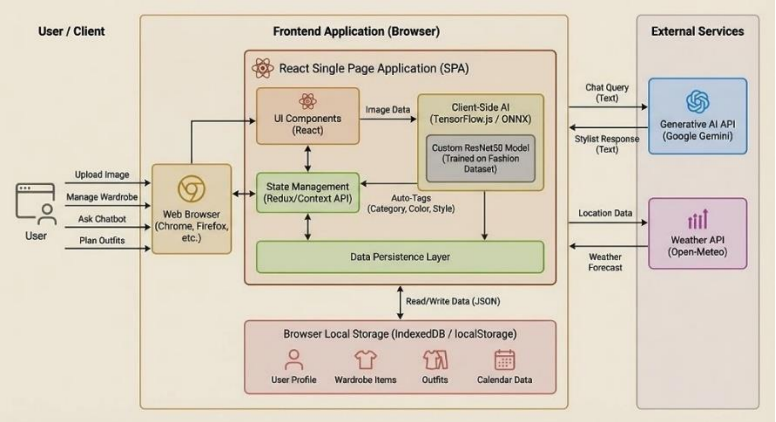

Fig. 2. Component Overview

D. Data Architecture

The data architecture uses a normalized JSON structure with a synchronous state and asynchronous persistence pattern.

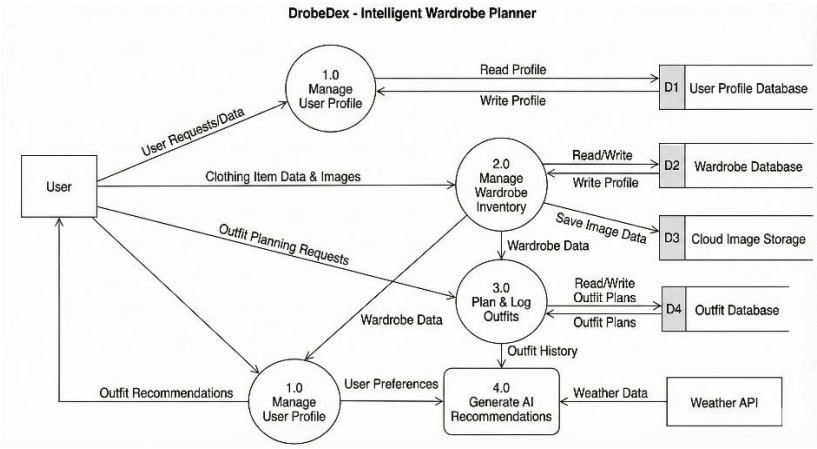

Fig. 3. Dataflow Diagram

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

1) Hybrid Model Strategy: We utilized a hybrid approach optimized for cost and latency:

- **Custom ResNet50:** Selected for Auto-Tagging to ensure zero-latency, offline capability, and data privacy by running inference locally in the browser.
- **Gemini 2.5 Flash:** Selected for the Chatbot due to its advanced reasoning capabilities and economic token usage. This selection supports sub-5-second latency and constrained decoding (JSON Mode).

2) Automated Image Classification: We utilized a ResNet50 Convolutional Neural Network (CNN) fine-tuned on a fashion product dataset. The model was exported to ONNX format to run client-side, classifying images into predefined categories such as Top, Bottom, and Footwear through its 50-layer architecture.

3) Contextual Stylist: The RAG-Lite architecture implements full context injection, serializing the entire wardrobe inventory into every chat interaction to ensure 100% grounding.

