Technical Requirements for Amolino ReactJS frontend

Key Idea

We are building an AI-based copilot that takes information from slack and email, summarizes the information and tells the user how to react to requests and updates them. This is a web based application that will work well on desktops, mobiles and tablets.

I will provide complete user design and if you need to talk to the UX designer that can also be arranged.

This project is also has a lot of details and requirements on the tech-stack, naming, components etc. This is a fully featured front-end (not backend) but the project will make real API calls to the backend server that will be mocked using Mock Service Worker (MSW).

While this Upwork job is structured as a one-time job, I am looking for a ReactJS engineer who I can work with on an ongoing basis. This job has a strict timeline of delivery of 5 days. This job will require working through the weekend so if you are not open to it, please don't apply.

Stack

The stack is not negotiable. If you have a strong reason to suggest an alternative for any of these please discuss before starting the project.

- Tanstack for async statement management
- Mantine for UI component
- Mock Service Worker
- Axios for network requests

All code has to be TypeScript5 only (no file should be JS or JSX, everything needs to be TSX or TS)

Please install ESLint and Prettier and make sure there are not ESLint or Prettier errors. Also, don't use ":any" anywhere. Use proper types.

The main part of the application consists of a lot of cards that are called **ActionPods**. Each of the **ActionPods** has different types of information. See the mockup.

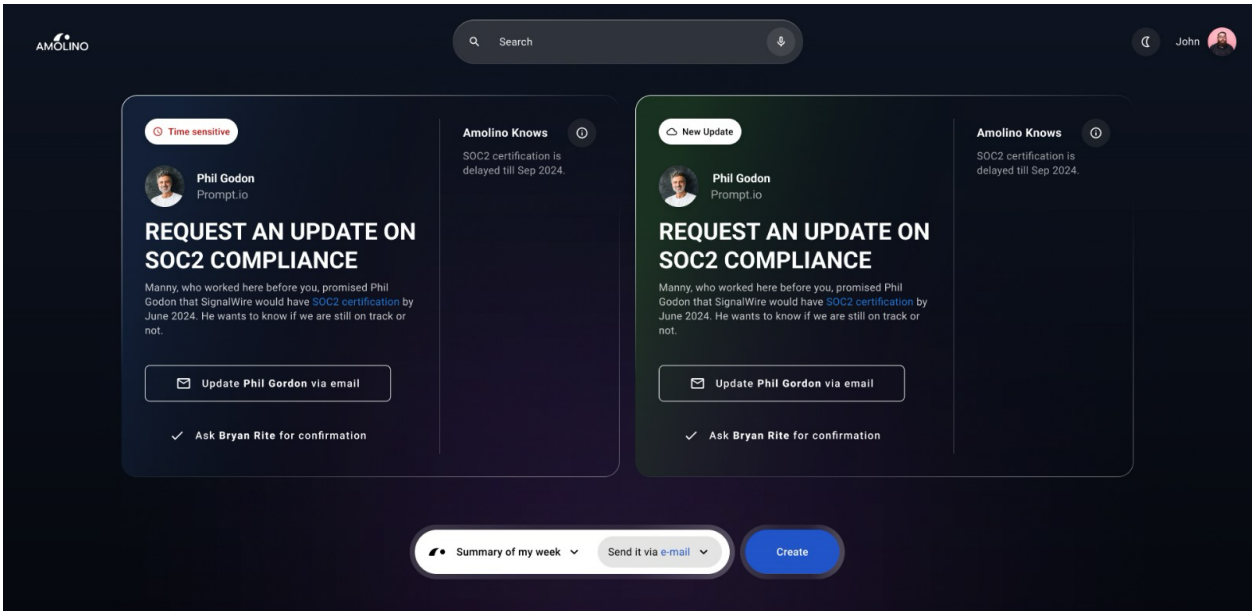
See the attachment for more specific

There are two different types of cards (only one kind is shown in the mockup).
- Request cards (shown in the mockup
- Update cards

Request cards are generally time sensitive and synchronous. They are based on a request from someone.

Update cards are based on information that the AI agent has learned from someone. They are asynchronous. Update cards have similar information as Request cards but need to look visually different

The bottom has a fixed section of things that the AI agent can help with. What is shown in the mockup is this "Create a summary of my weekend" and "send it via slack" and "send it via email". Each of these should be designed as small card that can be selected or unselected. In other words it should be possible to select the following - create a summary of my weekend and send it via slack and leave "send it via email" unselected". There needs to be a "do" or execute button for this line.



Server Interaction

The server will expose the following endpoints on localhost:8000 (make sure this is defined in a constant and not used all over the place)

Get list of actionpods

GET http://localhost:8000/actionpods
This will return an array of actionpods. Each action pod is defined in the following JSON file

```
{
        Id: <uuid>
        ActionPodType: enum { “request”, “update”, “summary”, “insight”}
        ActionPodLabel: array of enums of “time-sensitive”, “from-customer”, “from-coworker”
        ActionPodPerson: {
                Name: string
                Email: email
                Company: string
                Link: URL
        }
        ActionPodSubject: string
        ActionPodTopic: string
        ActionPodKnowledge: array of {
                infoItem: string
                infoSource: string
                }
        ActionPodActions: array of {
                actionName: string
                actionID: uuid
        }
}
```

Code Walkthrough

Before final payment is done and code is handed over, we’ll do a code-walk through for at least one hour or more if needed.
Code has to be handed over as a private github repository which we’ll create and add you as a collaborator

Components

Each visual element in the UI that is attached has to be a separate component (that needs to read from TanStack and not query the server directly)

ActionPod
ActionPodType
ActionPodLabel
ActionPodSubject
ActionPodTopic
ActionPodPerson
ActionPodKnowledge
ActionPodActions

Milestones

Milestone 1

code setup, files and directories are created, ESLint and Prettier installed, components are created, code is compiling not all component are empty (skeletons)
1 day

Milestone 2
Tanstack queries and mocks done
1 day

Milestone 3
React components fleshed out and working. Test with at least 4 ActionPods. Code handover.
2 days