StyleSphere: AI-Driven Personalized Wardrobe Curation

Dr. Navita Rana
Department of Information Technology
Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi
navitarana13@gmail.com

Dhruv Saini
Department of Information Technology
Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi
dhruvsaini.2412@gmail.com

Aditya Mishra
Department of Information Technology
Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi
adityaa.mishra1012@gmail.com

Pragyan Sharma
Department of Information Technology
Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi
pragyan2804@gmail.com

***Abstract* – This paper introduces StyleSphere, a web-based application that enables users to generate a virtual closet through uploading images of clothing items. The app applies artificial intelligence in offering daily styling suggestions with a basis on color theory and fashion heuristics. Additionally, StyleSphere introduces a peer-to-peer rental marketplace whereby users are allowed to lend out their apparel to others with the inclusion of the majority of the rental fee going to the lender. The system is built using React, Tailwind CSS, Firebase, and Cloudinary for image storage. StyleSphere aims at making managing one's wardrobe easier, fashion more sustainable, and more engaging for users through its intelligent styling suggestions.**

***Index Terms* – Artificial intelligence, cloud computing, fashion recommendation, peer-to-peer marketplace, virtual closet**

INTRODUCTION

Choosing daily outfits and managing a wardrobe can become quite cumbersome with the advent of fast fashion and minimal closet space. StyleSphere solves this problem by introducing virtual closets and AI-powered outfit suggestions backed by a rental marketplace. The paper provides insight into the architecture, system implementation, and potential impact of StyleSphere on personal styling and sustainable fashion.

RELATED WORK

Previous works have focused on machine learning and image recognition to develop fashion recommendation systems. Applications like Cladwell and Smart Closet allow users to plan outfits, but no integrated rental features are incorporated. StyleSphere extends these by incorporating fashion theory and a community-driven marketplace.

SYSYTEM ARCHITECTURE

StyleSphere is composed of three core modules:

- Frontend: Built with React and styled using Tailwind CSS for responsive design.
- Backend: Firebase handles authentication, database management, and real-time updates.
- Image Storage: Cloudinary is used to store and retrieve clothing images efficiently.

The AI engine processes uploaded images, tags clothing types, and generates outfit combinations using color harmony principles and seasonal trends.

FEATURES AND FUNCTIONALITY

- Virtual Closet: Users upload images of their clothes, which are then categorized and stored.
- Outfit Suggestions: AI recommends daily outfits based on color theory, garment type, and user preferences.
- Rental Marketplace: Users can put up clothes on rent. Payments are collected in a secure manner, with the majority share going to the lender.
- User Dashboard: closet inventory, rental history, outfit logs.

AI STYLING ENGINE: DESIGN AND LOGIC

The AI styling engine of StyleSphere is based on the rule-based system that evaluates the combination of clothes based on color theory and fashion heuristic principles. The following details in clothing are considered:

- Color Harmony: Complementary, analogous, and triadic schemes are employed. For instance, a blue shirt can be complemented by orange trousers for a contrasting look or with green for a natural palette.
- Garment Layering: Performs seasonal garment layering, such as base layer, mid layer, and outerwear,

and avoids making unsuitable combinations, such as a formal shirt with athletic shorts.

- Occasion Matching: Tags outfits for casual, formal, or semi-formal use based on garment metadata and user preferences.
- Weather Adaptation: Future versions will integrate weather APIs to suggest temperature-appropriate outfits.

The engine scores the combinations using a scoring matrix and presents the top three suggestions to the user each day.

MARKETPLACE DESIGN AND REVENUE MODEL

The marketplace for rentals is focused on sustainable fashion and community. Some key features include:

- Listing System: Users can post clothing items, describing them, stating the price for rent, and putting up dates of availability.
- Secured Transaction: Firebase routes the payment, which always routes 80–90% of the rent to the lender and keeps the small percentage as the platform's fee.
- Trust and Ratings: It involves a rating and review system where each transaction builds trust among the users.
- Inventory Management: Items that have been marked as rented are removed from the closet view to avoid any styling conflicts.

This model encourages circular fashion and monetization of underused wardrobe items.

UI/UX DESIGN PRINCIPLES

StyleSphere’s interface is done with Tailwind CSS and React, focusing on:

- Minimalist Design: Clean layout with intuitive navigation for uploading, browsing, and renting.
- Responsive Components: Mobile-first design allows access across devices.
- Visual Feedback: Show real-time previews of items uploaded, outfit suggestions, and rental status.
- Accessibility: Testing color palettes for good contrast and readability; support for colorblind-friendly modes.

User testing showed that 85% of participants felt that the interface was "easy to use" and "visually appealing."

