



SellScale's Helix: HR Agent

▼ 💡 Pre-Mission Brief

At SellScale, our mission is to grow businesses with Artificial General Intelligence. Currently, dozens of customers are on Selix to grow their sales.

You will be making our fictional HR department, called Helix.

Part of growth means finding great talent - so SellScale has decided to venture into Human Resources via recruiting. This agentic product will help with doing outreach to the best talent.

As a contributor to this project, you will help redefine how recruiters and companies find their best talent. Reshaping the future of work!

Onto the challenge.

🚀 Challenge: Recruiting Outreach

We are building a fictional prototype of the SellScale HR (not sales) Agent called Helix. This app will include a chat-driven interface for collecting user input and a dynamic workspace for displaying and editing a recruiting outreach sequences.

The app should include the following functionality:

1. **Chat Bar (Left Side):** A conversational interface where users can input ideas and preferences about their recruiting outreach. The AI will guide users with prompts and collect information.
2. **Dynamic Workspace (Right Side):** A live document editor where the AI populates a outreach sequence based on user input. Users can directly edit the content, and the AI will respond to edits dynamically.
 - a. **Live Updates:** The AI dynamically updates the recruiting sequence in real-time as the user interacts with the chat or the workspace.

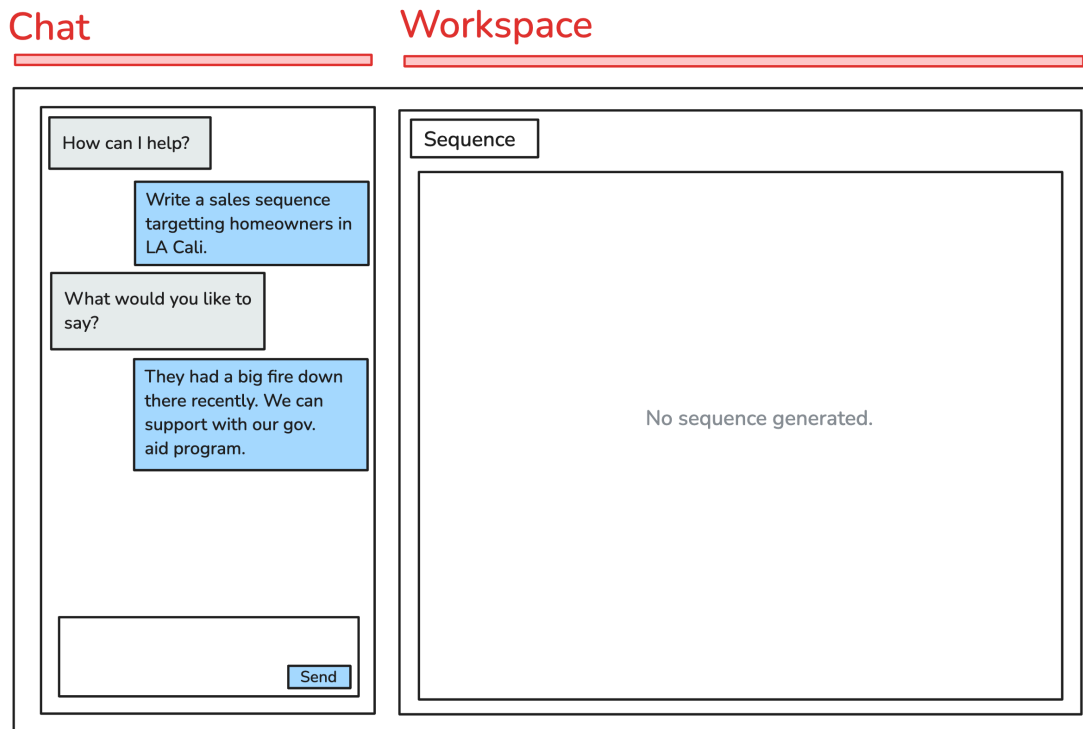
Core Deliverables:

1. A React-based frontend application with a chat bar and workspace layout (uses Typescript)
2. A Flask backend with endpoints for processing chat inputs, generating sequences, and handling live updates.
3. A relational database (eg., PostgreSQL) to store user preferences and generated sequences.

4. Agentic infrastructure - Feel free to use OpenAI Assistants, Langchain, or any agentic infrastructure (with LLM based chat and Function calling)

Example user workflow

1. User chat's with the AI, back and forth, about their campaign idea.



2. After the AI has enough information, it'll generate a simple sequence (dynamically on the right).

Chat

Workspace

Chat	Workspace
<p>How can I help?</p> <p>Write a sales sequence targetting homeowners in LA Cali.</p> <p>What would you like to say?</p> <p>They had a big fire down there recently. We can support with our gov. aid program.</p> <p>Generating sequence...</p> <p><input type="text"/></p> <p>Send</p>	<p>Sequence</p> <p>Step 1: Hi {{first name}} - I've been keeping up with the news in LA. I hope you and your family are safe. Let us know if we can help in any way.</p> <p>Step 2: I work at Trajillo Tech - we release a new government aid program for homeowners affected by the wildfires. Up to \$2mil in aid. Let me know if you'd like to learn more.</p> <p>Step 3: Also .. it's a fully government supported program. No cost or burden to you whatsoever.</p> <p>Let me know!</p>

3. After the sequence is generated, the user can give 'edits' to the AI directly (and/or change the sequence manually on the right) and can work with the AI to make live adjustments.

