Question 1: Can you explain the process of building data and logic-driven web-based workflow systems using bubble.io?

Building data and logic-driven web-based workflow systems using bubble.io involves creating user interfaces using bubble's visual editor, setting up
database structures and workflows, and defining the logic behind the workflows using bubble's built-in tools. Custom coding might be required if the built-in tools are not enough to meet the requirements.

Question 2: What projects have you worked on in the past with custom coding?

In the CV, the candidate has mentioned working on the DISTORT, MATCHER, and ALL_CRYPTO projects with custom coding. The DISTORT project is a Discord import ReactDOM from "react-dom/client";

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Layout from "./pages/Layout";
import Home from "./pages/Home";
import Blogs from "./pages/Blogs";
import Contact from "./pages/Contact";
import NoPage from "./pages/NoPage";
export default function App() {
 return (
  <BrowserRouter>
   <Routes>
    <Route path="/" element={<Layout />}>
     <Route index element={<Home />} />
     <Route path="blogs" element={<Blogs />} />
     <Route path="contact" element={<Contact />} />
     <Route path="*" element={<NoPage />} />
    </Route>
   </Routes>
  </BrowserRouter>
 );
}
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
```

Question 3: How do you typically go about debugging and testing your code?

Debugging and testing code involve using various developer tools, such as React Developer Tools for frontend debugging, and Google Chrome Developer Tools for checking API calls and error handling. Backend debugging might involve using a tool like Postman to test API endpoints. Unit testing frameworks like Jest or Mocha can also be used to test individual functions or components.

Question 4: What is your approach to documenting a project?

Documenting a project starts with writing clear and concise code comments that explain the functionality of the code segments. In addition, creating
a comprehensive readme file that details the project's requirements, technologies, installation process, and usage guidelines is crucial. Also, using tools like JSDoc or ApiDoc for generating API documentation, and documenting any custom configurations or settings unique to the project, is essential.

Question 5: Can you list the different data sources, SDKs, and APIs you have experience working with?

In the CV, the candidate has mentioned working with Firebase, Axios, and Heroku for backend-frontend connections and environment-variable protection. They have also likely worked with other data sources, SDKs, and APIs in their past projects, but specific examples are not provided.

Question 6: How do you demonstrate your strong analytical skills in your work?

Strong analytical skills can be demonstrated by efficiently analyzing project requirements, breaking complex problems down into manageable tasks, and using algorithms and data structures to optimize code performance. Additionally, these skills can be showcased by continuously seeking to improve the project deliverables while maintaining high code quality and adhering to deadlines.

Question 7: What is your experience with HTML/CSS and frontend frameworks like React?

The candidate's CV mentions experience in developing web applications using HTML, CSS, and JavaScript, along with frontend frameworks like

ReactJs. Moreover, they have used React components, and state management libraries such as Redux, and additional libraries like Tailwind CSS.

Question 8: What is Axios and how is it used in a React project?

Axios is a popular JavaScript library for making HTTP requests from the client to the server. In a React project, Axios is used to fetch data from APIs and send data back to the server. It provides a more flexible, easy-to-use interface than the built-in JavaScript fetch method and handles XMLHttpRequests, making it compatible with older browsers.

Question 9: How do you implement authentication using Google Auth in a project?

To implement authentication using Google Auth, you can use the Google APIs Client Library or a popular library like Firebase Authentication. You'll need to register your project on the Google API Console, obtain OAuth 2.0 client credentials, and include the appropriate SDKs or libraries in your project. Once set up, users can sign in with their Google accounts, granting access to their profile information and enabling you to integrate authentication seamlessly in your application.

Question 10: What are the main differences between SQL and NoSQL databases?

SQL databases are relational and use structured query language (SQL) for managing and querying data. They operate on predefined schema, have strict

relationships between tables, and usually follow the ACID properties. NoSQL databases, on the other hand, are non-relational and can store unstructured and semi-structured data. They don't follow a fixed schema and offer more flexibility in data storage and retrieval. Examples of NoSQL databases include MongoDB, Couchbase, and Cassandra.

Question 11: What is the role of React Router in a React application?

React Router is a powerful routing library used to handle client-side navigation between different views or components in a React application. It allows developers to define routes and associate them with specific components, controlling which components are rendered based on the current route in the application. This provides a dynamic, SPA (Single Page Application) experience for users.

Question 12: Explain the concept of a virtual DOM in React.

The virtual DOM is a lightweight, in-memory representation of the actual DOM used by React to optimize component rendering. When a component's state changes, React generates a new virtual DOM and performs a diffing algorithm to determine the minimal set of changes required to update the actual DOM. This process is called reconciliation, and it makes React applications render updates more efficiently and performantly.