This is to update you regarding the current status of the app builder product we have been working on.

**Tasks Completed:**

**1. LLM Evaluator**

**Overview:** The LLM Evaluator allows users to assess open-source Language Models (LLMs) across various coding-related domains. Users select an LLM from the list, initiate evaluation, and our server deploys the chosen model. The LLM generates responses for tasks in a predefined dataset, and PS Chat evaluates the generated code against sample code.

**Achievements:**

* Successful integration of LLMs into the evaluation process.
* Accurate and efficient handling of multiple coding domains.
* Seamless collaboration between LLMs and PS Chat for comprehensive evaluation.
* Intuitive visualization of scores through a pie chart.

**2. Figma to Code**

**Overview:** The Figma to Code feature streamlines the process of converting Figma design components into functional React code with Tailwind CSS. Users provide the JSON of a Figma component, and our backend leverages LLM to generate corresponding React code.

**Achievements:**

* Successful integration of Figma designs into React code generation.
* Utilization of LLMs for context-aware code generation.
* User-friendly display and modification of generated code on the frontend.

**AI App Builder**

**Overview:**

The AI App Builder is a feature designed to empower users in the process of application development by leveraging advanced language models (LLMs). The workflow involves selecting a deployment library, choosing an LLM, providing simple and detailed input about the desired product, and receiving application components generated by the LLM.

**Workflow:**

1. **Model Selection:**
   * User chooses between TGI or VLLM for deployment.
   * User selects an LLM from options like Llama2, Falcon, Zephyr, and Mistral.
2. **Query Input:**
   * User submits basic product information.
   * Input is fed into the chosen LLM for initial processing.
3. **Response & Requirement Submission:**

* LLM engages with the user, asking questions to gather detailed application requirements.
  + User provides comprehensive details about the desired product based on LLM queries.

1. **Confirmation or Additional Actions:**
   * LLM generates and builds application components.
   * User receives generated components and confirms the application structure or requests modifications.

**Achievements:**

* **Streamlined Model Selection:** Users can seamlessly choose deployment libraries and LLMs, enhancing flexibility.
* **Interactive Query Input:** The user provides initial information, and LLM-driven questions ensure comprehensive understanding of application requirements.
* **Detailed Requirement Submission:** Users articulate detailed specifications, enabling the LLM to produce more accurate and tailored application components.
* **Dynamic Response:** The LLM dynamically generates and builds application components based on user feedback.