Pulse

Presented by Brandon Budhan, Daniel Sirias, Michael Campos, Peter Georgaklis, Trinity Dhillon, Zaira Garcia
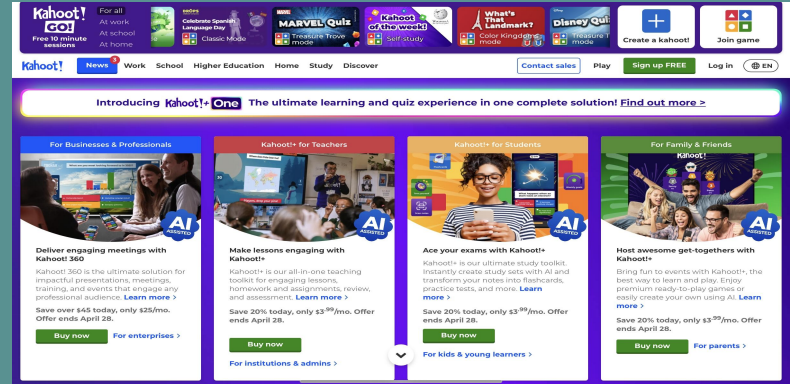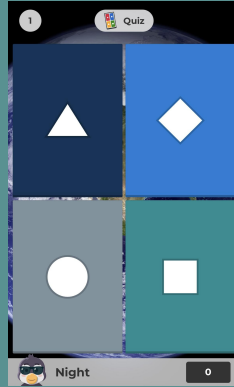
The Problem

❖ **Lectures not engaging enough!**

❖ **Lack of active engagement**

➢ **Active learning is better for retention than passive learning**

ME DURING A BORING LECTURE
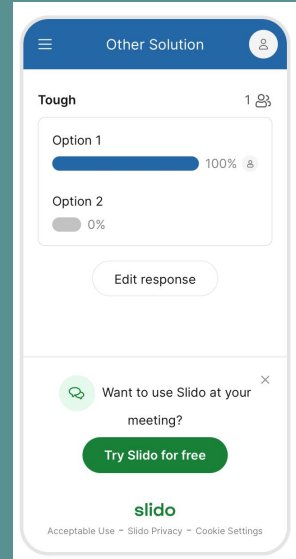makeameme.org

Existing Solution

Kahoot!

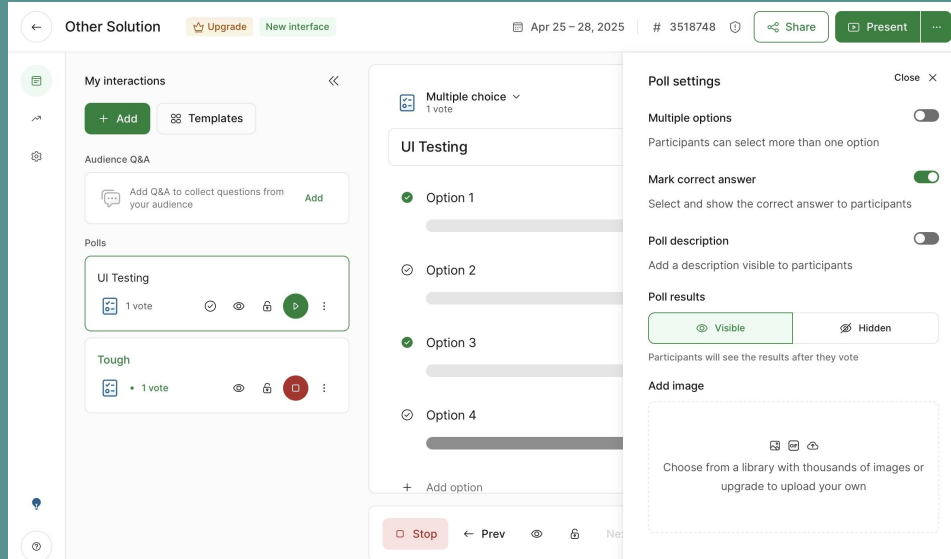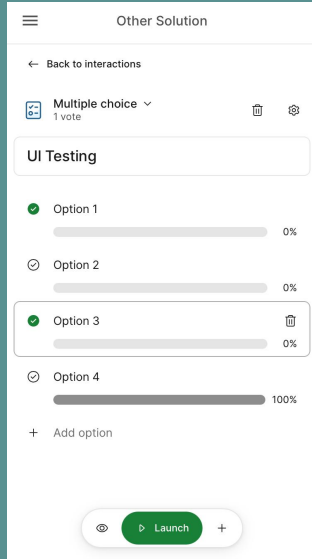❖ **Not appropriate for every occasion**
  ➤ Ex. College classes, board Meetings, Surveys, etc.
❖ **Unfriendly and clunky UI design**
  ➤ Users don't have text choices shown on their devices
  ➤ Point system not customizable