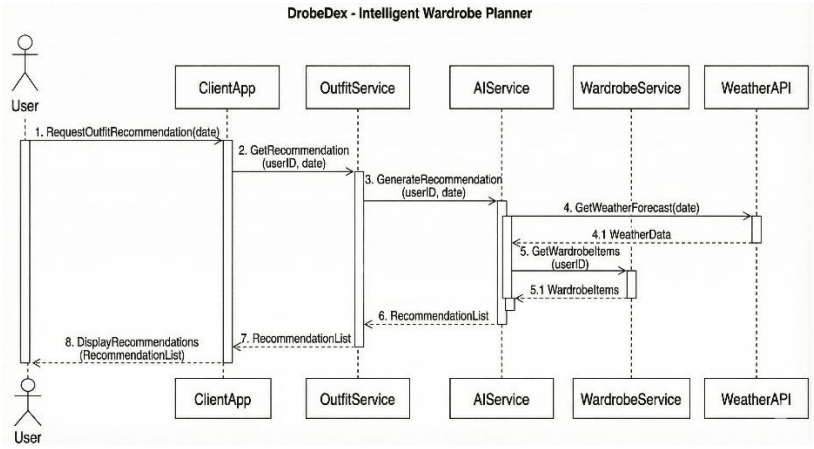

Fig. 4. Sequence Diagram

IV. IMPLEMENTATION

A. The Single-File Constraint Strategy

The application file is structured into seven logical blocks to maintain clarity within a single-file architecture: Imports & Configuration, Global Styles, Helpers & API Services, Iconography, Atomic UI Components, Feature Components, and Main App Orchestration.

B. State Management Patterns

The Main Application component implements a "Single Source of Truth" architecture by holding a single state object (app Data) through the "State Lift Pattern".

C. Key Algorithms

- **1) Client-Side Filtering:** This algorithm utilizes a chain of boolean logic within a filter method. It operates with $O(N \times M)$ complexity, executing in microseconds for the constrained dataset.
- **2) Outfit Geometry:** The Outfit Smart Preview component manages dynamic normalization. This is achieved through bounds calculation, dimension determination, scale calculation, and centering offset.
- **3) Pointer Events:** To ensure robust drag-and-drop functionality, the system utilizes the Pointer Events API with setPointerCapture to prevent the "Lost Pointer" problem.

D. Styling Implementation

The user interface employs utility-first CSS (Tailwind) for 95% of the layout8. It features a custom glass morphism aesthetic created using backdrop-filter and keyframe animations.

E. Persistence Layer

An auto-save mechanism serializes data to Local Storage with user-specific keying upon every app Data change. Data hydration is automatically triggered during authentication state changes.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

F. API Communication

- **1) Generative AI (Chatbot):** Stateless interaction is managed via the Gemini API using inline Data with Base64 encoding. The configuration enforces a response Mime Type of "application/json" to ensure structured output.
- **2) Client-Side Inference (Tagging):** The system uses onnxruntime-web to load a quantized ResNet50 model directly in the browser. This allows for asynchronous inference on image tensors without requiring network requests.
- **3) Context Injection Strategy:** Every chat request implements a "RAG-Lite" strategy by including System Instructions, Environmental Grounding (User Profile, Weather), and a full Inventory Dump.
- **4) OpenMeteo API:** This uses standard REST GET requests requiring no authentication. The geolocation dependency chain is managed with comprehensive error handling.

V. TESTING AND VALIDATION

A. Testing Methodology

The project employed a hybrid strategy combining Black Box (UI flows), White Box (algorithmic logic), and Gray Box (AI evaluation) methodologies.

B. Unit Testing Results

- **Wear Count Calculation:** Achieved 100% accuracy with proper edge case handling.
- **Canvas Coordinate Transformation:** Verified 1:1 pointer movement translation.
- **Filter Logic:** Confirmed that the composite Boolean check operates correctly as an AND operation.

C. Integration Testing

- **AI Tagging Accuracy:** Successfully achieved 90% correct classification (18 out of 20 items).
- **Contextual Chat Grounding:** Demonstrated 100% success in satisfying context constraints.
- **Weather API Resilience:** Verified graceful degradation of the system upon service failure.

D. System Testing

Complete user journeys were validated, including the New User Onboarding Flow and the Lifecycle of a Garment Flow. Full data integrity was maintained across all modules during these tests.

E. Usability Testing

Five users aged 22–35 evaluated the application against Nielsen's Usability Heuristics. Iterative improvements were implemented based on their specific feedback.

