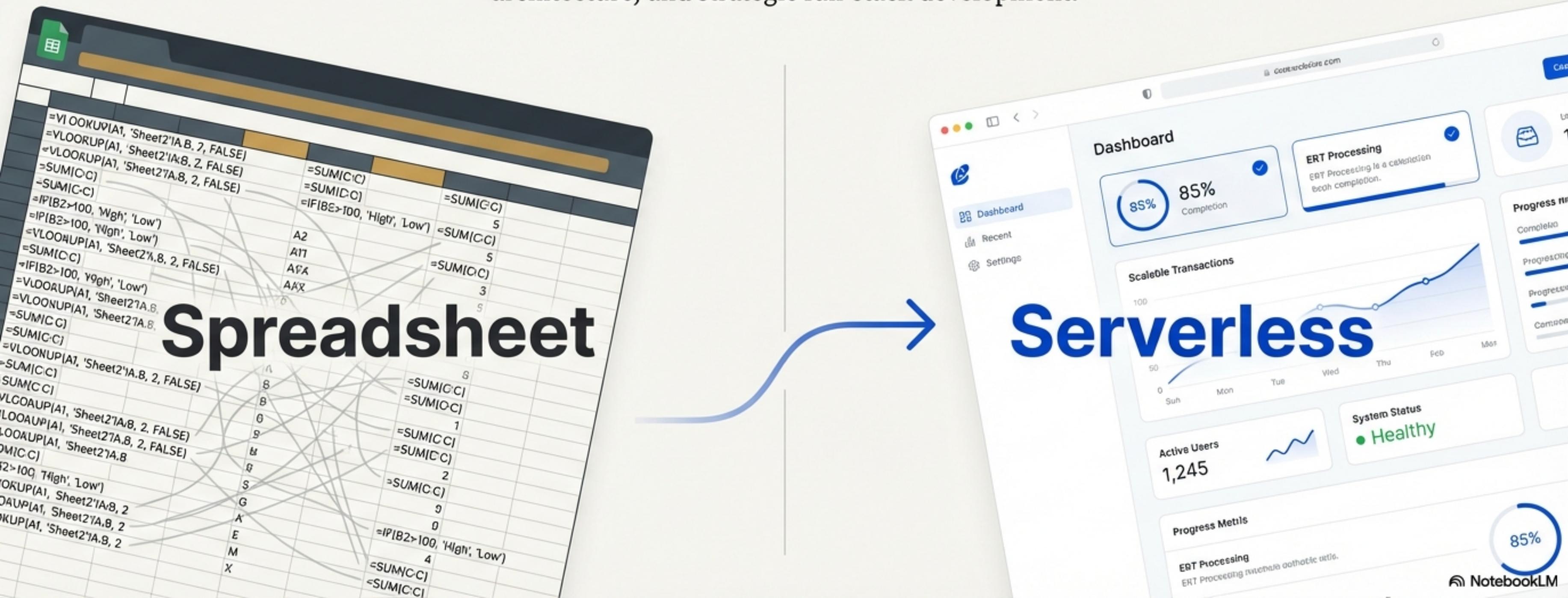


# From Spreadsheet to Serverless: The ERT Transformation Story

A case study in modernizing legacy logic, designing scalable cloud architecture, and strategic full-stack development.

