# Forique Customer Platform Development Report: Architectural Blueprint for an AI-Augmented Imitation Jewelry Marketplace

## 1. Executive Summary and Strategic Vision

The global imitation and artificial jewelry market is undergoing a seismic shift, transitioning from unorganized, offline retail to sophisticated digital marketplaces. However, the current e-commerce landscape for jewelry remains plagued by fundamental friction points that suppress conversion rates and inflate return logistics costs. Customers struggle to gauge size and scale on mobile screens, lack confidence in how specific metal tones will complement their skin complexions, and miss the tactile, emotional connection inherent in traditional jewelry gifting. This comprehensive development report outlines the architectural and functional strategy for **Forique**, a next-generation B2C multi-brand marketplace designed to resolve these specific deficiencies through aggressive technological intervention.

Forique is not merely a catalog of products; it is conceptualized as an intelligent style companion that leverages **Agentic AI workflows**, **Augmented Reality (AR)**, and **Computer Vision** to bridge the "trust gap" in online jewelry retail.1 The platform will operate on a **Centralized Product Master with Decentralized Seller Inventory** model, a structural choice that ensures catalog consistency and data hygiene while empowering individual vendors—ranging from large manufacturers to artisanal craftspeople—to manage their logistics and pricing autonomously.3 This hybrid approach allows Forique to scale its SKU count rapidly without sacrificing the user experience to the chaos of duplicate listings often found in open marketplaces.

Furthermore, the development methodology for Forique represents a departure from traditional Agile practices. We will adopt the **Google Antigravity** agentic development paradigm. By utilizing context-aware coding agents and "Context Engineering," the development lifecycle will be accelerated and rigor-tested. Product Requirement Documents (PRDs) will evolve from static text files into dynamic, context-rich prompts that drive AI agents to generate, verify, and iterate on code "Artifacts," ensuring that the final build aligns precisely with the architectural intent.5 This report serves as the master blueprint for stakeholders, developers, and product managers, detailing every facet of the platform from the database schema to the pixel-level logic of the AR try-on experience.

## 2. Market Context and User Psychology

### 2.1 The Friction of Digital Jewelry Retail

The primary challenge in selling imitation jewelry online differs significantly from apparel or electronics. Jewelry is an intimate, aesthetic purchase where "fit" is measured not just in dimensions but in proportion and color harmony.

* **The Size Ambiguity:** A necklace length of 18 inches vs. 22 inches radically alters the look, yet static images fail to convey this scale on different neck types. Similarly, bangle sizing is a notorious source of returns, with customers unable to translate diameter millimeters into a comfortable fit for their specific hand shape.7
* **The Color Theory Deficit:** Consumers often hesitate between gold, silver, or rose gold finishes because they lack the professional knowledge of skin undertones (warm vs. cool). This hesitation leads to "cart abandonment" or "bracketing" (buying multiple options with the intent to return most), which destroys margin.8
* **The Gifting Disconnect:** Jewelry is a primary category for gifting, yet the digital gifting experience is often sterile—a generic text note printed on a packing slip. Forique aims to re-emotionalize this process through "Phygital" video integration, making the unboxing moment as impactful as the product itself.10

### 2.2 Target Audience and Personas

The architecture is designed to serve distinct user personas identified in the imitation jewelry sector:

* **The Trend-Conscious Gen Z:** Seeks affordable, fast-fashion pieces (e.g., oxidized silver, chunky chains). They demand high-speed browsing, visual search capabilities, and social validation features.
* **The Occasion Buyer:** Purchases heavy sets (Kundan, Polki) for weddings or festivals. This user requires high-trust features, detailed zoom, and assurance regarding plating quality and longevity.11
* **The Gifter:** Often a male user buying for a partner, or friends buying for each other. This persona relies heavily on "recommendation engines" and sizing tools to reduce the anxiety of buying the wrong item.

## 3. Platform Architecture and Technology Stack

To support the high-concurrency traffic characteristic of flash sales and festival seasons in India (e.g., Diwali, Dhanteras), Forique will utilize a robust, scalable hybrid microservices architecture. The frontend will be decoupled from the backend to ensure flexibility in deploying "first-to-market" features without disrupting core transactional stability.