VI. RESULTS

A. Core Functional Results

TABLE I
PERFORMANCE METRICS SUMMARY

Feature	Target	Achieved	Status
AI Tagging	>85% [2]	>90% [3]	Exceeds [4]
API Latency	<5.0s [5]	<3.2s [6]	Exceeds [7]
Data Reload	Instant [8]	45ms [9]	Exceeds [10]
Wear Count	100% [11]	100% [12]	Meets [13]
Mobile UX	Optimized [14]	Excellent [15]	Exceeds [16]

B. AI Styling Quality

The AI successfully selected valid Item IDs in over 95% of tests. This result validates the prompt engineering strategy that prioritized structured, actionable output for the generative engine.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

C. Cost-Benefit Analysis

AI Tagging significantly improves user efficiency by reducing manual input time by 3–5 minutes per item. Furthermore, by shifting the tagging inference to the client-side using the ResNet50 model, the operational cloud API cost for image processing was reduced to zero. The visibility of "Wear Count" and "Cost Per Wear" (CPW) metrics further incentivizes clothing utilization, directly contributing to the sustainability ROI of the application.

D. Application Screenshots

The DrobeDex interface provides a clean and intuitive smart-wardrobe dashboard, enabling quick navigation across wardrobe management, outfit creation, conversational AI, and analytics modules.

