Combining your trained model with the ChatGPT API isn't a direct integration in the sense of merging the two models into one. However, you can architect your system to utilize both models sequentially and in tandem, depending on the specific use case. Here's how you can approach this:

**1. Sequential Use:**

In this scenario, a user's query first goes through one model (either yours or ChatGPT) and then potentially through the second model based on specific conditions or results.

**Example Flow:**

1. User sends a query asking for a recipe recommendation.
2. Your trained model, specialized in recipe recommendations, tries to handle the request.
3. If the model is uncertain or the confidence score of its prediction is below a certain threshold, the query is then passed to ChatGPT for a more general answer or further clarification.

**2. Parallel Use:**

Here, both models process the user's query simultaneously, and the system decides which answer to present based on predefined criteria.

**Example Flow:**

1. User asks for a recipe involving certain ingredients.
2. Both your model and ChatGPT process the request.
3. Your system compares the results, perhaps looking at confidence scores or other metrics, and selects the most appropriate response to present to the user.

**Steps to Combine Models in Your System:**

1. **Integration Points**: Establish endpoints in your backend to interact with both your model and the ChatGPT API.
2. **Query Processing**: Determine how user queries will be processed. If the queries need preprocessing before being fed to either model, set that up.
3. **Decision Logic**: Implement logic to decide which model's output to use. This could be based on confidence scores, the nature of the query, or other factors.
4. **Fallback Mechanism**: If one model fails to provide a satisfactory answer, have mechanisms to revert to the other model.
5. **Feedback Loop**: Implement a way for users to provide feedback on the answers. This feedback can be used to refine the decision-making process over time.
6. **Optimization**: Depending on the volume of queries and cost implications (especially with API calls to ChatGPT), you might want to optimize which model handles which type of queries.

Remember, while combining models can enhance the capabilities of your system, it also introduces additional complexity. You'll need to consider how to handle potential discrepancies in responses, how to merge or choose between them, and how to ensure a seamless user experience.