# **Helix: The Agentic Recruiter**

# **💡 Pre-Mission Brief**

The HR department noticed sales using Selix, and they got jealous!

After noticing this, you take it upon yourself to create a new product line… in **3** days 🤯.

The purpose of the agent is to help do outreach to find the best talent.

As an [**AI Engineer**](https://www.sellscale.com/blog-posts/the-rise-of-the-ai-engineer), you will be building this agent to help redefine the future of work, for yet another corner of the office!

Onto the challenge.

# **🧘 Purpose**

The purpose of this section tests a few things:

* Where your quality bar is for shipping product
* How well you can ship on an open-ended problem space
* How well you empathize with user (AKA the recruiter. There are many items not covered in this assignment you may want to think about)
* Familiarity with agentic product design, or ability to learn it (this is an Agentic-first AI product, not SaaS 1.0. There will be different design choices - eg chat based interface)
* Backend and agentic architecture

# **🚀 Challenge: Recruiting Outreach**

**We are building a fictional prototype of the SellScale HR** (not sales) **Agent called Helix.** Essentially think you are building this for a recruiter of a fast growing company. They don’t have much time to write out commas and sentences for outreach, but they have a great idea of how they’re recruiting for. You want the agent to do the work for them - collect input where useful, but have the agent do the rest - use the human for review.

This app will include a chat-driven interface for collecting user input and a dynamic workspace for displaying and editing a recruiting outreach sequences.

