

# README

## Project Overview

Welcome to **TA<sup>2</sup>I**! This README provides a detailed guide for navigating the **TA<sup>2</sup>I** prototype, a tool that personalizes student feedback and helps to organize and reduce the workload of teachers. The document outlines the tools, limitations, and techniques used in crafting the prototype, as well as contextual background on **TA<sup>2</sup>I**.

---

## About TA<sup>2</sup>I

**TA<sup>2</sup>I** is designed to be an intuitive, AI-powered tool that aligns with the real needs of teachers who manage diverse student abilities and high workloads. Through automation, **TA<sup>2</sup>I** significantly lightens the burden of grading, providing timely insights and enabling consistent, fair evaluations. By adapting feedback to each student's individual responses, **TA<sup>2</sup>I** allows for precise, actionable feedback tailored to their learning.

---

## Tools Used

### Primary Tool: Figma

We used Figma for prototyping because of its capabilities in user interface design and interaction modeling. This choice allows us to simulate TAI's user experience while iterating quickly. Below is a summary of Figma's strengths and limitations in this project:

### Pros of Using Figma for TA<sup>2</sup>I's Prototype

- 1. Capturing TA<sup>2</sup>I Functionality**

Figma allows for a detailed representation of TAI's unique features. For example, Figma allows us to simulate **TA<sup>2</sup>I's** automated feedback generation - a key feature of **TA<sup>2</sup>I**. By creating mock feedback flows and interactive screens, Figma helps showcase how TAI tailors feedback to each student's input.
- 2. Streamlined Prototyping and Feedback Collection**

Figma's ease of use makes it possible to build and share prototypes rapidly, crucial for gathering timely feedback on **TA<sup>2</sup>I's** interface and user experience. Users can view and interact with the TAI prototype easily, facilitating feedback loops that are essential for refining **TA<sup>2</sup>I's** uses.

### 3. **User-Friendly Interface for Efficient Collaboration**

Figma's intuitive design and low learning curve empower the **TA<sup>2</sup>I** development team to work efficiently.

## **Cons**

- **Limited Mobile Editing:** Figma does not support editing prototypes on mobile devices, which limits flexibility for making real-time adjustments, especially during user testing sessions with users.
- **Occasional Unintuitive Features:** Certain elements of Figma can feel unintuitive, sometimes hindering the design workflow.
  - **I.e Auto Layout and Constraints:** Setting up auto-layout properties and constraints for responsive design can be challenging, especially when simulating complex UI elements like **TA<sup>2</sup>I's** feedback processes. These panels are intended to expand or adjust dynamically based on user input, but achieving this effect in Figma requires careful manual configuration, which can be time-consuming and may not always reflect the intended flexibility of **TA<sup>2</sup>I**.

---

## **Limitations**

Due to the current capabilities of Figma, certain interactions in **TA<sup>2</sup>I's** prototype are simplified. The prototype cannot fully emulate real-time interactivity, which imposes a few limitations:

- **No Real-Time Interaction Support:** Key interactions like typing and pinch-to-zoom are not supported, limiting the prototype's ability to reflect these features accurately. These limitations mean that key aspects of **TA<sup>2</sup>I** usability—such as a teacher typing comments, reorganizing feedback components -- cannot be demonstrated fully.
  - **Simulated Interactive Elements:** Several interactive components of **TA<sup>2</sup>I**, like adding new classes or adding diagnostic sections, are represented as static, predefined elements in the prototype. These actions are core to **TAI's** adaptive, responsive design, but within the constraints of Figma, they must be simulated as preset actions.
-

# Hard-Coded and Wizard-of-Oz Techniques

Given Figma's limitations, we applied hard-coded solutions and Wizard-of-Oz techniques to demonstrate **TA<sup>2</sup>I** interactions effectively.

## Hard-Coded Elements

- **AI Features:** Functions like generating tailored feedback, diagnostics, and analyzing student work are hard-coded, simulating real AI-driven interactions within the prototype.
- **Integration Features:** Elements such as assignment scraping and student list management are also pre-built, standing in for full backend integration.

## Wizard-of-Oz Techniques

To bring **TA<sup>2</sup>I's** more advanced features to life, Wizard-of-Oz techniques simulate interactions that Figma's capabilities don't support. For example, screens that would typically display real-time data or updates are manually configured in advance, giving users an impression of **TA<sup>2</sup>I's** capabilities.

This approach ensures that the **TA<sup>2</sup>I** prototype is realistic and engaging, facilitating meaningful feedback and helping bring this vision for AI-driven educational support closer to reality.