CollectIQ – Project Specification

# 1. Introduction

CollectIQ is an AI-powered collector assistant that identifies, authenticates, and valuates Pokémon Trading Card Game (TCG) cards using AWS Bedrock, Rekognition, and related services. It transforms the collector experience by automating tedious manual tracking tasks, offering real-time market insights, and enabling secure cloud-based vault management.

The project originated from a growing demand in the TCG market for smarter valuation tools. With over 75 billion Pokémon cards printed and an estimated market value exceeding $14 billion, collectors need automation to handle authenticity checks, pricing fluctuations, and collection organization efficiently.

# 2. Objectives

• Develop a full-stack AI-powered web app capable of identifying and valuing Pokémon cards from images.

• Leverage AWS services (Bedrock, Rekognition, Lambda, DynamoDB, Step Functions) to build a scalable agentic architecture.

• Provide real-time and historical market pricing sourced from APIs such as eBay, TCGPlayer, and PriceCharting.

• Enable users to track and visualize their vault’s total value, rarity distribution, and performance trends.

• Offer both mock and live API modes for hackathon demonstration resilience.

# 3. System Architecture

The system architecture leverages AWS-managed services for compute, AI, and data persistence. The frontend is built in Next.js 14 using TypeScript and Tailwind CSS for styling, deployed via AWS Amplify. Backend logic runs on AWS Lambda functions orchestrated through Step Functions. Rekognition powers computer vision for card detection, while Bedrock provides LLM-based reasoning for market analysis and valuation summaries.

\*\*Key Services Used:\*\*

• Amazon Bedrock – LLM-based valuation summarization and natural language reasoning.

• Amazon Rekognition – Optical text and image analysis for card name, set, and rarity extraction.

• AWS Lambda – Serverless compute for individual business logic modules.

• AWS Step Functions – Workflow orchestration for Identify → Value → Summarize → Persist pipeline.

• Amazon DynamoDB – Primary database for users, cards, and price snapshots.

• Amazon S3 – Storage for uploaded card images and processed metadata.

• Amazon Cognito – Authentication and user identity management.

• Amazon CloudWatch – Metrics and observability dashboards.

• AWS EventBridge + SNS – Real-time price alerts and event notifications.

# 4. Data Model

The database design follows a single-table schema in DynamoDB for performance and simplicity. Each record stores structured data under partition and sort keys to group user-owned cards and track their price history.

\*\*Primary Table: CollectIQ\*\*

Partition Key (PK): USER#{userId}  
Sort Key (SK): CARD#{cardId} or PRICE#<ISO8601>

\*\*Attributes:\*\* cardId, userId, detectedName, set, rarity, conditionEstimate, holoType, imageS3KeyFront, imageS3KeyBack, valueEstimate, confidenceScore, sources, createdAt, updatedAt

\*\*Indexes:\*\*

• GSI1 – userId (for vault queries)  
• GSI2 – set#rarity (for analytical dashboards)  
• TTL – for temporary cached records

# 5. API Design

The backend exposes REST endpoints via Next.js API route handlers, designed for clarity and simplicity.

\*\*Endpoints:\*\*

• POST /api/cards/upload – Generates presigned S3 URLs for secure uploads.

• POST /api/cards/identify – Invokes Rekognition to extract text and detect card features.

• POST /api/cards/value – Queries pricing data sources or mock data fallback.

• POST /api/cards/summarize – Uses Bedrock to generate a valuation summary and confidence score.

• POST /api/cards/save – Persists the card entry to DynamoDB.

• GET /api/cards – Lists all cards in a user’s vault.

• GET /api/cards/:id – Retrieves a specific card and its valuation history.

# 6. AI and ML Integration

AI plays a critical role in CollectIQ’s intelligence loop. Rekognition handles object detection and OCR, while Bedrock models perform reasoning to synthesize market trends and valuation statements.

\*\*Bedrock System Prompt Example:\*\*

You are a precise Pokémon TCG pricing analyst. Given metadata and normalized market comps, produce a 2–3 sentence summary including the fair value, recent trend, and action recommendation.

\*\*Output Example:\*\*

Base Set Charizard (NM) fair value ≈ $276 based on 23 comps over the last 14 days (range $240–$320). Consider listing if you are price-sensitive; short-term trend is favorable. Confidence: 0.78

# 7. Mock Data and Offline Mode

CollectIQ includes mock datasets for Base Set and Scarlet & Violet series. These files simulate real-world pricing to ensure smooth demonstration without external API dependency. The mock JSON files are located under /data/mock-prices/.

# 8. Frontend Design

The user interface emphasizes simplicity and delight. Built using Next.js 14 App Router, shadcn/ui components, and Tailwind CSS, it enables fast uploads, clear result presentation, and real-time vault updates. AWS Amplify provides authentication, deployment, and hosting with CI/CD integration.

# 9. Security & Compliance

All uploads are signed via presigned URLs to prevent unauthorized access. Data is encrypted in transit (HTTPS) and at rest (AWS-managed KMS keys). User authentication is handled securely through Amazon Cognito. The system adheres to OWASP recommendations and avoids storing sensitive personally identifiable information (PII).

