

Tech test: B I N G O



“a game in which players mark off numbers on cards as the numbers are drawn randomly by a caller, the winner being the first person to mark off all their numbers.”

Overview

The local community hall has commissioned you to create a “BINGO” web application. It will be played on small devices while numbers are called out by their presenter (not a function of this application).

We have broken this test up into 2 parts: (1) Frontend and (2) Backend.

1. **The Frontend part** will test your existing knowledge of development using React + ChakraUI and your ability to pick up new concepts.
2. **The Backend part** is an extension of the Frontend part designed to test your knowledge of Backend technologies including Laravel + PHP or Node.js (this will be stipulated in our communication with you as to which language and frameworks to use)

Frontend Test

 Recommended time: 120 minutes

Requirements

- [React](#)
- [ChakraUI](#)

Brief

Read the full brief before starting. The brief is broken into separate tasks which include a recommended time investment. Complete as many tasks as possible while delivering an application free of issues.

Bingo card (est. 60 minutes)

1. When visiting the webpage:
 - a. the visitor enters a name
 - b. the React app will present a 5x5 grid and
 - c. a button to generate a new card.
2. Initially the grid will be empty save from the centre tile, labeled “FREE”. This tile will always be labeled free.
3. Clicking generate will randomly allocate the open tiles a unique number between 1 and 100.
4. In a game of Bingo, tiles are marked off as they are called. For your game, marking will be done by clicking the tile. Marked tiles should not completely hide their original number; it is up to you how that is shown.

5. Generating a new card will reset all marked tiles. If there are tiles marked when generating, an alert box will show to confirm their intention.

Sharing (est. 30 minutes)

The caller will need to confirm the winner of Bingo, but for reasons obvious, they can't request passing cards to the front. So the progress and card layout will need to be shared another way.

Create a share component that will allow the player to copy a unique URL which loads the current layout and marking.

Storage (est. 30 minutes)

Bingo players aren't used to modern technology and might be prone to closing their browser accidentally.

Ensure that subsequent loads of the page retains the layout and marking. Progress should be protected against refreshing the page and closing the browser.

Backend Test

 Recommended time: 75 minutes

Requirements

- [PHP](#) v8
- [Laravel](#) 10

Brief

Using backend technologies you will now extend the functionality of the Frontend application.

Number Generator (est. 15 mins)

1. Add a "Call next number" button
2. Clicking the button would show a random number 1-100
3. The same number cannot be called twice in a game of Bingo

Scoring (est. 60 mins)

1. If the user marks that same number on their card they are given 1 point. Marking the number on the card is not automatic and must be completed by the user. Validation that this is the correct number is required.
2. Score is calculated using the formula: $\text{score} = \text{abs}(100 - \text{<number of call button presses>})$
3. When all numbers have been marked a score is saved using an **API call** to a backend for the name i.e. [name, score]
4. We can then have a separate page that renders a leaderboard via an **API call** to the backend, listing name + score in a descending order

Delivery

All code must be presented via a GitHub repo. You may deliver the project by sending an email to Stefan Kovachev (stefan@antero.com.au) containing the link to a public git repository.

Please include instructions in your email explaining how the application can be started and viewed or better yet add this to the README.