COLOR THEORY IN OUTFIT GENERATION

The outfit recommendation engine within StyleSphere relies on established color theory principles that guarantee visually appealing and psychologically harmonious combinations. The AI analyzes the dominant and secondary colors of each garment, using Cloudinary's image metadata and custom color extraction algorithms accordingly, to apply the following:

- Complementary Colors: These will be colors found directly opposite one another on the color wheel, like blue and orange, or red and green. The system utilizes such combinations in making suggestions for bold, high-contrast outfits that would be appropriate to wear in expressive or statement looks.
- Analogous Colors: These are colors next to one another on the color wheel, such as green, teal, and blue. StyleSphere recommends these for similar, soothing attire in either casual or professional settings.
- Triadic Colors: Colors that are equidistant, like red, yellow, and blue, will create bright but balanced ensembles. These combinations are ideal for creative or festive occasions.
- Monochromatic Schemes: Variants of one hue are used-for example, light blue, navy, and sky blue-which provide minimalist and classy styling; these are usually recommended in formal and business-casual wear.

AI assigns a color profile to each item and then uses an algorithm that scores compatibility across the garments the user uploads. Skin tone matching and user-specific color preferences will be included in future iterations for personalized suggestions.

FASHION THEORIES AND STYLING HEURISTICS

In conjunction with color theory, StyleSphere incorporates fashion heuristics, which generate context-aware and aesthetically balanced outfits.

- The Rule of Thirds: A principle borrowed from visual design and applied to outfit proportions. A cropped top for instance is 1/3, high-waisted trousers are 2/3, and together the silhouette works visually.
- Layering theory: A system where outfits are structured into base, middle, and outer layers. AI considers the time of year and temperature to recommend appropriate layering.
- Style Archetypes: Each garment is tagged with style categories such as casual, formal, streetwear, and business casual. This tag enables AI to provide stylistically coherent results throughout the outfit suggestion process.
- Occasion-Based Styling: The user selects an occasion, such as work, date, and gym, and the engine filters the combinations according to formality, comfort, and cultural norms.
- Body Shape Theory: In future versions, users can select body shape and get recommendations for an outfit that fits in a way that will help improve the body's proportions.

These heuristics are encoded into a rule-based engine that filters and ranks outfit combinations before presenting them to the user.

DATA MODELLING AND DATA SYSTEM

The backend architecture of StyleSphere is designed in such a way that it efficiently stores, retrieves, and processes information about clothes. Every uploaded item gets tagged with metadata, including:

- Color: Extracted using Cloudinary’s dominant color API.
- Type: Shirt, pants, dress, jacket, etc.
- Style: casual, formal, etc.
- Fabric: cotton, denim, wool, etc.
- Season: Summer, winter, transitional.

These tags are stored in Firebase Firestore and used for building outfit graphs: representing possible combinations as nodes and edges. Each edge is weighted by compatibility scores computed from color theory and fashion heuristics.

In addition, user preferences such as likes/dislikes and saved outfits are stored and used to train the recommendation engine. This can also be represented diagrammatically to better visualize the flow of tagging an image through to outfit generation.

DESKTOP DEPLOYMENT WITH ELECTRON

StyleSphere is packaged as a cross-platform desktop application using Electron. This allows users to natively access the virtual closet and rental marketplace on Windows, macOS, and Linux. This decision is primarily driven by the following:

- The Offline Access: Users can view their closet and see cached outfit suggestions when internet access is not available.
- Native File Access: Electron provides drag-and-drop upload of images right from the file system, facilitating usability.
- Consistent UI: Using Tailwind CSS means the styles are consistent in web and desktop environments.

Electron bundles React frontend and Firebase backend into one executable, using electron-builder for packaging and auto-updates. The app intends to remain lightweight by shifting all heavy computation, like complex AI logic, to the cloud services.

CROSS PLATFORM ARCHITECTURE

The StyleSphere architecture is modular and designed for portability:

- The Offline Frontend Layer: Shared React components are reused across web and desktop builds.
- Electron Main Process: handles window management, file system access, and IPC with the renderer.
- Renderer Process: This is where the React application runs, interacts via REST APIs with Firebase and Cloudinary.
- Build Pipeline: Webpack and Rollup bundle the assets, and some conditional logic is done for platform-specific features.

This architecture ensures feature updates in core logic or UI go all the way into the platforms

IMPLEMENTATION DETAILS

This module uses Cloudinary's API for compressing and tagging images. Firebase Firestore is used to store metadata and user interactions. The AI logic in this system is implemented by using rule-based algorithms to find color compatibility and the layering of garments. Future versions may integrate deep learning for style prediction.

TECHNICAL CHALLENGES AND SOLUTIONS

The following were the difficulties faced during development:

- Image Compression and Tagging: Cloudinary's auto-tagging sometimes misclassified garments. Manual override and tagging options were added.
- Firebase Sync Issues: Real-time updates resulted in race conditions at the time of simultaneous uploads. This was resolved using Firestore transactions and debounce logic.
- Outfit Suggestion Latency: Initial AI logic was slow for large closets. Caching and pre-computation of combinations resulted in a 60% performance gain. These solutions contributed to a smoother user experience and scalable architecture.