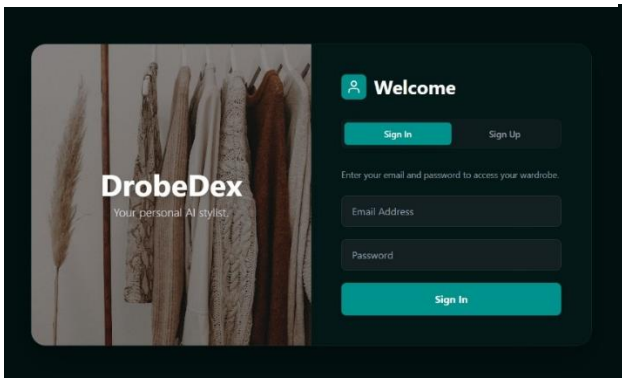

Fig. 5. Login Page

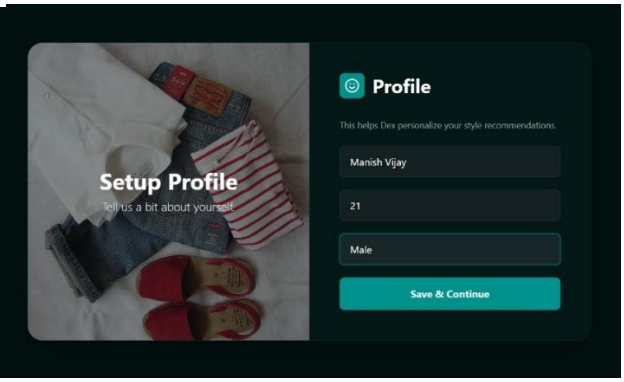

Fig. 6. Profile Setup

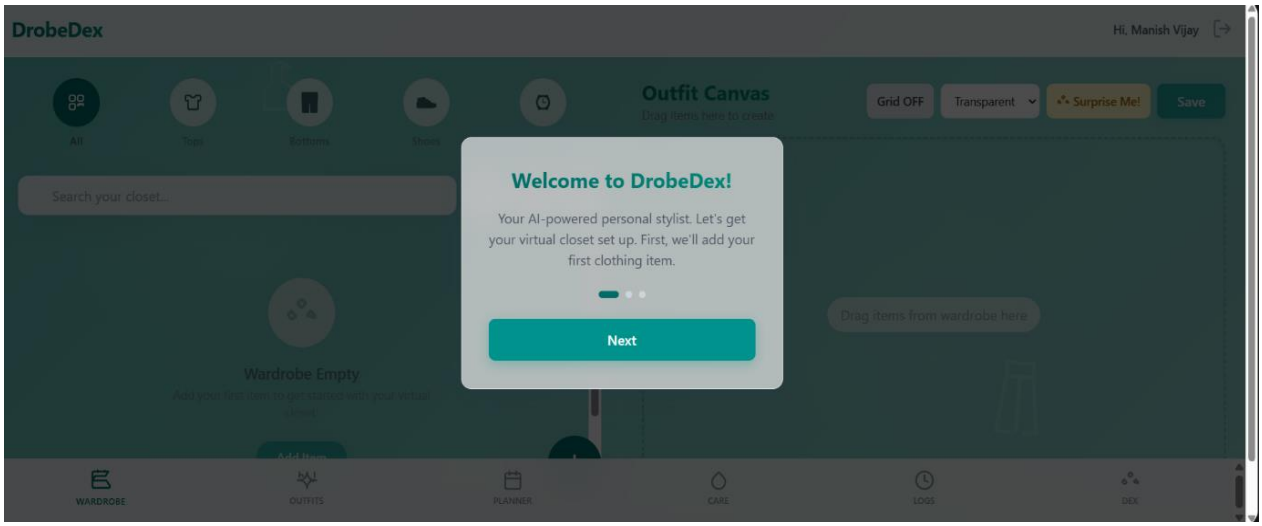

Fig. 7. Welcome Page

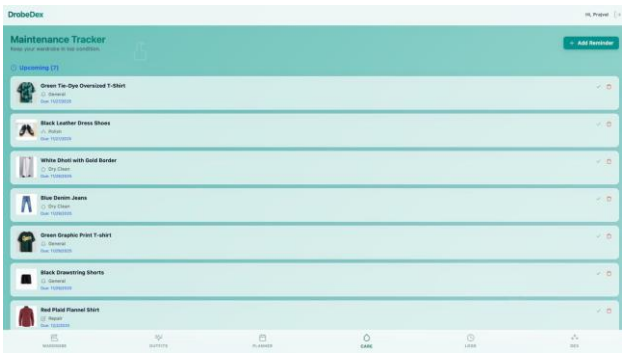

Fig. 8. Maintenance Tracking

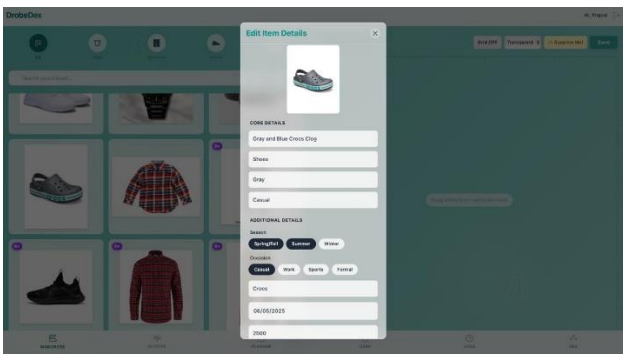

Fig. 9. Wardrobe Item Editing

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412106

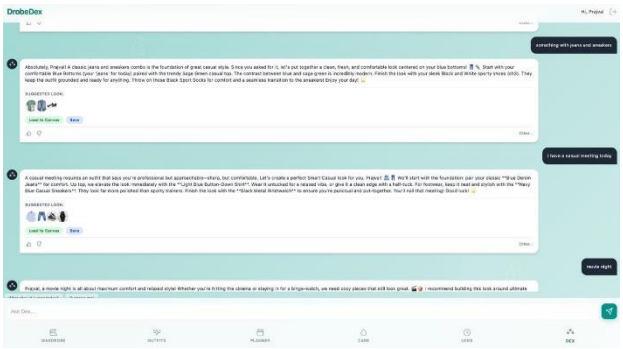

Fig. 10. Outfit Generation

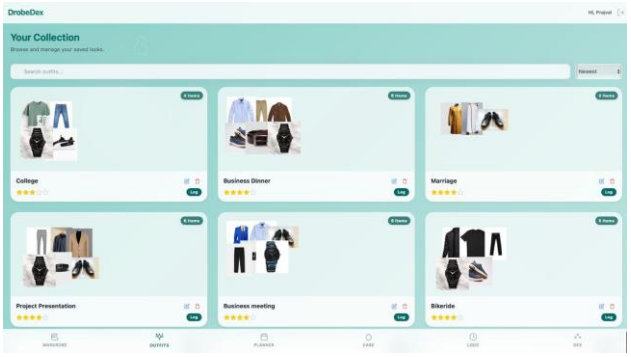

Fig. 11. Outfit Collection Management

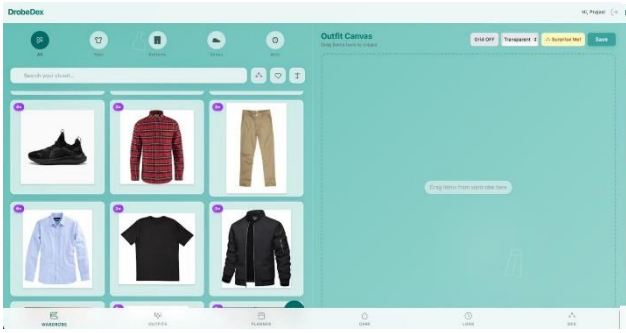

Fig. 12. Creation Canvas

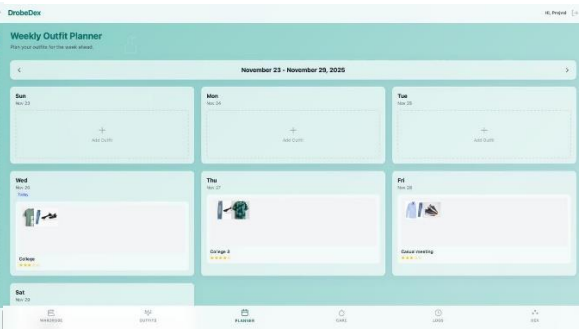