### 3.1 Frontend Strategy: The React Native Advantage

For the mobile application, **React Native** has been selected as the framework of choice. This decision is driven by the need for a single codebase that deploys to both iOS and Android while maintaining near-native performance—a critical requirement for the camera-intensive AR Try-On and Computer Vision features.

* **Performance Optimization:** We will utilize the **Hermes** JavaScript engine to reduce app start-up time and memory footprint.
* **Native Modules:** For the Bangle Sizer and AR features, we will write native bridges (Swift/Kotlin) where necessary to access raw camera data and GPU processing power, ensuring that the visual overlay is lag-free and responsive.13
* **Web Parity:** The web frontend will be built on **Next.js** (React). This allows for extensive code sharing of business logic (e.g., cart calculations, validation rules) between web and mobile, while Next.js's Server-Side Rendering (SSR) capabilities ensure that the product catalog remains highly indexable by search engines—a vital channel for customer acquisition.15

### 3.2 Backend Microservices and Data Layer

The backend will be orchestrated using **Kubernetes**, managing distinct microservices that can scale independently.

* **Core Transactional Service:** Built on **Node.js**, handling real-time operations such as user session management, cart updates, and order processing. This service prioritizes high throughput and low latency.
* **AI/ML Computational Service:** A separate service cluster built on **Python (FastAPI)**. This environment is optimized for heavy computational tasks, specifically the Skin Tone Analysis (OpenCV/TensorFlow) and the Recommendation Engine (Vector Search). Segregating this ensures that heavy image processing tasks do not block the checkout flow.5
* **Data Persistence Strategy:**
  + **Relational Data (PostgreSQL):** Used for strictly structured data requiring ACID compliance, such as User Accounts, Financial Transactions, and Order History.
  + **Catalog Data (MongoDB):** The diverse nature of jewelry attributes (e.g., a ring has "Size" but no "Chain Length," while a necklace is the inverse) necessitates a flexible, document-oriented database. MongoDB allows for an **Entity-Attribute-Value (EAV)** pattern implementation without the performance overhead of complex SQL joins, enabling a dynamic and expansive product taxonomy.17

### 3.3 The "Centralized Product Master" Model

A critical architectural decision for Forique is the **Centralized Product Master with Decentralized Seller Inventory**. In traditional marketplaces, if five vendors sell the same "Gold Plated Hoop Earring," the customer sees five distinct listings, leading to search pollution and confusion.

* **The Golden Record:** Forique will maintain a master catalog of unique products (Golden Records).
* **Seller Mapping:** Vendors do not create new products; they map their inventory to the existing master record.
* **Implication:** When a customer views a product, the system dynamically determines *which* vendor to fulfill the order from based on the "Buy Box" logic (detailed in Section 6.4), ensuring the customer always gets the best price and fastest delivery without wading through duplicate listings.3

## 4. The Google Antigravity Development Paradigm

The development of Forique will pilot the **Google Antigravity** agentic development platform. This is a strategic shift from manual coding to "Context Engineering," where the role of the senior developer evolves into an architect who orchestrates AI agents to execute implementation details.

### 4.1 From PRD to Context Prompts

In the Antigravity workflow, the Product Requirement Document (PRD) is not a static PDF read by humans; it is a structured data input for the AI. We will structure our PRDs as **Context Engineering Prompts**—comprehensive directives that define the role, context, constraints, and required outputs for the Gemini 3 Pro agents.

**Structure of a Context-Engineered PRD:**

1. **Role Definition:** "Act as a Senior Computer Vision Engineer specializing in React Native."
2. **Context Loading:** "You are working on the Forique repository. The tech stack is React Native CLI, TypeScript, and Python FastAPI backend. We use the 'react-native-vision-camera' library for all camera inputs."
3. **Task Constraints:** "Implement the Bangle Sizer module. It must detect a circular reference object (coin). It must NOT use LiDAR as we support mid-range Android devices. It must return measurements in millimeters."
4. **Artifact Definitions:** "Generate three artifacts: A mermaid sequence diagram of the logic, the TypeScript component code, and a Jest test suite for the calculation logic".19

### 4.2 The "Artifact" Verification Loop

Instead of scrolling through logs, the engineering team will validate progress through **Artifacts** generated by the Antigravity agents.