Chat

Workspace

Chat	Workspace
<p>Write a sales sequence targetting homeowners in LA Cali.</p> <p>What would you like to say?</p> <p>They had a big fire down there recently. We can support with our gov. aid program.</p> <p>Generating sequence...</p> <p>Add a 4th step and let them know I'm here to help. Just reach out.</p> <p>Editing sequence ...</p> <p><input type="text"/></p> <p>Send</p>	<p>Sequence</p> <p>Step 1: Hi {{first name}} - I've been keeping up with the news in LA. I hope you and your family are safe. Let us know if we can help in any way.</p> <p>Step 2: I work at Trajillo Tech - we release a new government aid program for homeowners affected by the wildfires. Up to \$2mil in aid. Let me know if you'd like to learn more.</p> <p>Step 3: Also .. it's a fully government supported program. No cost or burden to you whatsoever.</p> <p>Let me know!</p> <p>Step 4: {{first_name}} - let me know if I can help! I'm here to serve as a resource.</p>

Resources

Use these resources as starting points to complete the challenge. Feel free to explore additional tools and libraries.

- **Frontend:**

- React Documentation: <https://reactjs.org/docs/getting-started.html>
- Create React App: <https://create-react-app.dev/>
- Dribbble for UI Inspiration: <https://dribbble.com>
- Sockets (for dynamic UI updating from backend): <https://socket.io/how-to/use-with-react>

- **Backend:**

- Flask Documentation: <https://flask.palletsprojects.com/en/2.2.x/>

- **Database:**

- Relational
 - PostgreSQL: <https://www.postgresql.org/docs/>
 - SQLite: <https://sqlitebrowser.org/>
- Vector
 - Pinecone: https://www.pinecone.io/?utm_term=pinecone%20database&utm_campaign=brand-us-e&utm_source=adwords&utm_medium=ppc&hsa_acc=3111363649&hsa_cam=21023369441&hsa_grp=1674706671627713670725&hsa_kw=pinecone%20database&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid
 - Chroma: <https://www.trychroma.com/>

- **Agentic Infrastructure**

- Langchain: <https://www.langchain.com/agents>
- OpenAI Assistants: <https://platform.openai.com/docs/assistants/overview>
- Crew AI: <https://www.crewai.com/>

- **Sample Agentic Applications**

- SellScale AI: <https://www.sellscale.com/>
- Devin AI: <https://www.youtube.com/watch?v=fjHtjT7GO1c>

Checklist

User Checklist

- ☐ User should be able to chat 'back and forth' with the AI
- ☐ The AI should ask follow up questions as needed to identify sequence content, angle, # steps, etc
- ☐ After enough data is collected, the AI should 'create a sequence' (live) on the right workspace
- ☐ The user should be able to edit the sequence in the workspace on the right
- ☐ The user should be able to also request edits 'directly to the AI'

Technical Checklist

- ☐ Leverage a relational database for data store
- ☐ Use Flask for the backend and ensure backend is modular
- ☐ Use an agentic framework like Langchain, OpenAI Assistants, or Crew AI to build out the LLM portion.
- ☐ Ensure the frontend is modular and built in React + Typescript



IMPORTANT: Ensure that the core functionality is solid based on the golden flow highlighted in the section titled **Example user workflow**

Bonus

- ☐ (full stack) Have an ability to have multiple 'sessions' per user
- ☐ (full stack + agentic architecture) Have a way to store 'context' about the user + their company during some sort of sign up flow so the AI doesn't ask silly questions (ex. "Where do you work?")
- ☐ (full stack) Add more 'tabs' and functionality in the workspace
- ☐ (full stack) Add 'streaming' so the chat experience feels more ChatGPT-esque
- ☐ (agentic) Create a RAG memory so the agent remembers historical requests for future iterations



Deliverables

Submit the following deliverables as part of this challenge. Ensure all components are functional and tested.

Technical Deliverable

- Submit a zip file containing:
 - The full codebase.
 - A **README** with clear instructions to set up the application locally (test the setup instructions yourself!).

Video Deliverables



Please submit this as a **Loom** link or similar in the email body for fast viewing.

1. Product Demo (2-5 mins):

- Show the full functionality of your application.
- Demonstrate how the user can interact with the chat, see the AI-generated sequence, and edit it dynamically.

2. Technical Walkthrough (2-5 mins):

- Explain how the app is built.
- Discuss backend and frontend architecture.
- Highlight key decisions and improvements you'd make in the future.



How Long Should I Spend?

Time Window: Roughly 3 days from when assigned, but please refer to your email for specifics.

You can spend as long or as little as you'd like, but submissions must meet the deadline. Prioritize functionality and clarity. If we had to choose a benchmark, we can put 3-5 hours.



Assessment Criteria

We will evaluate submissions based on the following:

- ☐ Functionality - Does this hit the three requirements prescribed in the spec?

- ☐ Technical Requirements — Is there a backend, frontend, and database?
 - ☐ Solution Elegance — Did you use APIs, libraries, SDKs, etc to move fast?
 - ☐ Code Cleanliness - Can I read the code and easily understand / navigate the codebase?
 - ☐ Quality - Does the app look sick?
 - ☐ Communication — SellScale has a huge Shipyard culture! We love to see clear video demos.
 - ☐ Agentic Architecture - we'll be reviewing your LLM prompts and agentic architecture as well
-

Good Luck! 🍀 We're excited to see your deliverable. Remember to be creative and have fun building!