Product Portfolio: Kofoworola Harris

Summary

I have worked across AI products, mobile apps and web platforms. I have shaped roadmaps, managed backlogs and improved delivery by using data, user research and simple feedback loops. I have improved model accuracy, raised user trust and guided cross-team delivery from idea to launch. These case studies show how I manage the full product lifecycle, solve real user problems and ship solutions at pace.

CASE STUDY 1 – AI Product: Improving Language Accuracy for Duolingo-Style App

Problem Statement

Customers were upset because the AI tutor kept making spelling errors. It damaged user trust. Complaints rose. Retention dipped. The team needed a way to tighten accuracy and restore confidence.

Research and Insights

I have obtained insight by running interviews with active learners. Many said the errors made them doubt the lessons. I checked analytics and saw a spike in “incorrect suggestion” reports. I reviewed the LLM outputs and spotted two root causes:

- Training data had inconsistent regional spelling.
- The model lacked confidence scoring for words that needed higher certainty.

Customer Feedback and Pain Points

Users said:

- “I don’t know who to trust.”
- “I feel like I am learning the wrong thing.”
- “I want corrections to be consistent.”

This shaped the direction of the fix.

Solution Design and Process Improvements

I designed a three-step solution:

1. **AI Spell-Check Layer**
   I added a rule-based spell-check layer before responses reached users.
2. **Confidence Score Thresholds**
   I have obtained model confidence data and set a threshold where the AI would avoid “guessing”.
3. **Correction Feedback Loop**
   I added a one-tap “This looks wrong” button that fed samples back to the training pipeline for review.

Testing Methodology (Including A/B Testing)

I tested with 200 real learners split in two groups.

- **Control group** received the old model.
- **Test group** used the improved model.

I tracked accuracy, reported errors and lesson completion.

Results showed a clear lift in accuracy and trust.

Iterations Based on Feedback

Learners wanted slower suggestions when the AI was unsure.
I adjusted the threshold and added an "Are you sure?" check when confidence dropped.

Launch Strategy and Communication Plan

I rolled it out in two phases:

- Soft launch for heavy learners.
- Full launch with a short in-app message explaining how accuracy had been improved.

This restored trust and cut spelling-related complaints.

CASE STUDY 2 – Mobile App: Illicit Code Scanner (Android → iOS Expansion)

Problem Statement

The app scanned installations for illicit code but worked only on Android. iOS users kept requesting support. This limited adoption and slowed growth.

Research on Cross-Platform Security Challenges

I reviewed platform restrictions. iOS blocks deep system access. Android allows broader scanning. The main challenge was designing a safe, compliant method that still protected users.

Customer Feedback and Pain Points

Many users felt left out. Some said:

- "I can't recommend this to friends because they're on iPhone."
- "I want equal protection across both devices."

This showed the clear demand for parity.

Solution Design and iOS Process Improvements

I shaped an approach that used:

- Secure file-based scanning instead of system-wide scanning.
- On-device ML models to detect risky behaviour patterns.
- Apple-approved APIs to analyse installation metadata.

This matched iOS rules while still giving users protection.

Testing and A/B Experiments

To validate adoption:

- Group A: Android users using current product
- Group B: Early iOS testers

I checked activation, scanning frequency and trust ratings.
iOS adoption grew faster than expected.

Launch Strategy

I prepared a simple rollout:

- Closed beta with 500 iOS testers.
- Public launch timed with App Store promotion.
- Message across both platforms explaining “Now on iOS”.

It raised the app’s reach and repositioned the product as cross-platform and safer.

CASE STUDY 3 – Website Feature: Brand Protection Tool (Scam Site + Sentiment Monitoring)

Problem Statement

The tool scanned websites for scam copies and tracked social mentions. Bots were getting blocked by social platforms. Sentiment scoring was inconsistent because the AI struggled with sarcasm and short posts.

Analysis of Technical and Compliance Challenges

I reviewed the blocks and found two main issues:

- Social platforms tightened bot rules.
- Scraping raised compliance risks.

I designed a safer approach that used approved APIs and zero scraping.

Customer Feedback and Pain Points

Brands were worried about:

- Fake pages stealing customers.
- Negative posts spreading fast.
- No early warning system.

They wanted faster alerts and clearer sentiment.

AI Integration and Workarounds

I integrated:

● An API-based fetch system that used official social platform endpoints.
● A sentiment model trained on short-form posts.
● A “context booster” that expanded short messages to reduce misclassification.

Testing and A/B Validation

I split site owners into two groups:

● Control tool with existing detection
● New tool with upgraded AI

I tracked alert accuracy and false positives.
The AI version showed clear improvement in detection and sentiment reliability.

Launch Strategy

I released it in stages:

● First to high-risk brands
● Then wider rollout with a dashboard tutorial
● Clear message: “Smarter alerts. Better protection.”