* **Plan Artifact:** Before writing code, the agent generates a markdown file outlining the proposed file structure and logic flow. This allows the architect to catch logic errors (e.g., "You missed the error state for low light") before a single line of code is written.
* **Implementation Artifact:** The actual code changes, presented as clean diffs.
* **Walkthrough Artifact:** A unique feature of Antigravity is the agent's ability to "record" its work. The agent can generate a browser recording or a series of screenshots showing the feature functioning in a test environment, providing visual proof of completion.6

## 5. Core Customer Experience Modules (Standard Stack)

While innovation drives differentiation, the stability and fluidity of standard e-commerce modules drive the baseline conversion. The following modules form the operational backbone of the Forique Customer App.

### 5.1 Advanced Authentication and Profile Management

* **Multi-Modal Login:** Integration of mobile OTP login (the primary standard for the Indian market) alongside Google and Apple social login.
* **Biometric Security:** Biometric authentication (FaceID/TouchID) will be implemented not just for login, but as a security step for accessing sensitive areas like the "Refund Wallet" and saved payment methods, enhancing trust.1
* **The "Style Passport":** The user profile will evolve into a "Style Passport" that stores jewelry-specific anthropometric data:
  + **Skin Tone Hex Code:** Derived from the AI analysis.
  + **Wrist & Finger Measurements:** Saved from the sizing tools.
  + **Metal & Gem Preference:** Explicit preferences (e.g., "I only wear Silver") that influence the ranking algorithms throughout the app.

### 5.2 Dynamic Home Page and Discovery

The Home Page is not static; it is a dynamic canvas managed by a widget-based architecture in the Admin Panel.1

* **Context-Aware Banners:** The banner carousel will adapt based on the user's "Style Passport." A user identified with "Cool" undertones will see banners featuring Oxidized Silver and Platinum collections, while a "Warm" undertone user sees Antique Gold.
* **Algorithmic Trust Signals:** We will display "Trusted Brand" badges dynamically. These are not paid placements but earned badges based on a vendor's operational metrics (low return rate, high dispatch speed), transparently showing customers why a brand is trusted.

### 5.3 Advanced Product Catalog and Taxonomy

The complexity of imitation jewelry requires a deep, hierarchical taxonomy to enable precise filtering.

* **Attribute Granularity:**
  + **Base Material:** Differentiating between Brass (higher quality) and Alloy (cheaper).
  + **Plating Specification:** E.g., "1 Micron Gold Plated," "Matte Finish," "Rhodium."
  + **Stone Detailing:** Explicitly tagging "Kundan" (glass), "Polki" (uncut diamond lookalike), "CZ" (Cubic Zirconia), and "Meenakari" (Enamel work) allows for expert-level filtering.23
* **Visual Search Integration:** We will integrate a visual search module where users can upload a screenshot from Instagram or Pinterest. The backend (using vector embeddings) will search the catalog for visually similar items, solving the "I don't know what this style is called" problem.25

### 5.4 Cart, Checkout, and Logistics

* **Vendor-Split Cart:** Since inventory is decentralized, a single cart might contain items from a vendor in Jaipur and another in Mumbai. The UI must clearly separate these into "Shipment 1" and "Shipment 2," providing distinct delivery estimates for each to manage customer expectations.
* **Gift Wrapping Logic:** A toggle to "Make this a Gift" will trigger the specialized gifting workflow, hiding the price on the packing slip and enabling the Video Message feature.

## 6. Innovative "First-to-Market" Features: Deep Dive

To establish Forique as a market leader, we will implement three high-complexity features that leverage AI and Computer Vision. These features address specific friction points in the jewelry buying journey.

### 6.1 AI-Powered Skin Tone Analysis & Recommendation Engine

This feature solves the "suitability" friction. Users often struggle to determine if a specific gold tone or silver finish suits their complexion.

#### 6.1.1 Technical Implementation Architecture

The system utilizes a Python-based microservice using **OpenCV** for image processing and **K-Means Clustering** for precise color extraction.27