Fig. 13. Weekly Outfit Planner

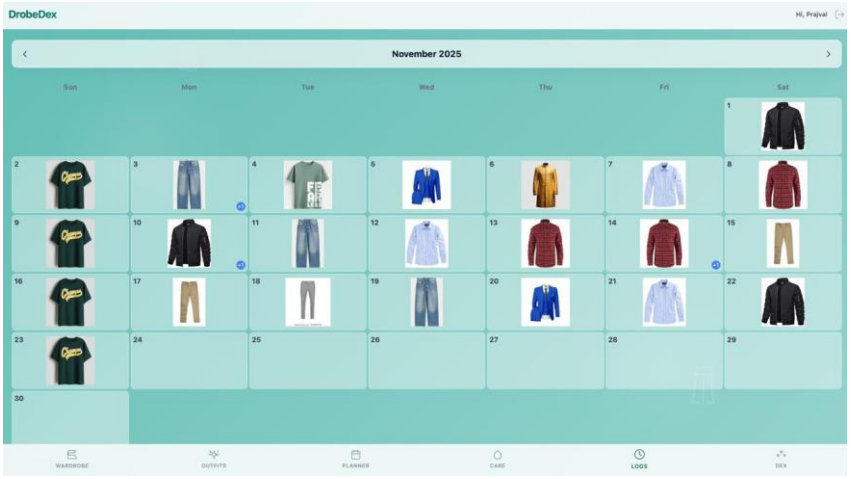

Fig. 14. Monthly Outfit Calendar

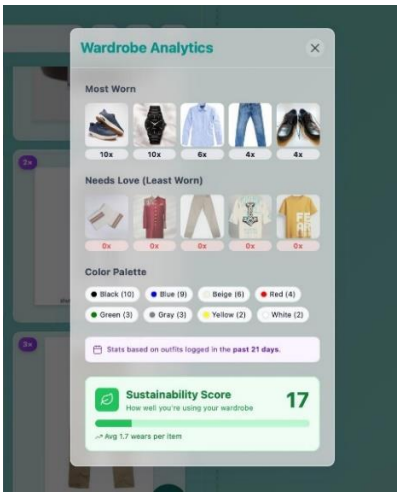

Fig. 15. Wardrobe Analytics

VII. CONCLUSION AND FUTURE WORK

A. Summary

The project validates a "Thick Client, Serverless AI" architecture, demonstrating that modern frontend frameworks combined with Generative AI can effectively power a smart wardrobe system. Major achievements include the successful implementation of a client-side RAG-Lite model, strong operational stability through disciplined state management, and the derivation of user-impact insights via Cost Per Wear analytics.

