CollectIQ

AI-powered trading card intelligence — real-time valuation, authenticity, and collector trust.

**Product Requirements Document (Hackathon Edition – v1.0)**Version 1.0  
Team: CollectIQ Project Team  
Date: October 14, 2025

# Table of Contents

1. Executive Summary
2. Market Opportunity & Proof
3. Product Vision & Scope
4. System Architecture & Data Flows
5. AWS Service Responsibility Matrix
6. API Contract Summaries
7. Frontend Specification
8. Backend Specification (Rekognition → Bedrock pipeline)
9. DevOps & Infrastructure Requirements
10. AWS Cost & Resource Model
11. Success Metrics & Beta KPIs
12. Risks, Dependencies & Roadmap
13. Appendices (Glossary + Key Metrics)

# 1. Executive Summary

CollectIQ is an AI-driven trading card intelligence platform that delivers real-time market valuation, authenticity assessment, and portfolio management for collectors and investors. This edition emphasizes rapid prototype delivery, functional orchestration of Rekognition→Bedrock, minimal AWS overhead, and a focus on delivering a working demo within hackathon constraints.

# 2. Market Opportunity & Proof

The trading card and collectibles ecosystem demonstrates consistent high growth, driven by alternative investment demand, nostalgia, and digital resale trends. Market valuations range from $7B–$15B for trading card games and ~$23B for sports cards by 2031. The global collectibles market exceeds $400B, with authentication services growing 13% annually, underscoring demand for trust-driven tools.

# 3. Product Vision & Scope

Vision: Build the world’s most trusted AI valuation and authenticity companion for trading card collectors.  
Scope: Deliver real-time valuation, authenticity verification via Rekognition + Bedrock, and secure user vault management. The system provides transparency, trust, and market intelligence for every collector level.

# 4. System Architecture & Data Flows

Architecture: AWS-native microservice design. Users upload card images (S3), which trigger Step Functions: Rekognition extracts visual features, Bedrock reasons over those signals for authenticity and pricing rationale. Results persist in DynamoDB and return via API Gateway.

Data Flow Summary:  
1. User uploads image → S3 (presigned URL)  
2. API Gateway → Lambda (trigger Step Functions)  
3. Step Functions orchestrates: RekognitionExtract → PricingAgent + AuthenticityAgent (Bedrock)  
4. Aggregator → DynamoDB, emits EventBridge event  
5. API → Frontend for visualization.

# 5. AWS Service Responsibility Matrix

| AWS Service | Responsibility | Owner |
| --- | --- | --- |
| Amazon S3 | Storage of uploaded card images | Backend |
| Amazon Rekognition | Feature extraction (OCR, borders, holo variance) | Backend |
| Amazon Bedrock | Authenticity & valuation reasoning | Backend |
| AWS Lambda | Compute for handlers, agents, and orchestration tasks | Backend |
| Amazon API Gateway | Public entrypoint with JWT authorizer | DevOps |
| AWS Step Functions | Multi-agent orchestration flow | DevOps |
| Amazon DynamoDB | Persistence layer for cards and valuations | Backend |
| Amazon EventBridge | Domain event bus | DevOps |
| Amazon Cognito | User authentication | DevOps |
| AWS Amplify Hosting | Frontend deployment | DevOps |
| Amazon CloudWatch/X-Ray | Monitoring and observability | DevOps |

# 6. API Contract Summaries

Endpoints (summary-level):

| Endpoint | Method | Description |
| --- | --- | --- |
| /upload/presign | POST | Generate presigned URL for image upload |
| /cards | POST | Create/update card metadata |
| /cards | GET | List all cards for authenticated user |
| /cards/{id} | GET | Retrieve card details |
| /cards/{id}/revalue | POST | Trigger revaluation workflow |
| /healthz | GET | System health check (unauthenticated) |

# 7. Frontend Specification

The Next.js 14 frontend (App Router) hosted on AWS Amplify Hosting manages user flows:  
• Login via Cognito Hosted UI  
• Upload card (camera or file)  
• Display AI valuation + authenticity confidence  
• Manage vault (add/remove/revalue)  
• Alerts and pricing trends visualizations  
Focus: responsive UI, accessibility, and 60 FPS animation parity across browsers.

# 8. Backend Specification (Rekognition → Bedrock pipeline)

Backend implemented as modular Lambdas in TypeScript. Step Functions workflow:  
1. RekognitionExtract Lambda → creates FeatureEnvelope (OCR, visual signals)  
2. PricingAgent Lambda → ingests & fuses comps  
3. AuthenticityAgent Lambda → invokes Bedrock for reasoning  
4. Aggregator Lambda → persists to DDB & emits events.

Auth: Cognito JWT validation (API Gateway Authorizer). Data: DynamoDB single-table design. Logging: CloudWatch structured JSON.

# 9. DevOps & Infrastructure Requirements

Terraform manages all environments (dev/prod). Pipelines:  
• Backend CI: lint → build → test → package → deploy.  
• Infra CI: terraform fmt → validate → plan → apply (approval).  
• Amplify auto-builds from main/dev branches.  
Security: least-privilege IAM, KMS encryption, Secrets Manager rotation.  
Monitoring: CloudWatch alarms + dashboards; budgets on AWS spend.

# 10. AWS Cost & Resource Model

Summary-level estimate for MVP scale:  
• S3 storage: <$5/month for 10k images.  
• Lambda invocations: ~$15–30/month for 100k calls.  
• Step Functions: <$10/month for 20k executions.  
• Rekognition API: ~$1.00 per 1,000 images (~$50 at 50k images).  
• Bedrock inference: ~$0.002–0.01/request (~$100/month at scale).  
• DynamoDB + CloudWatch + Amplify Hosting: ~$60/month combined.  
Total: ~$250–300/month MVP; scalable to $1–2K/month at moderate adoption.

# 11. Success Metrics & Beta KPIs

• DAU/MAU growth  
• Upload-to-valuation success rate  
• Authenticity flag accuracy  
• Price prediction variance vs real sales  
• Retention (7d/30d)  
• Subscription conversion  
• API latency and uptime  
• User satisfaction (NPS)

# 12. Risks, Dependencies & Roadmap

Risks:  
• Marketplace API dependency  
• Model drift (authenticity false negatives)  
• AWS cost variability  
• Limited hackathon time (for v1.0)  
Roadmap:  
• Phase 1: MVP (Hackathon demo)  
• Phase 2: Closed Beta (expanded sets)  
• Phase 3: Public launch + B2B API  
• Phase 4: Cross-TCG support and predictive analytics

# 13. Appendices (Glossary + Key Metrics)

Glossary:  
• FeatureEnvelope: structured visual signals extracted via Rekognition.  
• AuthenticityAgent: Bedrock-driven reasoning model.  
• PricingAgent: live comps aggregator.  
• DDB: DynamoDB.  
• MVP: Minimum Viable Product.

Key Metrics:  
• Accuracy of AI authenticity detection ≥ 90% on validated samples.  
• Average valuation latency ≤ 3 seconds.  
• Monthly active users ≥ 500 in closed beta.