1. **Image Acquisition & Normalization:** The user captures a selfie. The system first validates lighting conditions (rejecting images that are too dark or overexposed) and uses face landmark detection (via dlib or MediaPipe) to isolate the cheek and jawline areas, avoiding "noise" from lips, eyes, or hair.
2. **Color Space Conversion (The YCbCr Advantage):** Most images are captured in RGB, which mixes brightness and color. We will convert the image to the **YCbCr** color space. This separates the luminance (Y) from the chrominance channels (Cb and Cr). Human skin tones cluster very tightly in the Cb-Cr plane regardless of race or lighting brightness, making this the most robust method for skin segmentation.28
3. **Clustering & Tone Extraction:** We apply K-Means clustering (with K=3) to the segmented skin pixels to find the dominant color value, filtering out highlights (specular reflection) and shadows.
4. **Monk Scale Mapping:** To ensure inclusivity and accuracy across diverse Indian skin tones, we will map the extracted color to the **Monk Skin Tone (MST) Scale** vectors rather than the limited Fitzpatrick scale. This provides a 10-point granular classification.30
5. **Recommendation Logic Engine:**
   * **Warm Undertones (Greenish veins, tans easily):** The engine boosts products tagged with Yellow Gold, Antique Gold, Copper, and Kundan.
   * **Cool Undertones (Bluish veins, burns easily):** The engine boosts Silver, Platinum, Rhodium, and White Gold.
   * **Neutral Undertones:** The engine suggests Rose Gold and dual-tone mixed metals.8

#### 6.1.2 Skin Tone Analysis Logic Flow (Mermaid Diagram)

Code snippet

flowchart TD  
 A[User Opens Camera] --> B  
 B --> C{Lighting Check}  
 C -- Poor --> A  
 C -- Good --> D  
 D --> E  
 E --> F  
 F --> G  
 G --> H[K-Means Clustering]  
 H --> I  
 I --> J  
 J --> K{Determine Undertone}  
 K -- Warm --> L[Activate 'Gold/Kundan' Filter]  
 K -- Cool --> M  
 K -- Neutral --> N  
 L & M & N --> O

### 6.2 Computer Vision Bangle & Ring Sizer

This feature addresses the "fit" friction, which is the single highest cause of returns in the jewelry category.

#### 6.2.1 The "Coin Reference" Algorithm

Standard "AR measurement" on mobile phones is often inaccurate because most devices lack LiDAR depth sensors. To solve this, Forique will implement a **Reference Object** methodology. We use a standard coin (e.g., the Indian ₹10 coin), which has a known, constant physical diameter (27mm), to calibrate the scale of the image.32

1. **Scene Setup:** The user places their hand on a flat surface and places a ₹10 coin next to their wrist.
2. **Object Detection:** The application uses react-native-vision-camera to feed frames to a local OpenCV instance. The system detects the circular contour of the coin (using Canny Edge Detection and Hough Circle Transform) and the edges of the wrist.34
3. Scale Calibration: The system calculates the Pixel-to-Metric Ratio:  
     
   $$Ratio = \frac{\text{Real Coin Diameter (mm)}}{\text{Measured Coin Diameter (pixels)}}$$
4. **Measurement & Mapping:** The system measures the pixel width of the wrist, multiplies it by the *Ratio* to get the millimeter width, and then maps this width to the standard Indian Bangle Size Chart (e.g., 2.4, 2.6, 2.8).35

### 6.3 Social Gifting with "Phygital" Video Messages

This feature emotionalizes the unboxing experience, creating a "Phygital" (Physical + Digital) bridge that mimics the personal touch of handing over a gift in person.

#### 6.3.1 Technical Workflow

1. **Checkout Trigger:** During checkout, the user toggles "Add Video Message" (a monetized add-on, e.g., ₹49).
2. **Media Capture:** Post-payment, a browser-based recorder (using HTML5 MediaStream API for broad compatibility) opens. The user records a message (max 60 seconds).
3. **Cloud Storage:** The video is compressed (FFmpeg) and uploaded to AWS S3. A unique, short-code URL is generated.
4. **Dynamic QR Generation:** This unique URL is encoded into a high-density QR code.
5. **Logistics Integration:** This is the critical physical link. When the Vendor generates the shipping label/manifest from the Vendor Panel, the system dynamically injects this specific QR code onto the packing slip layout in a designated "Gift Message" quadrant.
6. **The Unboxing:** The recipient receives the package. The packing slip invites them to "Scan for a special message." Scanning the QR code launches a branded web player with the video, creating a surprise emotional moment.36

### 6.4 Multi-Vendor Smart Buy Box

In a marketplace model, multiple sellers may list the same standard items (e.g., "Plain Gold Plated Chain"). The "Buy Box" algorithm determines which seller's offer is added to the cart when the user clicks "Buy Now."

