CMPS 356 – Fall 2022
Web Applications Design and Development

Lab 11
Next 13 and Mutations

Objective

1. Next 13:
   1.1. Application directory
   1.2. Filename conventions: `page`, `layout`, `loading`, `error`, `template`, and `not-found`
   1.3. Root layout and nested layouts
   1.4. Routing using pages
   1.5. Links, fonts, and images
   1.6. Global application state
2. Data synchronization using React Query
3. Form validation and submission using React Hook Form

Prerequisites

1. Next 13: `https://beta.nextjs.org/docs`
2. Next 13 examples: `https://app-dir.vercel.app`
3. Next API Endpoints: `https://nextjs.org/docs/api-routes/introduction`
4. React Query: `https://tanstack.com/query/v4`
5. NextUI: `https://nextui.org`

1. New Features

1. Create a new directory `01-world-cup` and a Next 13 application under it: "`npx create-next-app@latest --experimental-app`".
   1.1. Install React Query: "`npm i @tanstack/react-query`".
   1.2. Install NextUI: "`npm i @nextui-org/react`" and provide it in the application.
2. Add two endpoints: "`/api/teams`" and "`/api/matches`" for serving the team and match data, respectively.
3. Update the root layout:
   3.1. Add a header, footer, and main content.
   3.2. Define and use `Header` and `Footer` components.
   3.3. Update your `Head` component.
   3.4. Display the World Cup 2022 logo in the header.
   3.5. Add a search input to the header.
   3.6. Add copyright information to the footer.
4. Display the list of groups and corresponding teams in a responsive grid using cards.

    4.1. Create a page for the groups.
    4.2. Add a link in the header.

5. Display the list of matches and corresponding teams in a responsive grid using cards.
    5.1. Create a page for the matches.
    5.2. Add a link in the header.
    5.3. Display the home/away teams, date/time, venue, and location for every match.
6. Display the list of teams and corresponding matches in a list using cards.
    6.1. Create a page for the matches.
    6.2. Add a link in the header.
7. Add loading interfaces and error handlers to your routes.
8. Build the app using SSG, SSR, and ISR
    8.1. Export a `generateStaticParams()` function
    8.2. Export a `revalidate` constant

2. Mutations

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