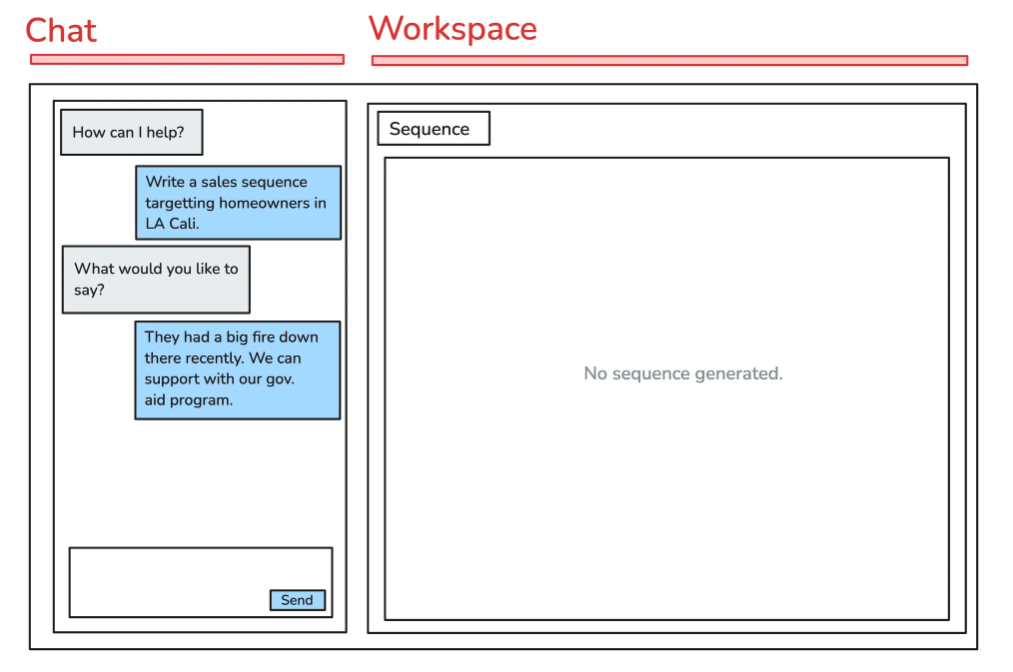
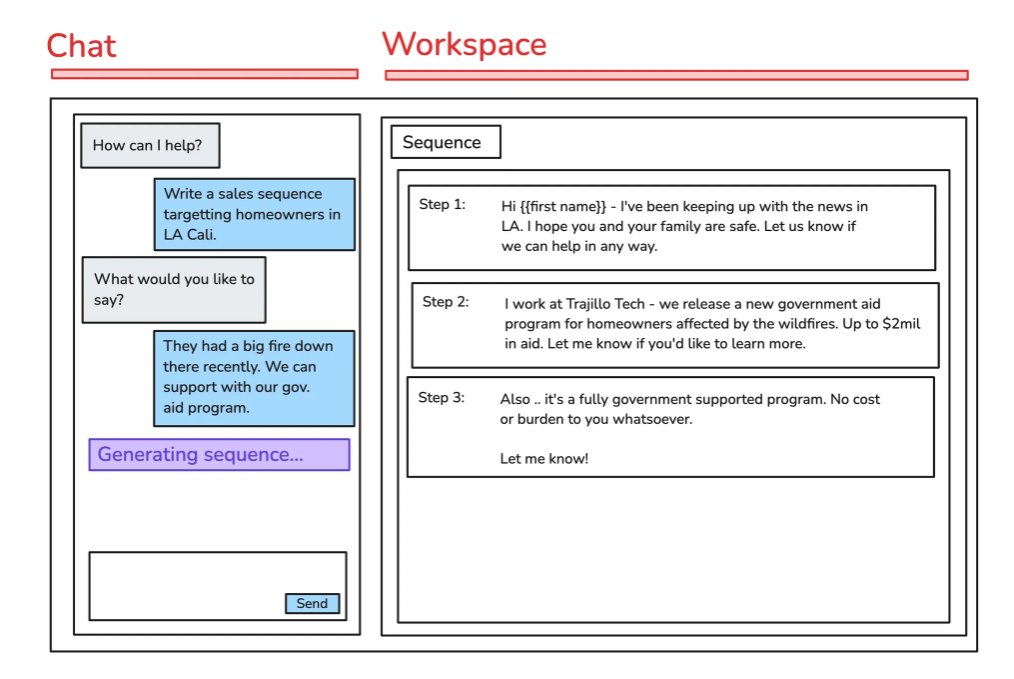
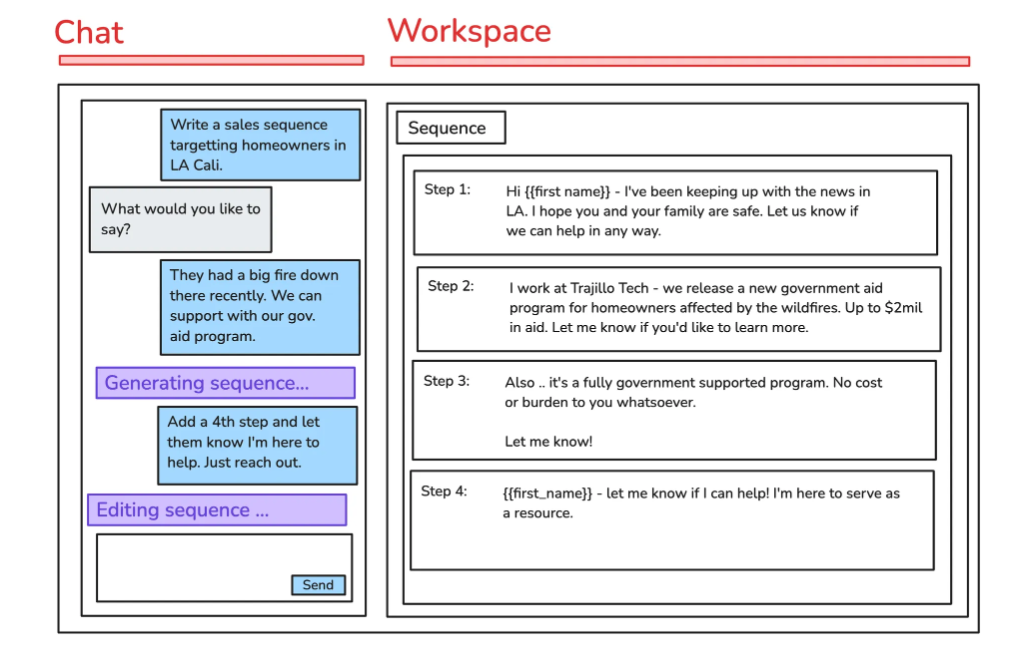
**The app should include the following functionality:**

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 workspace with individual artifacts for each sequence step. Users can directly edit the content, and the AI will respond to edits dynamically.
   1. **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).  
     
    
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.  
     
    