#### 6.4.1 Algorithm Logic

The Forique Buy Box algorithm calculates a **Seller Score ($S$)** for each vendor listing the SKU, based on weighted parameters designed to optimize for customer satisfaction.38

The Formula:

$$S = (0.4 \times P\_n) + (0.3 \times D\_n) + (0.2 \times R\_n) + (0.1 \times I\_n)$$

Where:

* $P\_n$ (Price Score): Normalized score based on the "Landed Price" (Product Price + Shipping). Lower price = Higher score.
* $D\_n$ (Delivery Score): Based on the estimated delivery days to the user's pincode. Faster = Higher score.
* $R\_n$ (Rating Score): The vendor's aggregate feedback rating.
* $I\_n$ (Inventory Score): A binary boost for high-depth inventory (preventing stockouts).

**Tie-Breaker Logic:** If two vendors have equal scores (within a 1% margin), the system applies a **Round Robin** distribution logic to ensure fair play and prevent monopoly by a single large seller.40

#### 6.4.2 Buy Box Logic Flow (Mermaid Diagram)

Code snippet

flowchart TD  
 A[User Loads Product Page] --> B{Is Product Multi-Vendor?}  
 B -- No --> C  
 B -- Yes --> D  
 D --> E  
 E --> F  
   
 subgraph Scoring Logic  
 F --> G  
 F --> H  
 F --> I  
 F --> J  
 end  
   
 G & H & I & J --> K  
 K --> L  
 L --> M{Is Top Score Tied?}  
 M -- Yes --> N  
 M -- No --> O  
 N --> P  
 O --> P  
 P --> Q

## 7. Logistics and Vendor Management

The operational success of Forique depends on the seamless management of a decentralized inventory model.

### 7.1 Vendor Onboarding and QC

* **KYC & Verification:** Automated GST verification API integration to ensure only legal entities onboard.
* **Image QC AI:** To maintain aesthetic consistency, an AI layer (Google Vision API) will scan vendor-uploaded images during the listing process. It checks for:
  + **Background Quality:** Must be white/grey/neutral.
  + **Content Policy:** Rejects images with offensive content, watermarks from other platforms, or text overlays.1

### 7.2 Decentralized Order Routing

Forique routes orders; it does not fulfill them (initially).

* **Order Splitting:** If a customer orders Item A (Vendor X) and Item B (Vendor Y), the system internally splits this into two sub-orders. The customer sees a unified checkout but distinct tracking numbers.
* **SLA Monitoring:** The system tracks the "Time to Dispatch." If a vendor fails to scan the manifest within 24 hours, their "Delivery Score" ($D\_n$) is penalized, directly impacting their Buy Box win rate. This gamification forces operational efficiency.41

## 8. UX/UI Design System and Experience Philosophy

### 8.1 Visual Identity: "Luxury Minimalist"

The design language must balance the "trust" of a jewelry store with the "speed" of an app.

* **Typography:** A pairing of a high-contrast Serif font (e.g., *Playfair Display*) for headings to evoke elegance and tradition, with a clean Sans-Serif (e.g., *Lato* or *Inter*) for readability in specifications and pricing.
* **Color Palette:** Deep Emerald Green (#004d40) as the primary brand color to signify wealth and trust, accented with Matte Gold (#c5a059) for call-to-action buttons (CTAs) and success states.

### 8.2 Behavioral UX Strategies

* **Scarcity Cues:** Real-time inventory counters ("Only 2 left!") driven by the decentralized inventory data to induce urgency.
* **Visual Hierarchy:** The "Try On" and "Size Guide" buttons will be given equal visual weight to the "Add to Cart" button on the Product Detail Page (PDP). Data suggests that engagement with VTO tools increases conversion probability by up to 60%, so driving users into this funnel is a priority.7

## 9. Phase-Wise Development Roadmap

### Phase 1: The Foundation (Months 1-3)

**Focus:** Core Marketplace Transactional Loop.

* **Goal:** Launch a stable MVP where users can browse, buy, and track orders.
* **Deliverables:**
  + Auth & User Profile (Basic).
  + Centralized Product Master & Search.
  + Vendor Panel (Listing & Order Management).
  + Checkout with Payment Gateway Integration (Razorpay/Stripe).
  + Logistics Integration (Shiprocket/Delhivery).
* **Key Metric:** Successful Order Completion Rate.

