

Trivia App



React app with API

Your Mission

- Build a react application using [vite](#) and deploy with [netlify](#)
- You will build a Trivia app utilizing API
- Minimum requirements finish level 1&2

Main Features

Displaying questions: The application fetches a set of questions from hardcoded file (questions.js) OR external API and display it for the user.

Selecting answers: Users can select an answer for each question by clicking on the corresponding Answer.

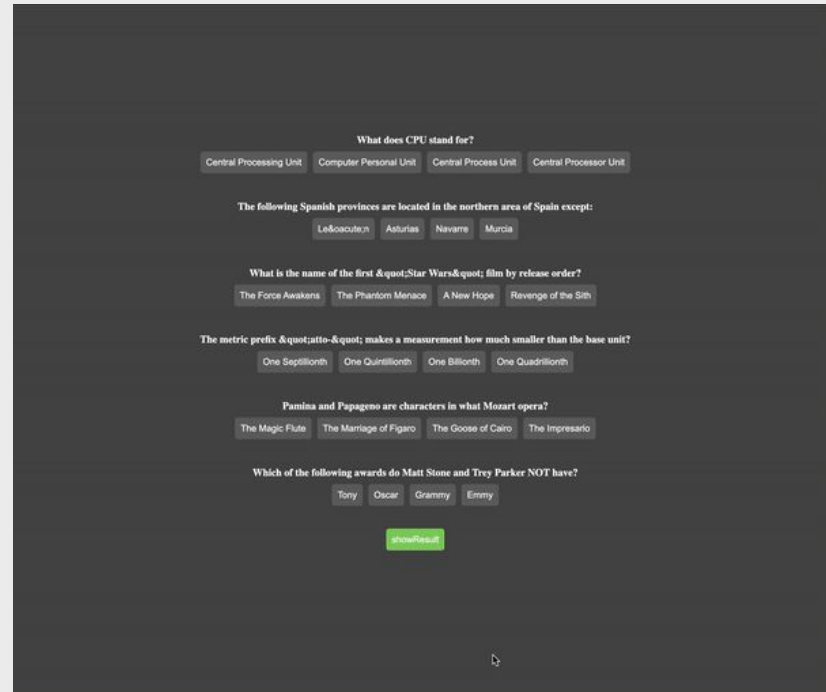
Tracking score: The application keeps track of the user's score based on their selected answers.

Reviewing answers: After completing all questions, users can review their answers along with the correct answers for each question.

Restarting the quiz: Users have the option to restart the quiz and retake it from the beginning.

Level1

- Using [questions.js](#) render questions with its answers and the user should be able to choose one answer per question
- once all questions have been answered then you should be able to view the final score with answer review
- User should be able to restart the app to retake the questions



Level2

- We want to get dynamic questions from API.
- Instead of using questions.js a hardcoded array use <https://wd40-trivia.onrender.com/>
- Please note the api is hosted on a free shared hosting which means if the app is not active it go to sleep and needs **30-40 seconds** to response back for the first time after sleep (so make sure you go to browser and wait for response then try to test with your react app)
- Main.jsx is using **<React.StrictMode>** that will cause 2 renders

Level3

- Now that all questions are coming from the API lets change the UI allowing only one question to be displayed at a time.
- Once all questions are answered show the final score with the questions review



Bonus Level (optional only if you finish level 3)

- Add animations or transitions to enhance the user experience during question transitions or interactions.
- Add sound effects or background music to make the app more engaging.

Styling

- Basic CSS
- [ReactBootstrap](#) (optional)

Resources

- We are utilizing <https://opentdb.com/api>
- Since the original api response is not structured in the way we want to use it that's why we built our own api that use it and customize the response