evabot

San Francisco, USA-based company provides a cloud-based AI gifting chatbot platform. It collects and analyses customers' information using a conversational interface, which they then use to ship a memorable physical gift.

Assisting in product development and growth to build "fully automated" gifting experiences

Engagement Overview

Crownstack's partnership with EvaBot. The scope included the platforms' Admin/web & CRM. They have also made seamless connections with their pre-existing CRM tools. The team works with EvaBot to achieve their vision to scale Evabot's operations, product development and growth with AI to build "fully automated" gifting experiences.

About the Customer

EvaBot AI is the sales Intelligence and engagement layer across your sales cycle that enables reps to automate the research, personalisation and engagement. Sales teams use Eva to multithread and build more champions faster, driving pipeline velocity.

Industry	Software Development

Business Challenges

- Evabot has a pre-existing Chat Flow where they share the chat links with the user, and based on the response and ML analysis, they customise the gifts. This complete flow needed end-to-end refactoring for better performance.
- Evabot has many umbrellas; they planned to have their customised theme library similar to Material UI.
- Evabot wanted to develop a new system based on the accessibility of salesforce data during the signup process for the new flow named Rapport Dashboard.
    - With the rapport dashboard system, they wanted their customised Chrome extension that helps generate emails and LinkedIn messages based on your selections and with Chat GPT integration.

Business Solution

Our Engineering team worked with the EvaBot team to conceptualise the solution and implement various integrations with their existing systems.

- The complete project has been refectory for the Evabot Chat UI Flow. Best coding practices and architecture were followed. Reusable components, redux state management, and the latest hooks, like useCallback, useMemo, etc., were integrated, and a new API structure was added. Multiple levels of testing were done at each level of refactoring.
- The Evobot Core Library was built from scratch with styled-components, styled systems, material-ui, formik, yup validations, typescript, and react-intl, and the team followed atomic structure. Different base components were developed based on the theming, and a library was published via the npm package.
- A new umbrella named Rapport Dashboard was developed with the help of core-library components. Initially, it had two roles: the Admin and the User. Both flows had multiple-step signup systems that had salesforce data integrated. Later, they had just the user login and forgot password flows, followed by the dashboard, profile, nudges, stakeholders, analytics, activities, and tags sections integrated.
- The team integrated the Chrome extension initially built using vanilla JS, but later, it planned to move forward with React. It had many enhancements and explorations over the time of development.

Technology Stack

Frontend	ReactJs, HTML/CSS
Workflow	Slack, Github, Figma, JIRA

Business Outcome

- With the refactoring of Evabot Chat UI Flow, the code became more stable, optimised, cleaner and approachable for developers as we followed atomic structure architecture. The platform performance increased due to more efficient code and less API calls. In the end, the overall user experience has been improved.
- Core Library is quite helpful for Evabot structure as they use these components in their new umbrellas like rapport dashboard and chrome extension. These components are flexible and reusable.
- Rapport Dashboard and Chrome Extension are their ongoing umbrellas. With time, these are shaped quite well and stable, making it easier for the Evabot product and sales teams to approach multiple clients via them.

Further Links

[Learn More About Crownstack's Offerings]()
[Learn More about EvaBot]()