### Phase 2: The "Smart" Layer (Months 4-6)

**Focus:** AI Differentiation & Personalization.

* **Goal:** Deploy the AI features to increase engagement and reduce decision fatigue.
* **Deliverables:**
  + AI Skin Tone Analysis Microservice.
  + Smart Buy Box Algorithm Implementation.
  + Video Message Gifting Workflow (QR generation on manifest).
  + Refund Wallet System.
* **Key Metric:** Average Session Duration & Add-to-Cart Ratio.

### Phase 3: The Immersive Experience (Months 7-9)

**Focus:** Visual Augmentation & Scale.

* **Goal:** Reduce returns through better sizing and visual confirmation.
* **Deliverables:**
  + Computer Vision Bangle Sizer (Coin Reference).
  + AR Virtual Try-On (Earrings/Necklaces) using DeepAR/Banuba SDK integration.
  + Visual Search (Vector Embeddings).
* **Key Metric:** Return Rate Reduction & Conversion Rate.

## 10. Conclusion and Recommendations

The Forique platform architecture outlined in this document represents a sophisticated response to the unique challenges of the imitation jewelry market. By integrating **Agentic AI** via Google Antigravity, we not only accelerate development but also ensure a higher standard of code quality through artifact-based verification. The functional feature set—specifically the **Skin Tone Recommendation Engine**, **Coin-Reference Sizer**, and **Social Gifting Logic**—creates a moat of utility and emotional connection that generic marketplaces cannot easily replicate.

**Strategic Recommendations:**

1. **Data Quality is Paramount:** The success of the AI recommendation engine depends entirely on the quality of the product metadata. Investment in a robust "Cataloging Team" or AI-assisted tagging tools to ensure every product has accurate attributes (e.g., "Antique," "Matte," "Kundan") is non-negotiable.
2. **MVP Focus:** While the AR features are exciting, the **Skin Tone Analysis** and **Buy Box** logic offer the highest immediate ROI with lower technical risk than full 3D AR. These should be prioritized in Phase 2.
3. **Vendor Education:** The "Decentralized Inventory" model requires vendors to be operationally disciplined. A comprehensive "Vendor Success Program" must be developed alongside the software to ensure they understand how dispatch speed affects their Buy Box visibility.

Forique is positioned to be more than a shopping app; it is a technology-driven fashion enabler that respects the nuance, tradition, and emotion of jewelry buying.

### **Table 1: Competitive Feature Matrix (Forique vs. Market Standards)**

| **Feature** | **Standard Marketplaces (Amazon/Flipkart)** | **Niche Jewelry Apps (Giva/Caratlane)** | **Forique (Proposed)** |
| --- | --- | --- | --- |
| **Inventory Model** | Marketplace (Mixed) | Inventory Led (Single Brand) | **Hybrid (Central Master + Decentralized Inventory)** |
| **Try-On Tech** | None / Basic 2D Overlay | AR (High End Only) | **AR VTO + CV Sizer (Coin Reference)** |
| **Personalization** | Browse History Based | Quiz Based | **AI Skin Tone Analysis (Computer Vision)** |
| **Gifting** | Gift Wrap / Text Note | Gift Wrap | **Video Message via Dynamic QR Code** |
| **Vendor Logic** | Buy Box (Price Only) | N/A (Single Seller) | **Smart Buy Box (Price + Speed + Quality)** |
| **Development** | Traditional Agile | Traditional Agile | **Agentic (Google Antigravity + Context Engineering)** |

### **Table 2: Key API Integrations & Tech Stack**

| **Service** | **Purpose** | **Provider/Library** | **Reference** |
| --- | --- | --- | --- |
| **Face Tracking** | AR Try-On & Skin Tone | Banuba SDK / DeepAR / OpenCV | 42 |
| **Camera Access** | Image Capture | React Native Vision Camera | 44 |
| **Payments** | Transactions | Razorpay / Stripe (Split Settlements) | - |
| **Maps/Location** | Address Validation | Google Maps API / MapmyIndia | - |
| **Cloud Storage** | Video Messages | AWS S3 / Google Cloud Storage | - |
| **Search** | Catalog Indexing | ElasticSearch / Algolia | 25 |
| **AI Processing** | Skin/Image Analysis | Python (FastAPI) + TensorFlow | 29 |
| **Development** | Agentic IDE | Google Antigravity | 5 |