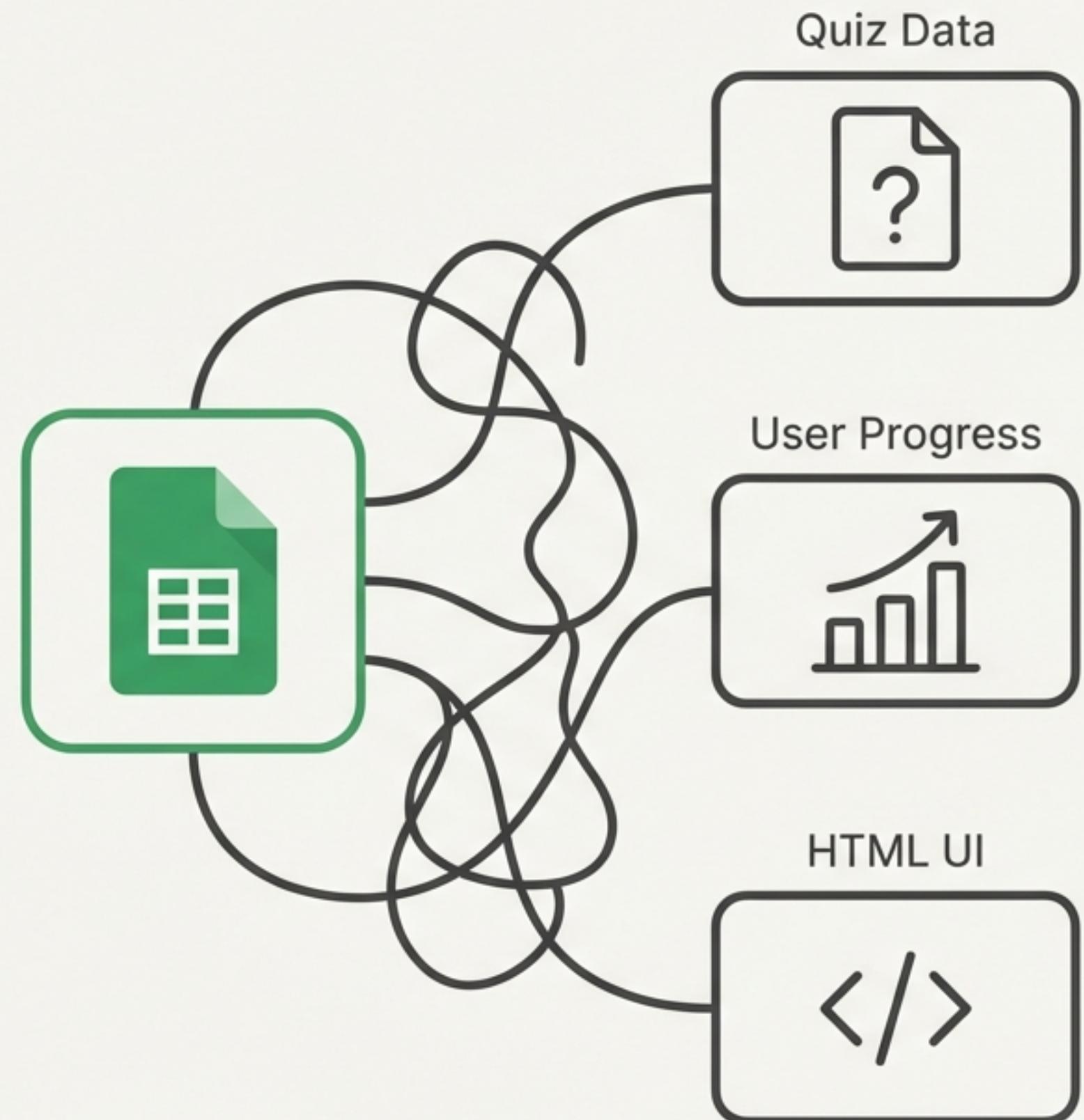


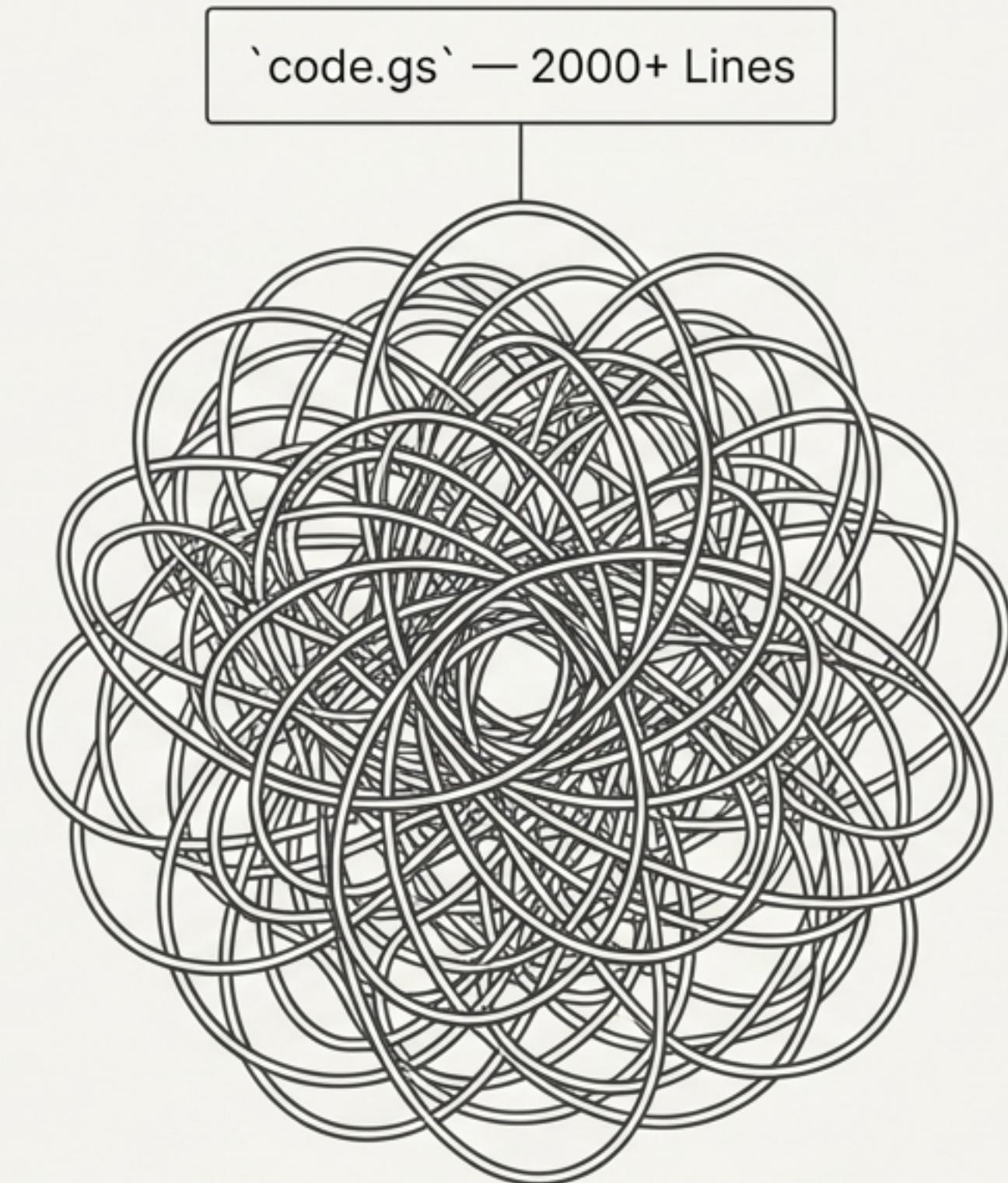
# The Origin: An Exam Readiness Tracker Born in Google Sheets

The project began as a personal tool for AWS certification prep. It quickly evolved into a complex system built entirely within Google Sheets, powered by a single, monolithic script.

## Key Features of the Original System:

- ☰ Quiz questions stored directly in spreadsheet cells.
- ☰ Manual progress tracking across multiple sheets.
- ☰ A basic UI rendered from HTML files within the script.





## Reaching the Limits of a Monolithic Script

The initial solution, while functional, became a significant bottleneck. A single 2000+ line Google Apps Script ('code.gs') was responsible for all logic, leading to severe scaling and maintenance challenges.



- **No User Isolation:** Lacked multi-user support, a critical feature for growth.



