

Technical Test

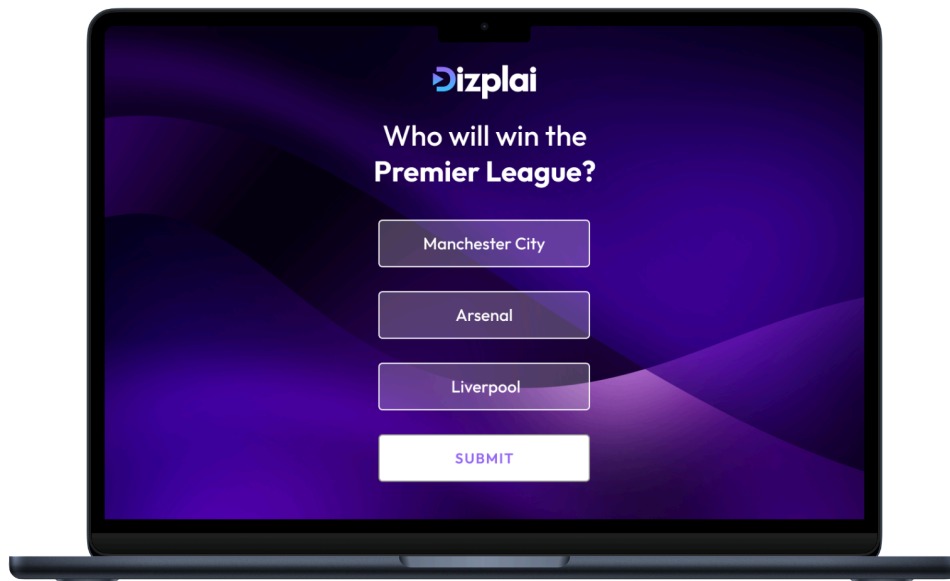
We require a very basic web app that can allow users to vote on a poll, see the example below. It will consist of two web pages, and a simple server exposing a RESTful API. You do not need to worry about creating polls, as the mechanism to create a poll and store it to a database already exists.

- It needs to have a readme on how to run it, and explain anything you did not have time to finish, or any assumptions that were made.
- The technical test only needs to run locally, but think about how this could be hosted somewhere.
- Show how you have tested both the frontend and backend to ensure it does not break. What methods or tools did you use?
- We would not be able to deploy this from a security perspective. What are the potential security problems? And, how could they be solved?
- We suggest a node.js express server for the api. However, use any technology that you think is best. Write down why you have chosen your selected technologies.
- Feel free to use better names than what has been provided.

Web App

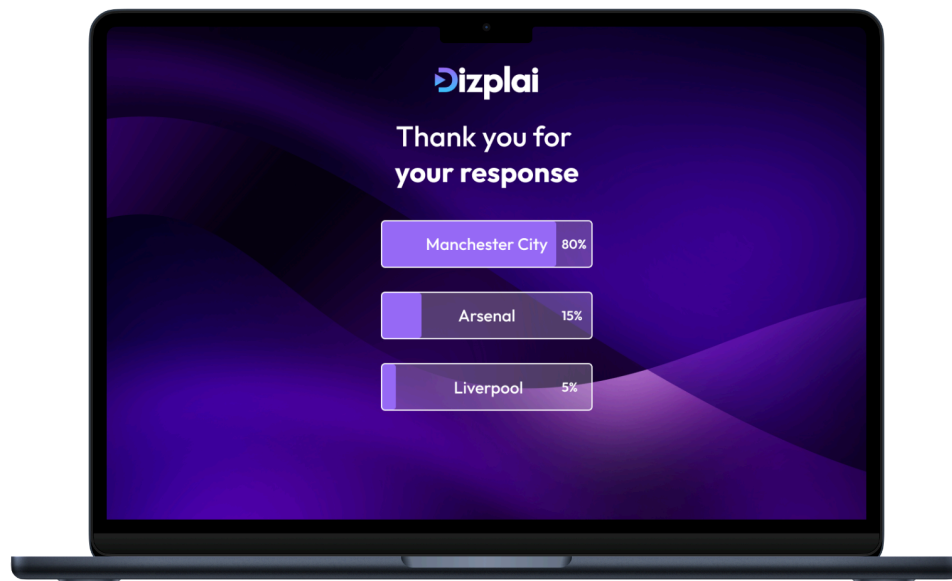
- This consists of at least 2 pages
 - The Voting Page, which is a page that will get the poll, and display the options. A user will be able to select one and submit a vote.
 - A Confirmation Page, which shows a thank you message and the percentage split of the votes for the options.
- Both pages must be responsive and work on both mobile and desktop devices.

Voting Page



- It must show which option a user has selected before they click submit.
- Only one option can be selected at a time.
- A user has to select an option to be able to submit a vote.

Confirmation Page



- Results should be ordered by most popular to least popular.
- Percentages should have no decimal places.

Colours

purple: #996BF9,

white: #FFFFFF

grey button border: #8C8C8C

Font

Outfit - <https://fonts.google.com/specimen/Outfit?query=outfit>

SERVER

- Two resources
 - Polls
 - Votes
- At a minimum, 3 actions will be needed. They are to:
 - get a Poll. This can just return the JSON of the example poll.
 - post a vote of what option the user selected. This will need to be stored locally for the next action. This could be stored in memory, but how could we do something more permanent?
 - get the votes for a poll by the poll's id. This is needed for the Confirmation Page.
- It must handle error scenarios gracefully. For example, what happens if you try and vote on a Poll that doesn't exist?

Provided example poll:

```
JavaScript
{
  "pollId": 1,
  "pollName": "Premier League Winner",
  "question": "Who will win the Premier League?",
  "options": [
    {
      "optionId": 1,
      "optionText": "Manchester City"
    },
    {
      "optionId": 2,
      "optionText": "Arsenal"
    },
    {
      "optionId": 3,
      "optionText": "Liverpool"
    }
  ]
}
```

A poll will have:

- a maximum of 5 options to choose from
- a minimum of 2 options to choose from
- requires a question

Submitting your work

Please send a link to a github repo hosting your project. Also, send the project as a .zip file in case something goes wrong.

If successful, you will demonstrate your work in the interview. Here you will also be asked questions about why you did things the way you did, and how you would do differently.