

INSTALILY.AI

Congratulations, and welcome to your final on-site case study!

Welcome to the final stage of your on-site case study for the **Pool Equipment Chat Agent**.

Your challenge is to develop a **smart chat assistant** that can handle user queries related to **pool equipment, store details, and product information**. The assistant must be able to determine the best way to **retrieve relevant data** by interacting with multiple APIs and structuring responses in a way that's useful to the user.

You'll have access to a [set of documented APIs](#) (you can preview the markdown file in VS Code) that provide product search, pricing, store locations, and more. Additionally, an API key for GPT-4o is provided to you. Before you start coding your solution, you are encouraged to explore the APIs to gain an understanding of what they return and how they can be used. It is also helpful to check out the [website](#) and explore a few product pages to understand how products are identified using their part number, manufacturer ID, manufacturer, etc.

Once the backend is working, it must be **integrated into a messaging platform** of your choice ([Telegram](#), WhatsApp API, Twilio, or another platform with prior approval).

What You'll Need to Do

- Build a **chat-based agent** that can interpret user queries and respond with accurate information.
- Use an **agentic approach** so that the system can decide dynamically which API(s) to interact with.
- Ensure the assistant can **handle multiple types of queries**—some that require direct API lookups and others that need API lookups combined with the reasoning from GPT-4o.

- Design the system in a way that is **scalable** and **adaptable**, allowing for future expansions.
 - Develop a **working front-end integration on a messaging platform** so users can interact with the assistant through a real-world messaging platform.
-

Constraints & Timeline

- This is a **timed challenge** – you have until **End of Day (EOD)** to complete it.
 - It's not about perfection. The focus is on **how you think, design, and build under time constraints**.
-

Evaluation Criteria

You'll be assessed on:

1. **How well the assistant determines which APIs to use** based on user input.
 2. **Accuracy of responses** and the ability to provide useful, structured information.
 3. **Optimised response times** - too many LLM API calls would slow down the chat experience.
 4. **Scalability and maintainability** – how easy it would be to extend the system.
 5. **Effective use of LLMs** to improve responses where API data alone isn't enough and maintain factual accuracy.
 6. **User experience in the chat integration** – responses should be clear, concise, and easy to understand.
 7. **Your approach to problem-solving and reasoning behind key design choices**.
-

Deliverables

1. **A functioning chatbot** that can successfully process queries and fetch relevant data.
 2. **Code repository** with setup instructions and documentation.
 3. **A deployed chat interface** on an approved messaging platform.
 4. **(Optional) A short presentation** outlining your design approach, architecture, and key features.
-

Example Queries Your Assistant Should Handle

The assistant should be able to process and respond to questions such as:

- "Do you have Hayward SuperPumps?"
- "What are the hours of branch XYZ?"
- "What's the price of part LZA406103A?"
- "Can you show me an image of component Z?"
- "What can I use part Y for?"
- "My pool has algae, what's the best algaecide you can recommend?"

Feel free to reach out if you have any questions. We look forward to your agent in action. Good luck!