- **Difficult Maintenance:** The monolithic script was brittle and hard to debug or extend.



- **Scaling Infeasibility:** The architecture was not designed to handle more users, content, or features.

# The Strategic Pivot: Reverse-Engineering Logic with an AI Partner

*“Instead of starting from scratch, we asked Kiro to reverse-engineer the requirements from the existing functions. This preserved years of refined business logic.”*



2000+ Line Legacy Script



Kiro Partnership



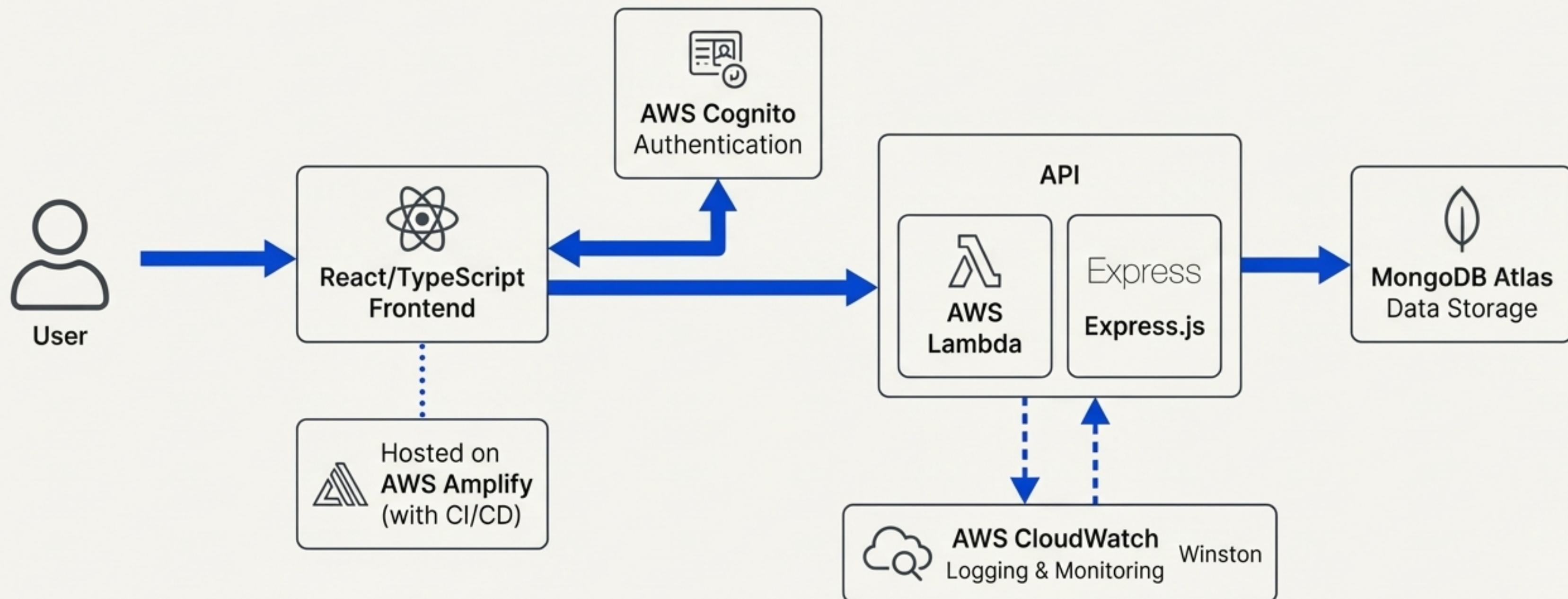
Modernization Roadmap

## Key Benefits:

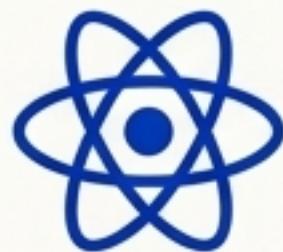
- Identified the core functionality that truly mattered.
- Revealed the architectural patterns hidden within the script.
- Provided a clear and validated roadmap for modernization.