# **🥽 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 database&utm\_campaign=brand-us-e&utm\_source=adwords&utm\_medium=ppc&hsa\_acc=3111363649&hsa\_cam=21023369441&hsa\_grp=167470667468&hsa\_ad=690982708943&hsa\_src=g&hsa\_tgt=kwd-1627713670725&hsa\_kw=pinecone database&hsa\_mt=e&hsa\_net=adwords&hsa\_ver=3&gad\_source=1&gclid=Cj0KCQiAkJO8BhCGARIsAMkswyh3xTPWddfipP\_I57j6CHFBWRqvWZj1vOoL2DdEW5becfLAhxNQcg8aAqhxEALw\_wcB](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=167470667468&hsa_ad=690982708943&hsa_src=g&hsa_tgt=kwd-1627713670725&hsa_kw=pinecone%20database&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=Cj0KCQiAkJO8BhCGARIsAMkswyh3xTPWddfipP_I57j6CHFBWRqvWZj1vOoL2DdEW5becfLAhxNQcg8aAqhxEALw_wcB)
    - Chroma:<https://www.trychroma.com/>
* **Agentic Infrastructure**
  + SellScale:<https://www.sellscale.com/blog-posts/the-magic-of-agentic>
  + Langchain:<https://www.langchain.com/agents>
  + Crew AI:<https://www.crewai.com/>
* **Useful live examples**
  + Selix:<https://www.sellscale.com/>
  + Replit:<https://www.youtube.com/watch?v=IYiVPrxY8-Y&ab_channel=Replit>
  + 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  
    
   <aside> 💡  
    
   Note - a “Brain” to not ask redundant / silly questions (eg company name) would be smart here.  
    
   </aside>
* [ ] 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
* [ ] (preferred) Build your own agentic framework to store messages, call tools, and more
  + Or, use an agentic framework like Langchain, OpenAI Assistants, or Crew AI to build out the LLM portion (less preferred but still acceptable)
* [ ] Ensure the frontend is modular and built in React + Typescript

<aside> 💡

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

</aside>

### **Bonus 🎉**

* [ ] (agentic) Additional tools - e.g., web browsing, unique architecture, actually sending the messages - anything to help a recruiter do their job even quicker!
* [ ] (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) Have an ability to have multiple ‘sessions’ per user
* [ ] (full stack) Add ‘streaming’ so the chat experience feels more ChatGPT-esque
* [ ] (agentic) Display calling\_tool notifications in-chat to help the user understand what’s happening

# **⏰ 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. We believe a strong engineer should be able to go above and beyond in 4-8 hours.

# **🌊 Assessment Criteria**

We will evaluate submissions based on the following:

* [ ] Functionality - Does this hit the 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 production-grade, ready to publish for the masses?
* [ ] 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

# **✉️ Final submission**

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

* [ ] **1. Zip file** - **containing…**
  + The full codebase.
  + A **README** with clear instructions to set up the application locally (test the setup instructions yourself!).

<aside> ⭐

Two videos below - please submit this as a **Loom** or similar in the email body for fast viewing.

</aside>

* [ ] **2. Product Demo video (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.
* [ ] **3. Technical Walkthrough video (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.

