

Bling - Frontend Code Challenge

Make a simple single-page web application for Broccoli & Co.

Broccoli & Co., an upcoming online service company, would like to let people "Request an invitation" on their website.

The functionality

- Create a simple yet clean homepage for them that allow users to enter their name and email to receive email invitations.

Visual Requirements

- The UI should occupy the full height of the screen.
- Shows a fixed header that is always on top of the window and a footer that is always on the bottom of the window (assuming a reasonable window height).
- The page content is sandwiched in the middle, containing just a heading, a small piece of text and a button to request an invite.
- A rough mockup of the basic layout is attached. While preserving this layout on desktop, you may style it however you wish, with or without images.
- The solution must be mobile friendly (users won't need to pinch and zoom on their mobile devices).

UI Behaviour / Validation

- When the Request Invite button is clicked, a popup shows containing the **Full name**, **Email** and **Confirm Email** input fields.
- The user needs to fill in all three fields to request an invite. **Full name** needs to be at least 3 characters long, **Email** needs to be in a valid email format and Confirm Email needs to match **Email**. (You may display the validation errors inline)
- If the user clicks **Send** and one or more fields do not validate properly, the app should not contact the backend but instead provide appropriate feedback to the user (use your judgement on what this UX should be).
- If the user clicks **Send** and all fields validate properly, the app should send the request to the backend server (see specs below) and inform the user that the request is being sent.
- If the server returns 200 OK, it should switch to another popup, indicating that everything went through OK. This popup can be dismissed and will simply close - revealing the homepage again.
- The server may return a 400 Bad Request, in which case the app should simply display the error message from the server.
- The Send button can be clicked again to re-attempt the submission.

Tooling

- The solution should be implemented in React.
- The solution should use a modern build tool, e.g. create-react-app, Webpack, NextJs. The solution should use a modern styling approach.
- The solution should run on most current browsers (e.g. flexbox okay).

Backend API

- The API endpoint is available on <https://us-central1-blinkapp-684c1.cloudfunctions.net/fakeAuth>
- The request is in the form of a JSON payload { "name": "XXX", "email": "XXX" }
- The request must use the POST method.
- Email "usedemail@blinq.app" is hardcoded in the backend to trigger a specific error that needs to be handled by the front-end app.

Submission

- Please submit your code via an online Git repository or simply zip it to us, including a README.md for building, running and testing the app.
- Write some tests to make sure the app behaves as expected.
- Do not develop your own backend API, use the one specified above.

Appendix - Mockup

