1: Create a new Angular project with routing.

2: Add Tailwind

```
npm install tailwindcss@latest @ngneat/tailwind@latest
npx tailwindcss init

/* Add these to the root stylesheet */
@import "tailwindcss/base";
@import "tailwindcss/components";
@import "tailwindcss/utilities";

// Add the following to tailwind.config.js
module.exports = {
        content: [],
 },
purge: {
        content: ['./src/**/*.{html,ts}'],
        },
};
```

3: Create 2 layouts

Create 2 different layouts, each with their own module and routes, so the main app module has routes, and each module has its own routes, with their own layouts. The design of these layouts is not important.

4: Services

Create a service: api.service.ts. This will just have methods that return Observables of mock data, as if you were calling an HTTP API that returns a list of objects to display in your UI. Create one called ‘getCards’ that returns an observable list of objects like this:

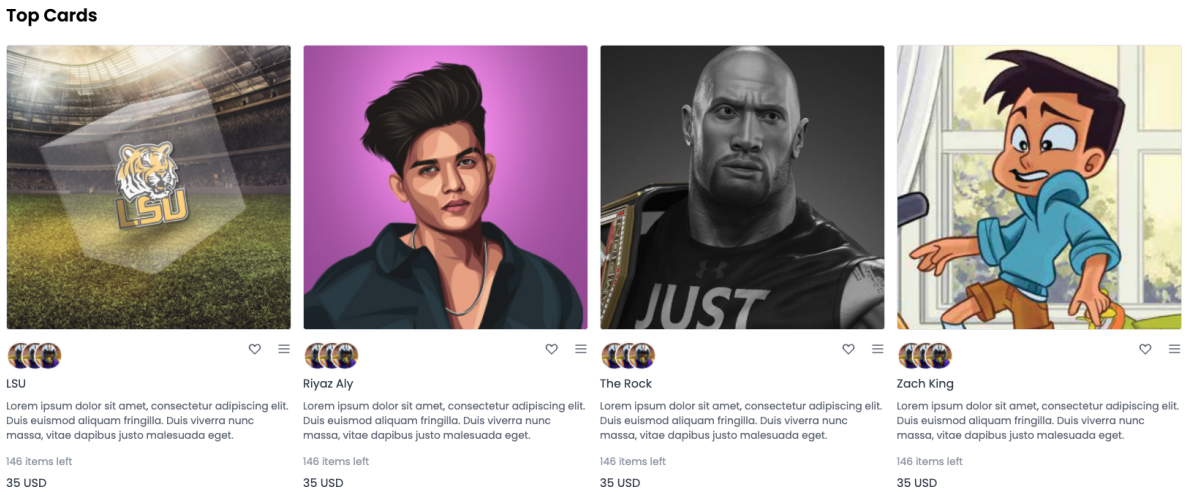
```
{
        imageUrl: string;
        price: number;
        name: string;
        description: string;
        itemCount: number;

         // This will be a timestamp that you can create from random dates
        createdDate:number;
}
```

Create a second method that returns a Promise of the exact same list of objects.

5: Components

Create a component that belongs to one of the submodules that implements **AfterViewInit,** This component will be a container for a list of child components that should like this:



This component should accept the list of objects from the API service as Input.

Call the component <top-cards>

Now create the card component that will be rendered inside of the top cards component that looks like one of the cards above, please use Tailwind classes to create the card design above. This layout needs to be responsive so that on a mobile device 2 columns of cards are displayed. Please use Tailwind CSS classes for the layout and the card design.

I would like the cards to be sorted by the *createdDate* in descending order, please do this inside the TopCards component, inside the ngAfterViewInit hook.

Finally I would like you to map the list of objects that you get from the api service in your top cards component to a list of objects that look like this:

```
{
        imageUrl: string;
        price: number;
        name: string;
        description: string;
        itemCount: number;
        // This is the only property that is different, you can make this a random timestamp
        updatedDate:number;

}
```

Please console.log the result of this map.