# The Blueprint for a Scalable, Serverless Platform

The modernization roadmap led to the design of a robust, multi-user platform built on a serverless AWS architecture, designed for scalability, security, and maintainability.



# Engineering the Full Stack for a Modern Experience



## Frontend

React & TypeScript (for type safety and a modern UX).



## Hosting & DevOps

AWS Amplify (for seamless CI/CD and global distribution).



## Authentication

AWS Cognito (for secure, managed user identity).



## Database

MongoDB Atlas (for flexible, document-based data storage).



## API / Backend

AWS Lambda with Express.js (for infinitely scalable, serverless business logic).



## Monitoring

Winston & AWS CloudWatch (for robust, integrated logging and observability).

# The Development Journey: A Phased Transformation



## 1. Requirements Engineering

Extracting business rules from 2000 lines of script.



## 2. Data Migration

Moving quiz and user data from spreadsheet cells to MongoDB collections.



## 3. User Experience Design

Creating an intuitive, responsive interface from the ground up.



## 4. State & Security

Implementing proper session management and data isolation with Cognito.



## 5. Testing & Deployment

Establishing CI/CD pipelines via Amplify and comprehensive monitoring.



# From Brittle Cells to Flexible Documents: The Data Migration

**Before:** Spreadsheet Schema

Question_Text	Option_A	Option_B	Option_C	Correct_Answer
"What is serverless computing?"	"A server in your closet"	"Code that runs without provisioning servers"	"A type of cloud storage"	"Code that runs without provisioning servers"



**After:** MongoDB Document

```
{  
  "_id": "62a8b9c...",  
  "text": "What is serverless computing?",  
  "options": [  
    "A server in your closet",  
    "Code that runs without provisioning servers",  
    "A type of cloud storage"  
  ],  
  "answer": "Code that runs without provisioning servers",  
  "examId": "aws-saa-c03"  
}
```

# Key Technical and User-Facing Achievements



## Adaptive Learning

Smart question selection that intelligently skips mastered content.



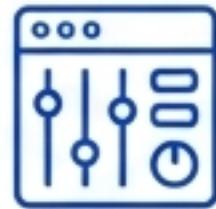
## Multi-Exam Support

Extensible architecture designed to support various certifications.



## Real-time Progress

Immediate user feedback with detailed performance analytics.



## Admin Capabilities

Full content management suite for questions and user administration.



## Scalable Infrastructure

Serverless design that scales automatically with user demand.



## Modern DevOps

Fully automated build and deployment pipelines with integrated monitoring.

# A Portfolio Piece Demonstrating End-to-End Expertise

This project is a comprehensive demonstration of the skills required to take a complex legacy system and transform it into a modern, scalable, and user-centric platform.



- **Legacy Modernization:** Successfully migrating complex, embedded business logic from a monolithic script.
- **Cloud Architecture:** Designing and implementing a production-ready, secure serverless solution on AWS.
- **Full-Stack Development:** Proven expertise across the entire stack: React/TypeScript frontend, Node.js (Lambda) backend, and MongoDB database.
- **AI Collaboration:** Effectively leveraging AI tools to accelerate development and de-risk requirements engineering.
- **Strategic Problem Solving:** Transforming a personal tool into a scalable, multi-tenant platform.



# The Next Chapter: Evolving the Platform for New Challenges

This platform serves as a strong foundation. The next phase of development will focus on adding features specifically designed to showcase advanced skills for targeted engineering roles.



## Targeting Full-Stack Roles

- Integrate new third-party APIs (e.g., for official documentation lookups).
- Develop more complex data reporting and analytics features.



## Targeting Cloud/DevOps Roles

- Implement Infrastructure as Code (IaC) using AWS CDK or Terraform.
- Enhance monitoring with custom dashboards and advanced alerting in CloudWatch.



## Targeting Frontend Roles

- Refactor state management with advanced libraries and add sophisticated UI animations.
- Conduct A/B testing on UI components using feature flags.