Full Stack Developer Interview Coding Task

Careology's Mission

Our mission is to 'transform traditional cancer care by building patient-first technology that is connected, intuitive and easy for everyone to use'.

With our groundbreaking Digital Cancer Care app for patients and Careology Professional, our innovative Remote Patient Monitoring platform, healthcare professionals can access real-time (and historic) patient information that will enable them to identify potential complications during treatment and facilitate early interventions. This will transform the patient experience, unlock a new standard of digital cancer care and has the potential to save lives.

Our Technology

We are a SaaS platform leveraging the AWS platform and an ecosystem of external SaaS components when needed. Architecturally we strive to be "serverless first", where practical, and recognise and use containerization where necessary.

A high level view of our tech stack is:

- AWS tools and resources inc: AppSync, DynamoDB, CodeBuild/CodePipeline, IAM, Lambda, S3, SNS, SQS, CloudWatch, Cloudformation, RDS, Cognito, API Gateway, S3+CloudFront hosting
- CI/CD with Bitbucket/CodeBuild/CodePipeline/Cloudformation
- Front-end: ReactJS with styled-components
- Mobile: ReactJS with CapacitorJS
- Languages: Javascript.
- Ops: CloudWatch / Data Dog

Coding Task Background

We want you to **shine** and bring **yourself** into this task. We've opted for a simple to-do app and will want to see how you bootstrap this project, how you **build** the **front end UI** and also how the **backend** code looks like.

Coding Task Requirement

Please build a simple to-do app using **ReactJS** and **NodeJS.**
Tip: no need to build everything from scratch, AWS Amplify is your way in, and remember google is your friend (but we do expect you to understand the code you submit and answer questions about it).

The app will need to have the following:

1. Login / Register screen(s)
2. Ability to logout
3. Ability to add task
4. Ability to edit task
5. Ability to delete task
6. Ability to mark/check task as done
7. Ability to attach a file to a task.
8. If a file was attached to a task be able to hover over it and see metadata for it (name, size, type, last modified date and time)

9. If a new task contains a city name in it then use an external API to show the weather for this city on the task card using weather.com API. Swagger Link. API Key - 5e5763c7175b4db792853554242107
10. Be sure that tasks are protected from viewing and editing by another user

- Todo app backend

We use AWS technologies in our tech stack. In this task you will need to work with this AWS technologies to create your backend.

1. AWS AppSync
2. AWS Cognito
3. AWS Amplify
4. AWS DynamoDB
5. AWS S3
6. GraphQL

- Todo app frontend

We use **ReactJS** and **Javascript** for our frontend. We typically use **styled-components** library for our UI. To cogent to our **graphQL** backend API we also use the **Apollo client**.
Figma is used for designs and prototyping. Please implement this Figma design for the UI of the to-do app

- Bonus Requirements (optional)

- Using AWS technologies, make your app available **online** (send us the link to it).
- Implement **dark** and **light** themes.
- Add snackbars.
- Implement drag n drop.
- Create hamburger menu option to sign out
- Pop up for share menu

- Task Notes

- The app should render the to-do list of the logged in user.
- Our apps are written in **javascript**, make sure your task aligns with it
- When developing your ui, please consider the use of the **styled-components** library.
- Please make sure your design is **responsive**.
- Please take into consideration concepts like component **hierarchy, modularity, reusability**.
- Error Handling - Implement robust error handling to manage any potential issues.

Deliverable

Please send a link to a git repository with a **readme file** on how to run the app (don't forget to send us the .env file separately).

There will be a demo of your app and thought process to a small group from the R&D department. Please be prepared to talk through your reasonings and answer technical questions.