CMPS 356 – Fall 2022
Web Applications Design and Development

Lab 12
Mutations

Objective

1. Using page parameters with Next 13
2. Rendering with Next 13
3. Creating and applying mutations using React Query
4. Next API endpoints for handling actions

Prerequisites

1. React Query mutations: `https://tanstack.com/query/v4/docs/guides/mutations`
2. Next API Endpoints: `https://nextjs.org/docs/api-routes/introduction`

1. Parameters and Rendering

1. Reuse the code from the previous lab.
2. Create a page for the teams and add a link in the header.
   - 2.1. Display the list of teams in a responsive box using chips.
   - 2.2. Create a sub-route for every team and list its matches.
   - 2.3. Add a not found handler for this sub-route.
3. Display the list of matches that the user has added to their schedule under `/schedule`.
   - 3.1. Add a corresponding link in the header.
   - 3.2. Display the total number of matches of the schedule in the header.
   - 3.3. Protect the schedule route using a password form that the user must submit to be able to access the page.
4. Build the application using SSG, SSR, and ISR.
   - 4.1. SSG, SSR, ISR: Update the `fetch` calls.
   - 4.2. SSG: Export a `generateStaticParams()` function generating the valid paths.
   - 4.3. ISR: Export a `revalidate` constant.

2. Mutations

1. Reuse the code from the previous exercise.
2. Use a store to keep track of a schedule of matches selected by the user.
3. Allow the user to add and remove a match from their schedule.
4. Allow the user to add a list of notes to every match in their schedule.
5. Allow the user to predict the teams for future matches. Use mutations and update the endpoints to support them.