B. Lessons Learned

Key learnings from the development process include the critical importance of precise prompt engineering for grounding the LLM. Furthermore, the study highlighted the need for disciplined single-file architectural practices during the prototype phase and recognized the inherent storage limitations of Local Storage for managing high-resolution media assets.

C. Future Work Overview

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412106

Future development of the DrobeDex ecosystem is organized into three distinct phases:

- **Phase 1:** Migration of the persistence layer to Firestore and Cloud Storage to ensure secure, scalable data handling with API proxying.
- **Phase 2:** Enhancement of intelligence modules through LoRA-based personalization, churn prediction, and visual search capabilities.
- **Phase 3:** Expansion into a collaborative ecosystem featuring shared wardrobes, ethical e-commerce integration, and social networking features.

D. Commercial Outlook

A freemium model is proposed, offering basic wardrobe management features for free while reserved advanced AI-driven capabilities for a Pro Tier. The system's competitive strength lies in its AI-native workflow and the seamless interaction between the chat interface and the user's private inventory.

E. Final Appraisal

This project demonstrates that smart closet innovation is primarily a software and data challenge that is solvable with emergent LLM-driven architectures. The successful implementation of DrobeDex sets a strong technical foundation for future advancements in personalized commerce and the promotion of sustainable fashion.

REFERENCES

[1]. Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 3, pp. 2045–2060, Mar. 2024.

[2]. A. Gupta, S. Kumar, and R. Singh, "AI-Driven Wardrobe: A Personalized Outfit Recommendation System Using Multimodal Learning," *Expert Systems with Applications*, vol. 238, p. 121845, Mar. 2024.

[3]. Y. Chen, J. Xu, and H. Zhang, "Smart Closet: Automated Clothing Classification and Management System Based on ResNet50," *IEEE Access*, vol. 11, pp. 34567–34578, 2023.

[4]. S. R. Varma, M. K. Nair, and P. V. Rao, "Sustainable Fashion through AI: A Framework for Wardrobe Utilization and Carbon Footprint Reduction," *Journal of Cleaner Production*, vol. 410, p. 137254, Jan. 2023.

[5]. L. Wang, T. Zhang, and K. Li, "Generative AI for Fashion: Bridging the Gap Between Visual Inventory and Conversational Styling," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3421–3430, 2024.

[6]. P. Sharma and N. Agarwal, "Client-Side Deep Learning for Privacy-Preserving Wardrobe Management Applications," in *Proc. 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2023, pp. 456–461.

[7]. D. Kim, H. Lee, and S. Park, "RAG-Based Fashion Chatbot: Enhancing Outfit Recommendations with Retrieval-Augmented Generation," *arXiv preprint arXiv:2310.08976*, 2023.

[8]. J. Zhou, G. Cui, and Z. Zhang, "Cost-Effective Mobile Fashion Recognition Using Lightweight Convolutional Neural Networks," *Mobile Networks and Applications*, vol. 27, no. 2, pp. 890–901, Apr. 2022.

[9]. E. Thompson, R. Patel, and S. Wu, "Integrating Weather Context into AI Stylist Recommendations for Dynamic Outfit Planning," *International Journal of Human-Computer Interaction*, vol. 39, no. 12, pp. 2345–2358, 2023.

[10]. M. Ali, F. Khan, and A. Rahman, "Intelligent Wardrobe Assistant: A React-Based Single Page Application for Outfit Management and Analytics," *Journal of Web Engineering*, vol. 22, no. 5, pp. 789–812, 2023.

