Rozgrzewka dla frontendowców:

1. Zapoznać się z **TailwindCSS**. Rezygnujemy całkowicie z pisania kodu CSS.
   Po co: przyspieszy nam proces stylizacji, nie musimy wymyślać nazw klas ani rozdzielać kodu na dwa pliki (tsx, css)

2. Przypomnieć sobie wzorce **Compound Component** i **Render Props**
   Po co: Pozwalą w łatwy sposób rozdzielić reużywalną logikę/UI od dynamicznej zawartości.
   Od strony praktycznej spójrz na przykłady gdzie te wzorce są wykorzystywane w już gotowych bibliotekach/frameworkach:
   https://ui.shadcn.com/docs/components/accordion
   https://ui.shadcn.com/docs/components/breadcrumb
   https://mui.com/material-ui/react-card/

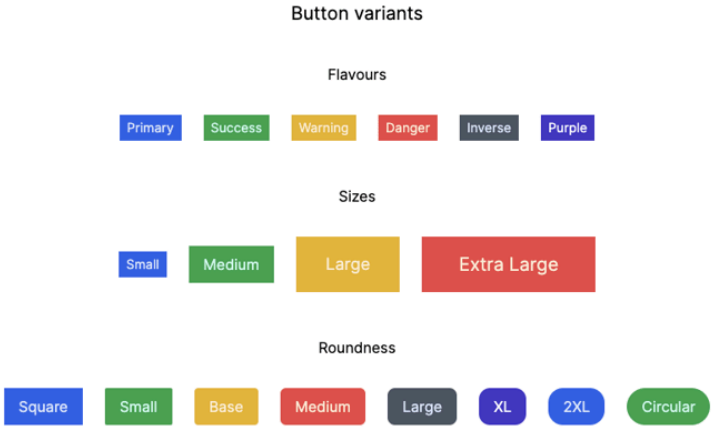
3. Zapoznać się z listą komponentów dostępnych w bibliotece **MUI**
   oraz w jaki sposób się z nich korzysta:
   https://mui.com/material-ui/all-components/
   Po co: będziemy tworzyć swoją własną bibliotekę UI, ale biblioteka MUI jest świetnym przykładem jak nazywać reużywalne komponenty UI oraz w jaki sposób z nich korzystać
   (jakie propsy przyjmują, z jakich wzorców korzystają).

4. Zapoznać się z dokumentacją **NextJS**, w tym:
   - **App Route**r (Page Router jest przestarzały),
   - Rola plików w app router **layout.tsx, page.tsx**, route.ts itd.
   - Komponenty **Server-Side** i **Client-Side**

5. Zapoznać się z dokumentacją **StoryBook**
   https://storybook.js.org/
   Po co: będziemy tworzyć testy wizualne dla komponentów UI tak, aby analizować ich różne warianty (wpływające na wygląd i zachowanie) w odizolowanym środowisku,
   bez konieczności używania ich w kodzie źródłowym.

6. Zapoznać się z biblioteką **CVA**
   https://cva.style/
   https://stackblitz.com/edit/joe-bell-cva-jxlmzf?file=src%2Fcomponents%2Fbutton%2Fbutton.tsx
   https://akhilaariyachandra.com/blog/using-clsx-or-classnames-with-tailwind-merge
   Po co: będziemy tworzyć różne warianty dla reużywalnych komponentów UI, np. przycisk może mieć różne warianty (kolor, rozmiar, zaokrąglanie itd.). Chcemy połączyć cva z tailwindCSS do łatwego ustalania wyglądu dla wariantów.

   ZADANIE NA ROZGRZEWKĘ:
   Stworzyć własny projekt NextJS + TypeScript + TailwindCSS + StoryBook + cva
   a następnie stwórz reużywalny przycisk, który może posiadać poniższe warianty…
   Utwórz do tego odpowiednie historyjki za pomocą Story Booka.



Rozgrzewka dla backendowców:

1. Zapoznać się z dokumentacją NextJS, w tym:
    - **App Route**r (Page Router jest przestarzały),
    - Rola plików w app router layout.tsx, page.tsx, **route.ts** itd.
    - Komponenty **Server-Side** i **Client-Side**
    - Zapoznać się z środowiskiem **Edge**
    - Zapoznać się z rolą pliku **middleware.ts**

2. Zapoznać się z **NextAuth@5-beta**
   https://authjs.dev/getting-started/installation?framework=Next.js

   Po co: pozwoli nam w łatwy i przystępny sposób stworzyć autoryzację za pomocą różnych metod.

3. Zapoznać się z **PayloadCMS**
   https://payloadcms.com/

   Po co: Pozwoli nam w łatwy sposób zbudować panel admina + api na podstawie schematów kolekcji. PayloadCMS od teraz jest ściśle związany z NextJS

   ZADANIE NA ROZGRZEWKĘ:
   Stwórz własny projekt NextJS + PayloadCMS, W PayloadCMS niech będzie jedna prosta kolekcja Products, każdy produkt niech ma nazwę, cenę oraz obrazek.

   Dodaj parę przykladowych produktow za pomocą panelu admina oraz wyświetl na froncie listę produktów. Jeśli Ci się uda zrobić to w szybkim tempie to spróbuj zintegrować NextJS z NextAuth tak, aby można było się logować za pomocą dowolnej metody.