Existing Solution
Generic polling/Quiz websites

- ❖ **Unfriendly and clunky UI design**
- ❖ **Little Statistics**
- ❖ **Redundant Poll Maker!**
  - ➤ **Overcomplicated**

The Solution

Pulse

- ❖ **Single Page Web Application**
- ❖ **Versatile Design for every occasion**
  - ➢ **ex. College/High School Classes, Board Meetings, Surveys, etc.**
- ❖ **Easy Poll Creation!**
  - ➢ **Plus AI Assist**
  - ➢ **Comprehensive and Accessible Poll Results**
  - ➢ **View Participated/Hosted Polls**
  - ➢ **Custom Metrics for Graded Sessions and Submissions**
- ❖ **Polls can be used as study guide during and after poll session**
- ❖ **Accessibility features**
  - ➢ **Color-blind friendly palette for UI**
  - ➢ **Viewing of question/ image and answer options for participants**

Project Goals

Key Requirements:
- ❖ **Main Requirements:**
  - ➢ **Host a website that everyone can access in real time**
  - ➢ **Make Multiple Choice questions for all participants.**
  - ➢ **Output report with a students name, score, and list of incorrect answers.**
  - ➢ **Each student will be individually notified of their results.**

- ❖ **Enhancements:**
  - ➢ **AI Integration to generate questions from uploaded files.**
  - ➢ **User Experience (UX)**
    - ■ **Participant Login/Hosting in real-time polls**
    - ■ **Creating & Managing polls**
    - ■ **Viewing poll results.**

Project Architecture

❖ **Architecture Design**
  ➢ **C4 Model**

System-End Users [Hosts/Participates]

Hosts Polls
Participates in Polls
Creates Polls
Views Poll Results

Real Time Polling System

Authenticates Users with Google or Email/Password

Firebase Services

Stores all inforation about the users, accounts, polls, and results.

System-End Users [Hosts/Participates]

Visits https://pulsecheck-7cf2b.web.app/ using [HTTPS]

Hosts Polls
Participates in Polls
Creates Poll
View Poll Results

Web Application
[Container: Firebase Hosting]
Delivers static content and the single page real-time polling web application.

Delivers to the user's web browser

Single-Page Web Application
[Container: Typescript React, Material-UI]
Provides poll functions to users via their web browser.

Firebase Authentication
[Container: Firebase]
Allows users to login with Google or Email/Password

Makes API call to

Makes function calls to

API Store
[Container: Firestore Decorator]
Facade Design Pattern: Abstracts Firebase SDK to create, retrieve, update, delete data in Firestore

Reads from and writes to

Reads from and writes to

Firestore No SQL Database
[Container: FireStore]
Stores user, polls/questions, and poll sessions, poll results data.

Real Time Polling System

Project Design

- ❖ Software Tools
  - ➤ Github Projects
    - ■ Task Management With Categorization:
      - ● TODO, IN-PROGRESS, DONE
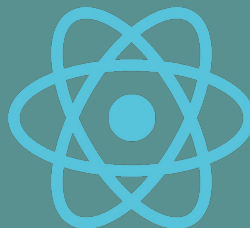   - ■ Assignees
  - ➤ Hardware: None
  - ➤ Software: Firebase
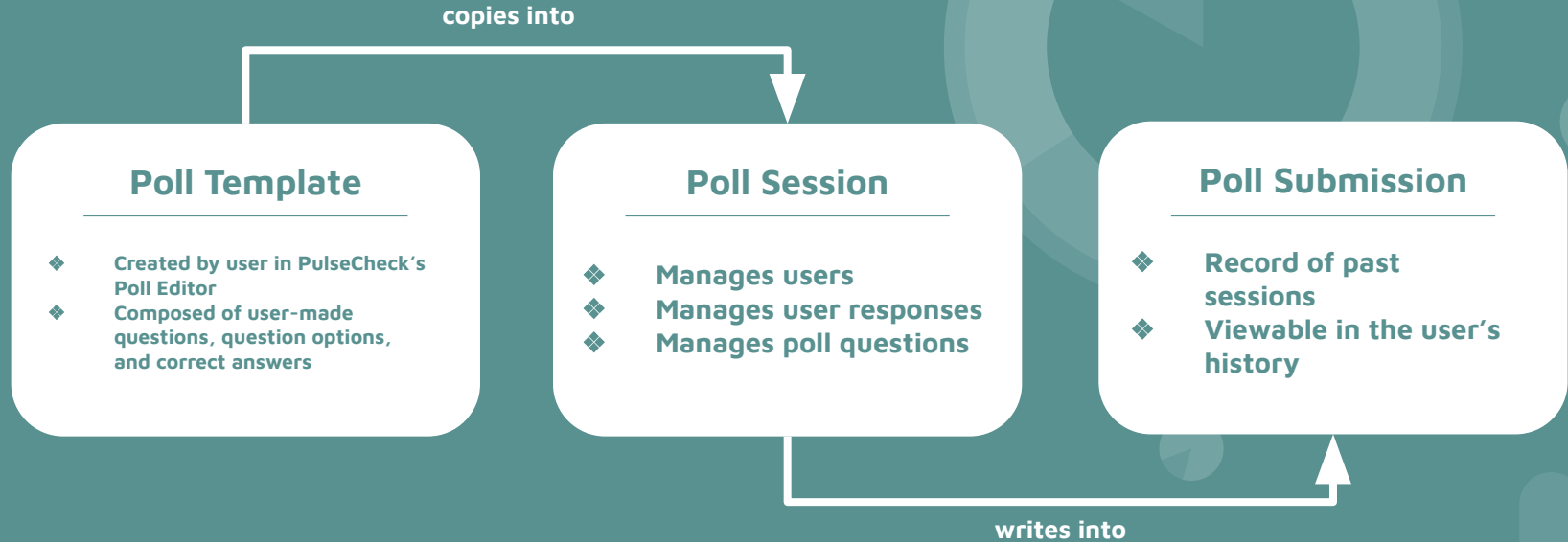- ❖ Changes to the Plan
  - ➤ Removal: Apple Authentication - Lack of CEO funding
  - ➤ Addition: AI Integration - Allow for question generation
- ❖ Changes to the Documentation
  - ➤ Removals and Additional Features Updated in:
    - ■ SRS
    - ■ Project Plan
   - ■ Software Architecture and Design

