



Website Building: Weekend Assignment REACT – Voting Application

The following exercise contains the following subjects:

- HTML + CSS + JavaScript + React

Submitting instructions:

Please add the following:

- A link to the repository
- Free text – a description of the app. Stuff you found hard to implement, known bugs, and your assignment review.
- A link to a netlify deployment

Understanding the task:

- We are going to build a voting application in React.
- This application allows users to login, and vote for one of the parties.
- Each vote that a party gets - will be shown.
- The first party to reach X votes - wins!

The application will have to following sections:

- A Login Page – enter username and password

- A Voting Page – Where you have the buttons to vote for each party(dogs party, cat party, cow party and the lion party).
- An Admin Page – Where all the users are shown in a table, and the users who voted are marked.
Also, stats (35% of users voted, etc.)

Approaching the assignment:

Login Page:

- We will give you a built-in array of user objects (“db”). Each object will include: name, password, email, user type (“user”/” admin”)
- Login will include a stylized form, to enter the email and the password.
- If the email and password don’t match a specific user in the “db” - a “failed to log in” component will show.
- If the login is successful - we then move to the voting page, and a message with the name of the user (taken from the “db”)
- Design ideas:
 - Option
 - Option

Voting Page:

- The user’s name will be displayed.
- There will be 4 buttons, for each party: cats, dogs, cows, lions.

- All the total votes for each party will be shown, next to the party's image.
- The user can click on one of the buttons only once. After selection, the number of votes for this party will be updated.
- The voting data for each party should be shown for every login and every user - it needs to be persistent!
- After a vote, two new buttons will appear - "done" and "change vote".
- If "change": the party's vote will subtract 1, and the 4 buttons become active again.
The "done" and "change" buttons will disappear until another vote is made.
- After a vote is done:
 - If the user is an admin - move to the admin page, to see stats about the vote.
 - If the user is just a regular user, display a "thank you for voting" message, the buttons will disappear, and a "logout button" will be shown.
 - If the user performs a logout: all the user related data will be erased, and redirect him to the login page.
- Design ideas:
 - option/after vote
 - option

Admin Page:

- In the admin page, we can see all the users in a table. The table has 3 columns - name, email and "did vote".
- Give the rows for user who voted a different style.
- Show the total votes in this page too.
- Add logout button to the screen.
- When clicked - remove specific user data, load login page.
- Design ideas:
 - Option
 - Option

Tips:

- We are writing REACT code!
- This means that your components should be short, logical, reusable, dynamic and reactive.
- Don't just wrap html in JSX components!
- Take time for planning. Allocate time, priorities. Use github issues, milestones and projects! Track your time on each task
- Start from implementing the simple tasks, and then add complexity
- For the responsive design, you can use these principles:
 - <https://uxplanet.org/principles-of-mobile-site-design-c4c721693c42>

- https://www.thinkwithgoogle.com/_qs/documents/538/multi-screen-moblie-whitepaper_research-studies.pdf