# 10. Scalability and Performance

The serverless design ensures horizontal scalability. AWS Lambda scales automatically based on concurrent uploads. DynamoDB supports millions of requests per second with low latency, and S3 provides virtually unlimited storage capacity.

# 11. Roadmap

• M1: Core MVP (Identify → Value → Save flow working end-to-end)

• M2: Add Cognito Auth and multi-user vaults

• M3: Integrate live eBay and TCGPlayer APIs

• M4: Introduce Bedrock-based price trend forecasting

• M5: Enable grading simulator (image centering + edge detection)

• M6: Launch public beta and community feedback program

# 12. Business Model & Market Fit

CollectIQ operates under a freemium model: users can track up to 50 cards for free, while premium subscribers gain unlimited tracking, advanced analytics, price alerts, and integration with Discord or eBay for listings. Potential revenue streams include affiliate partnerships, API licensing for grading companies, and enterprise dashboards for card marketplaces.

# 13. Conclusion

CollectIQ represents the next evolution of collector intelligence: merging AI, automation, and cloud computing into a seamless experience. By leveraging AWS’s robust AI and serverless ecosystem, it delivers both scalability and sophistication while remaining accessible to everyday collectors. The system’s modular design allows for rapid expansion to other franchises (Yu-Gi-Oh!, Magic, One Piece) and long-term integration into the broader collectibles economy.

# 14. Fake Detection & Authenticity Analysis

Authenticity verification is a core feature of CollectIQ, designed to help collectors detect counterfeit Pokémon cards. Fake cards are widespread in the global market, especially high-value ones like Charizard, Lugia, or Pikachu Illustrator editions. This subsystem integrates computer vision, statistical image analysis, and AI reasoning to identify and classify potential forgeries.

## 14.1 Overview

The authenticity detection pipeline operates alongside the identification process. Once an image is uploaded and processed by Rekognition, the fake detection module analyzes the visual and textual characteristics of the card image. Each card is assigned an authenticity score ranging from 0.0 (likely fake) to 1.0 (likely authentic).

## 14.2 Detection Techniques

1. Visual Fingerprinting (Rekognition + OpenCV):

• Extracts holographic pattern, border ratio, and logo alignment.  
 • Generates a perceptual hash (pHash) for each card image.  
 • Compares the pHash against reference authentic hashes stored in a secure dataset.  
 • Flags anomalies such as misaligned logos, incorrect font positioning, or border inconsistencies.

2. Text and Font Validation:

• OCR is performed using Rekognition to extract text blocks (card name, attacks, HP, move descriptions).  
 • Validates the font family and kerning against known authentic cards.  
 • Identifies linguistic anomalies — fake cards often use mistranslated text or inconsistent terminology.

3. Holographic Surface Analysis:

• Computes pixel variance and RGB scatter across holographic areas.  
 • Compares reflection intensity and noise levels with authentic card holograms.  
 • Detects over-saturated holo regions or missing reflective texture patterns indicative of counterfeit printing.

## 14.3 AI Judgment via Bedrock

Once statistical indicators (hash similarity, text match, holographic metrics) are computed, they are passed into an LLM-based decision layer running on AWS Bedrock. This model synthesizes the numeric signals and provides an overall authenticity score and human-readable rationale.

\*\*Prompt Example:\*\*

You are an authenticity analyst for Pokémon TCG cards. Given numeric confidence scores from Rekognition and hash analysis, estimate whether the card is genuine. Output a probability and rationale. Format:  
{ authenticityScore: float, fakeDetected: boolean, rationale: string }

\*\*Example Output:\*\*

{ authenticityScore: 0.93, fakeDetected: false, rationale: 'Text font and holo patterns match authentic sample set with >90% similarity.' }

## 14.4 Data Model Extensions

The DynamoDB table is extended with additional attributes to store authenticity data. Example schema:

{  
 "cardId": "uuid",  
 "authenticityScore": 0.92,  
 "fakeDetected": false,  
 "visualHashConfidence": 0.94,  
 "textMatchConfidence": 0.89,  
 "holoPatternConfidence": 0.87,  
 "rationale": "Font, border, and holo texture match authentic sample.",  
 "verifiedByAI": true  
}

## 14.5 Feedback Loop & Continuous Learning

Users can flag results when the authenticity assessment appears incorrect. These feedback reports are stored and used to refine the authenticity model. Over time, this system learns from community input to improve detection accuracy. Each new confirmed authentic or fake sample enhances the reference dataset, strengthening model reliability.

## 14.6 Security and Privacy

To maintain dataset integrity, authentic reference hashes are stored in a private S3 bucket with IAM-based access controls. Only anonymized feature vectors are used for model training. This ensures no copyrighted images are redistributed and user-uploaded images remain confidential.

## 14.7 Future Enhancements

• Incorporate multi-angle holographic scanning via smartphone ARKit/ARCore sensors.

• Introduce blockchain-based authenticity certificates using AWS QLDB for immutable provenance tracking.

• Partner with PSA or CGC grading APIs for hybrid verification workflows.

• Develop an on-device inference model for offline fake detection during conventions or trade events.