Live Polling

Poll Template

- Created by user in PulseCheck's Poll Editor
- Composed of user-made questions, question options, and correct answers

copies into

Poll Session

- Manages users
- Manages user responses
- Manages poll questions

Poll Submission

- Record of past sessions
- Viewable in the user's history

writes into

Project Implementation

- Cycle 1: Setup Environment (March 25th)
  - Setup Github Repository
    - Initialize Ruleset
  - Setup Routes to React Components
  - Setup Firebase
    - e.g. Setup Authentication
  - Learn React
- Cycle 2: **CODE** (April 1st)
  - Core Features
    - Authentication (Peter/Trinity/Zaira)
    - Poll Editor (Michael/Brando/Daniel)
    - Poll Sessions (Michael/Brando/Daniel/Trinity)

- Cycle 3: **CODE MORE** (April 17th)
  - Core Features
    - Poll Sessions (Michael/Brando/Trinity)
    - Poll Results (Michael/Daniel/Zaira)
  - Polish System (Everyone)
  - Testing (Daniel/Peter)
  - Additional Features
    - AI Integration (Michael/Zaira)
    - Memory Game (Brando)
    - Miscellaneous (Peter)
- Issues
  - Merge Conflicts
  - Bugs 🐛
  - Time Crunch

Testing Strategies

➢ **Integration Testing**
➢ **Source Code Testing**
➢ **System and Components Testing**
➢ **Use Case Testing**
➢ **Regression Testing**
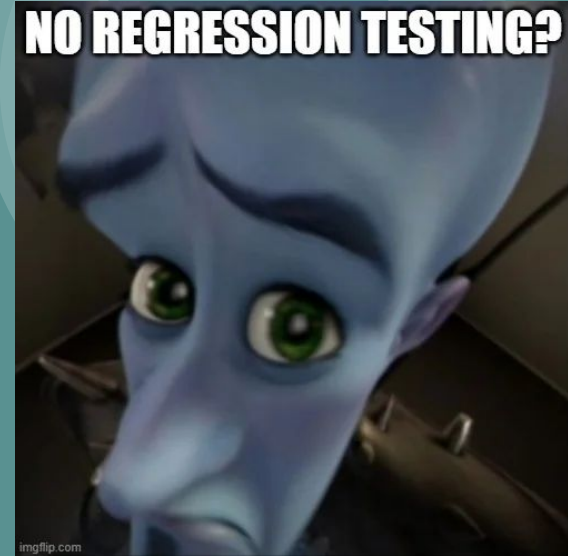
Risk Analysis and Mitigation

❖ **Risk**
- ➢ **Lack of Knowledge of**
  - ■ **JavaScript/TypeScript, React, Firebase, Material UI, Github, and Yarn Knowledge**
- ➢ **Firebase Service Disruption or limitations.**
- ➢ **Communication Risk**
- ➢ **Task Overload**

❖ **Mitigation**
- ➢ **Firebase Service Disruption or limitations.**
  - ■ **Monitor Firebase Status**
- ➢ **Communication Risk**
  - ■ **Use Discord, Google Docs,Github Issues,**
- ➢ **Task Overload**
  - ■ **Prioritize Tasks based on project goals and deadlines**
  - ■ **Drop tasks if necessary**

Team Experience

- What was the teamwork process?
- What did I do?
- What was my role?
- How did we communicate?
- What project management techniques and software did we use?
- What version control methodology did we use?

Conclusions

- What did I learn?