COMPARATIVE ANALYSIS

TABLE 1. COMPARATIVE ANALYSIS

Feature	**Applications**		
	Cladwell	**Smart Closet**	**Stylesphere**
Virtual Closet	Yes	Yes	Yes
AI Outfit Suggestion	No	Yes	Yes
Color Theory Integration	No	No	Yes
Rental Marketplace	No	No	Yes
Community Features	No	No	Planned

As shown in Table 1, Stylesphere stands out by combining styling intelligence with a rental company.

TESTING AND DEBGGING WORKFLOW

Both automated and manual testing strategies are incorporated into StyleSphere in the following ways:

- The Unit Testing: Jest is used to test the core logic regarding outfit scoring and color matching.
- Integration Testing: The testing of Firebase rules and interactions with the Cloudinary API is performed using mock environments.
- Electron Debugging: DevTools are enabled in development mode. Logs are captured by electron-log.

The application also includes a diagnostic dashboard for developers showing API latency, image upload success rates, and user engagement metrics.

EVALUATION

Initial testing with 20 users indicated high engagement with the outfit suggestion feature and positive feedback on the clothing-for-rent model. The users liked the intuitiveness of the UI and the ability to monetize their unused clothes.

SCALABILITY AND DEPLOYEMENT

StyleSphere is designed to scale with a growing user base and feature set.

- Cloudinary Optimization: Compression of images to cut down on bandwidth and caching to facilitate faster loading.
- Firebase Performance: Firestore real-time database allows for concurrent users and dynamic updates. Large closets are handled with indexing and pagination.

- CDN and Caching: Static assets are served via CDN, and outfit suggestions are cached to reduce computation time.
- ML Model Hosting: Future versions will be deploying models, such as TensorFlow Lite for on-device outfit generations to reduce loads on servers and latencies.

The architecture allowing horizontal scaling, container-ready to be deployed on platforms such as Vercel or Firebase Hosting.

INTEGRATION POSSIBILITES

To implement more personalized and useful features, StyleSphere would use integrations with third-party services:

- Weather APIs: Outfit suggestions will adapt to local weather conditions, such as rainwear and layering.
- Calendar Sync: Users can connect their calendars to receive occasion-based outfit suggestions.
- E-commerce APIs: Missing items in an outfit can be suggested from online stores, enabling seamless shopping.
- Social Sharing: It will allow users to share the outfit on social networks or inside the community of the app for feedback and inspiration.

These integrations will transform StyleSphere from a utility app into a lifestyle platform.

FUTURE ENHANCEMENTS AND RESEARCH DIRECTIONS

StyleSphere roadmap includes as follows:

- AI Personalization: Using reinforcement learning to adapt outfit suggestions based on user feedback.
- 3D Closet Visualization: Rendering garments in 3D using WebGL for immersive browsing.
- Sustainability Analytics: Tracking carbon savings from rentals vs. purchases.
- Fashion Trend Forecasting: Integrating social media and fashion blogs to predict emerging styles.

These features will position StyleSphere as a leader in intelligent fashion tech.

CONCLUSION AND FUTURE WORK

StyleSphere serves as a model example of AI and the cloud for advanced personal styling in support of sustainable fashion. Future work will focus on adding social features, expanding the fashion theory models covered, and allowing outfits to be shared across platforms.

ACKNOWLEDGEMENTS

The authors would like to thank the faculty of the Department of Information Technology at Dr. Akhilesh Das Gupta Institute of Professional Studies for their guidance and support throughout the development of StyleSphere. Special thanks go to Dr. Navita Rana, whose insights into user experience and AI integration were instrumental in shaping the project. Thanks also go to the beta testers who provided valuable feedback during the user study phase. This work was completed as part of the author's final year BTech major project.

REFERENCES

[1] M. Tohidi et al., “Getting the right design and the design right: Testing many is better than one,” in Proc. ACM-SIGCHI Conf. on Human Factors in Computing Syst. (CHI’06), 2006, pp. 1243–1252.

[2] J. H. Watt and S. A. van den Berg, Research Methods for Communication Science. Boston, MA: Allyn and Bacon, 1995.

[3] M. S. MacNealy, Strategies for Empirical Research in Writing. Boston, MA: Allyn and Bacon, 1999.

[4] S. Kleinmann, “The reciprocal relationship of workplace culture and review,” in Writing in the Workplace: New Research Perspectives, R. Spilka, Ed. Carbondale, IL: Southern Illinois University Press, 1993, pp. 56–70.

[5] K. St. Amant, “Virtual office communication protocols: A system for managing international virtual teams,” in Proc. IEEE Int. Professional Commun. Conf., 2005, pp. 703–717.

[6] N. Ching and M. L. Wong, “Color harmony in fashion design: A study of consumer perception,” J. Fashion Marketing and Management, vol. 22, no. 3, pp. 345–360, 2018.

[7] A. Norman, The Design of Everyday Things. New York, NY: Basic Books, 2013.

[8] Firebase Documentation. [Online]. Available: https://firebase.google.com/docs

[9] Cloudinary Documentation. [Online]. Available: https://cloudinary.com/documentation

[10] Tailwind CSS Documentation. [Online]. Available: https://tailwindcss.com/docs