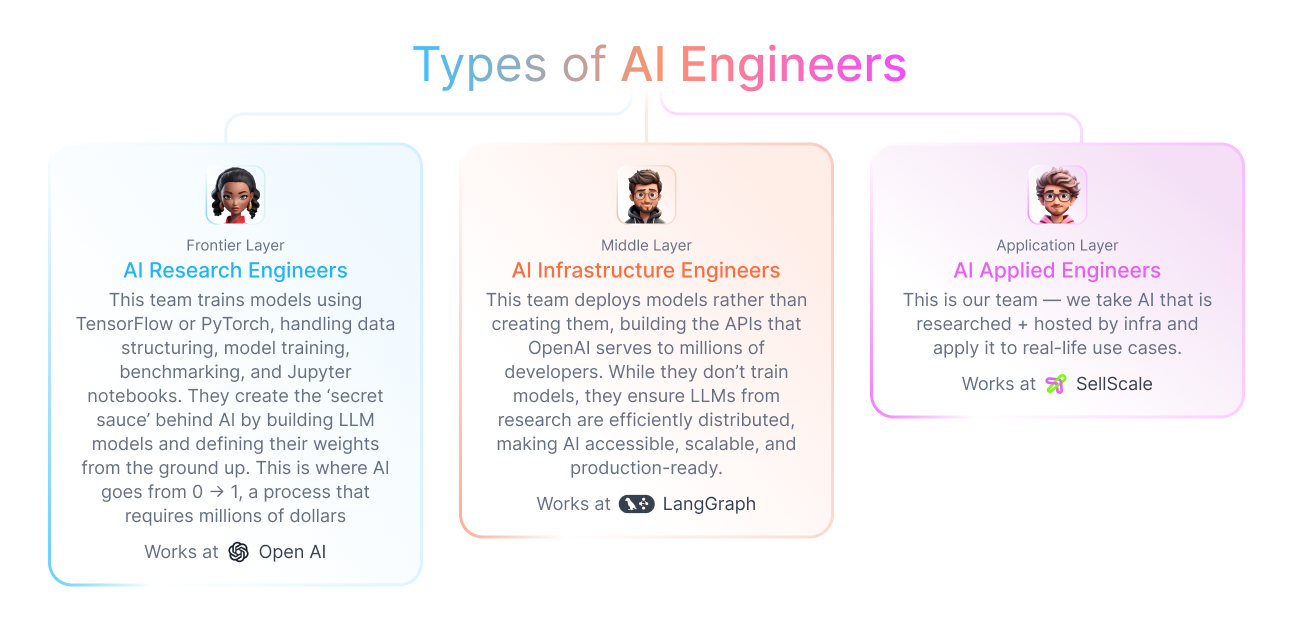
—

### **Good Luck! 🤞We’re excited to see your deliverable. Remember to be creative and have fun building!**

Reference:  
  
As AI proliferates, a new breed of engineers is emerging: AI Engineers. Unlike traditional software engineers, AI Engineers specialize in integrating, deploying, and optimizing AI models. They navigate complex ecosystems involving LLMs, APIs, and traditional software to develop apps.

### **Types of AI Engineers**

AI Engineers can generally be categorized into three main types based on their focus areas and contributions:



### **1. Frontier AI Engineers**

*Example: Member of Technical Staff at OpenAI*

These engineers work at the cutting edge of AI research, typically employed by organizations like OpenAI or Anthropic. They often hold titles like "Member of Technical Staff" and are responsible for designing and training large-scale models. Their work involves:

* Developing foundational AI models (e.g., GPT, Claude, Gemini)
* Training deep learning networks using TensorFlow and PyTorch
* Benchmarking and optimizing model architectures
* Managing vast datasets and computing infrastructure

Frontier AI Engineers operate at the 0 → 1 stage of AI development, requiring extensive resources, research capabilities, and financial investment—often in the millions of dollars—to build net-new AI models.

### **2. Infrastructure AI Engineers**

*Example: SWE at LangChain*

These engineers specialize in deploying and scaling AI models, ensuring that foundational AI research is accessible via APIs. Companies like [Anon.com](http://anon.com/), LangChain, and BaseTen fall into this category. Their primary responsibilities include:

* API development and model hosting
* Optimizing inference for speed and cost-effectiveness
* Ensuring distributed and scalable AI deployments
* Creating SDKs and tools for developers

While they don’t create new AI models from scratch, Infrastructure AI Engineers ensure that cutting-edge AI research reaches millions of developers worldwide in an efficient and usable manner.

### **3. Applied AI Engineers**

*Example: Founding Engineer at SellScale*

Applied AI Engineers focus on integrating AI models into real-world applications. They leverage APIs and pre-trained models to build functional products, such as:

* AI-powered sales outreach tools (e.g., SellScale)
* AI-driven software agents (e.g., Devin, Harvey)
* Custom AI applications for various industries

These engineers work at the application layer, optimizing AI workflows, refining prompts, and ensuring that AI interacts effectively with users. Their primary focus is on making AI useful and impactful in practical scenarios.

### **The Core Skills of an AI Engineers in the Application Layer**

#### **Previous Key Skills:**

* Logic, systems, and scalability
* Basic UI/UX
* CS fundamentals

#### **New Key Skills:**

* Liberal arts
* Prompting techniques
* Self-evaluation

### **What skills are new?**

*While the field of AI Engineering is evolving rapidly, several foundational skills are essential for success—especially in the application layer.*

1. **Liberal arts**

* AI is fundamentally about processing and understanding language.
* The ability to craft effective prompts and interpret AI outputs is a core skill.
* Engineers with strong linguistic intuition will have an advantage in fine-tuning AI behavior.

1. **APIs & JSON Proficiency**

* APIs are the backbone of AI applications.
* Engineers need to understand how to integrate AI via API calls, interpret documentation, and structure JSON responses.
* The ability to modify and manipulate JSON payloads creatively is a critical skill.

1. **Self-Critique**

* Unlike traditional engineering, where user feedback drives iteration, AI Engineers must assess and refine model outputs independently.
* AI degradation is a major concern—feeding AI-generated content back into AI models can lead to quality loss over time.
* Vigilance in evaluating and optimizing LLM outputs is essential.

### **What skills are still important?**

**Traditional CS Fundamentals**

* A strong computer science background remains a prerequisite.
* Over-reliance on tools like Cursor for autocomplete will limit deep technical understanding.
* Core skills in data structures, algorithms, and software engineering are still critical.

### **Conclusion**

Most engineers are SaaS or application based today.

The AI Engineer will continue to proliferate. You can use these skills to grow into it.