*End of Report*

#### Works cited

1. Zeera\_Module\_Wise\_FRD.docx
2. AI Jewelry Design Trends 2025: Innovative Customization - Continental Bead Suppliers, accessed on January 8, 2026, <https://www.continentalbeadsuppliers.com/blogs/blog/ai-jewelry-design-trends-2025-innovative-tools-and-customization-techniques>
3. Centralized vs. Decentralized Inventory Management: Finding the Ideal Approach, accessed on January 8, 2026, <https://www.labmanager.com/centralized-vs-decentralized-inventory-management-finding-the-ideal-approach-33644>
4. Centralized vs. Decentralized Warehousing: Which Is Better for Your Business?, accessed on January 8, 2026, <https://www.warehousingexpress.com/blogs/centralized-vs-decentralized-warehousing-which-is-better-for-your-business>
5. How to Set Up and Use Google Antigravity - Codecademy, accessed on January 8, 2026, <https://www.codecademy.com/article/how-to-set-up-and-use-google-antigravity>
6. Build with Google Antigravity, our new agentic development platform, accessed on January 8, 2026, <https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
7. Revolutionizing Jewelry Shopping: Where to Find AI-Powered Virtual Try-On Experiences for Your Website - Belgium Webnet, accessed on January 8, 2026, <https://belgiumwebnet.com/blogs/ai-powered-virtual-jewelry-try-on-for-jewelry-website>
8. Color analysis for jewelry: choose what suits your skin tone - Studio Kroewe, accessed on January 8, 2026, <https://www.studiokroewe.com/blogs/nieuws/choosing-jewelry-with-color-analysis>
9. Jewelry Color Analysis Guide | Jennifer Fisher, accessed on January 8, 2026, <https://jenniferfisher.com/blog/jewelry-color-analysis>
10. Personalized Gifting | Add Video Messages to Gifts - Livstick, accessed on January 8, 2026, <https://www.livstick.com/our-solutions/solutions-for-gifting/>
11. Imitation Jewelry Industry in India [Trends, Challenges, Solutions] - Unicommerce, accessed on January 8, 2026, <https://unicommerce.com/blog/imitation-jewelry-industry-india-trends-challenges-solutions/>
12. India|APAC Costume Jewelry Market Analysis, Size, and Forecast 2025-2029 - Technavio, accessed on January 8, 2026, <https://www.technavio.com/report/costume-jewelry-market-industry-in-india-analysis>
13. How To Integrate Augmented Reality Features in a React Native App - Banuba, accessed on January 8, 2026, <https://www.banuba.com/blog/how-to-integrate-augmented-reality-features-in-a-react-native-app>
14. Jewellery Virtual Try On Solutions with Augmented Reality (AR) - Softices, accessed on January 8, 2026, <https://softices.com/blogs/ar-in-jewellery-shopping-apps>
15. Google Antigravity: From Beginner to Expert in 14 Minutes - YouTube, accessed on January 8, 2026, <https://www.youtube.com/watch?v=mbxq88TOxp4>
16. Building an AI-powered Fashion Recommendation App | Langflow | Low-code AI builder for agentic and RAG applications, accessed on January 8, 2026, <https://www.langflow.org/blog/building-an-ai-powered-fashion-recommendation-app>
17. Ecommerce Website Development For Jewelry: A Complete Guide | Codilar Technologies, accessed on January 8, 2026, <https://www.codilar.com/ecommerce-website-development-for-jewelry-a-complete-guide/>
18. E-Commerce Database Design: Managing Product Variants for Multi-Vendor Platforms Using the EAV Model - Amit Singh, accessed on January 8, 2026, <https://np4652.medium.com/e-commerce-database-design-managing-product-variants-for-multi-vendor-platforms-using-the-eav-01307e63b920>
19. My Agent System Looks Powerful but Is Just Industrial Trash - DEV Community, accessed on January 8, 2026, <https://dev.to/qtalen/my-agent-system-looks-powerful-but-is-just-industrial-trash-d10>
20. Effective context engineering for AI agents - Anthropic, accessed on January 8, 2026, <https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>
21. Tutorial : Getting Started with Google Antigravity | by Romin Irani - Medium, accessed on January 8, 2026, <https://medium.com/google-cloud/tutorial-getting-started-with-google-antigravity-b5cc74c103c2>
22. Artifacts - Google Antigravity Documentation, accessed on January 8, 2026, <https://antigravity.google/docs/artifacts>
23. Tips for listing jewelry - Amazon Seller Central, accessed on January 8, 2026, <https://sellercentral.amazon.com/help/hub/reference/external/G201827270?locale=en-US>
24. Class 14 - Nice Classification, accessed on January 8, 2026, <https://nclpub.wipo.int/enfr/?class_number=14&explanatory_notes>
25. Visual Search for Jewelry Brands: What You Need to Succeed - Syte.ai, accessed on January 8, 2026, <https://www.syte.ai/blog/visual-ai/jewelry-visual-search/>
26. Build an AI-powered outfit recommendation app with AlloyDB and serverless runtimes, accessed on January 8, 2026, <https://codelabs.developers.google.com/smart-stylist-app>
27. Skin Segmentation and Dominant Tone/Color Extraction - DataDrivenInvestor, accessed on January 8, 2026, <https://medium.datadriveninvestor.com/skin-segmentation-and-dominant-tone-color-extraction-fe158d24badf>
28. Color spaces in OpenCV, accessed on January 8, 2026, <https://opencv.org/blog/color-spaces-in-opencv/>
29. KasrAskari/Human-Skin-Detection: skin detection with OpenCV - GitHub, accessed on January 8, 2026, <https://github.com/KasrAskari/Human-Skin-Detection>
30. Beyond Fitzpatrick: automated artificial intelligence-based skin tone analysis in dermatological patients - PMC - PubMed Central, accessed on January 8, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12179258/>
31. Skin Tone Research @ Google AI, accessed on January 8, 2026, <https://skintone.google/>
32. Image Based Measurement of Ancient Coins - Computer Vision Lab, accessed on January 8, 2026, <https://cvl.tuwien.ac.at/wp-content/uploads/2014/12/caa09.pdf>
33. COMPUTER VISION PROJECT - Angelfire, accessed on January 8, 2026, <https://www.angelfire.com/vt2/project/>
34. Easy Coin Detection with Python and OpenCV - YouTube, accessed on January 8, 2026, <https://www.youtube.com/watch?v=_Coth4YESzk>
35. Measure Bangle Size Online, accessed on January 8, 2026, <https://www.perfectcorp.com/business/products/how-to-measure-bangle-size-online>
36. 5 Curated Gift Shops Adding Video Messages to QR Codes - VidDay, accessed on January 8, 2026, <https://www.vidday.com/post/5-curated-gift-shops-adding-video-messages-to-qr-codes>
37. QR Code Customer Gifts | Boost Engagement with Post Purchase Gifting - ChocoCraft, accessed on January 8, 2026, <https://chococraft.com/blogs/corporate-gifts/qr-code-post-purchase-gifts>
38. The Amazon Buy Box Algorithm: Explained - Brandwoven, Empowering Commerce, accessed on January 8, 2026, <https://gobrandwoven.com/resources/articles/the-amazon-buy-box-algorithm-explained-how-to-win-the-buy-box-on-amazon/>
39. Amazon Buy Box: Its Algorithm Explained and How-to Win Guide - AMZScout, accessed on January 8, 2026, <https://amzscout.net/blog/amazon-buy-box/>
40. How to win the buybox of online marketplaces - onQuality, accessed on January 8, 2026, <https://www.onquality.de/en/blog-en/demystifying-the-buy-box-everything-you-need-to-know-about-online-marketplace-buy-boxes/>
41. Crack the Code: 10 Ways to Win the Buy Box - SellerAmp, accessed on January 8, 2026, <https://selleramp.com/tips-and-tutorials/crack-the-code-10-ways-to-win-the-buy-box/>
42. Virtual Jewelry Try-on by Banuba: AR Solutions for Jewelry Brands, accessed on January 8, 2026, <https://www.banuba.com/virtual-try-on-jewelry>
43. Compare 8th Wall vs. DeepAR in 2025 - Slashdot, accessed on January 8, 2026, <https://slashdot.org/software/comparison/8th-Wall-vs-DeepAR/>
44. ShagunDubey1/Virtual-try-on - GitHub, accessed on January 8, 2026, <https://github.com/ShagunDubey1/Virtual-try-on>