miss-toy !(yourToy)

Setup

Setup the project folder: *misstoy*-proj

Notes:

- Use Git for version management, the Git log should be meaningful and present the progress of the development work.
- Use async-await, try-catch across your app

Part 1 – Getting started

Here is an initial model:

```
const labels = ['On wheels', 'Box game', 'Art', 'Baby', 'Doll', 'Puzzle',
'Outdoor', 'Battery Powered']

const toy = {
    _id: 't101',
    name: 'Talking Doll',
    price: 123,
    labels: ['Doll', 'Battery Powered', 'Baby'],
    createdAt: 1631031801011,
    inStock: true,
}
```

Build your frontend from scratch.

- Use the **CLI** to create this project
- Commit and push the code
- implement full CRUD, manage your state with a store.

You should have the following:

1. store
2. toyService
    a. We kick off the frontend first using a service that works with storageService which provides an async access (CRUDL) on a collection kept to the browser's localStorage)

3. <ToyDetails> (Smart, Routable)
    a. This page renders full details about the toy
4. <ToyEdit> (Smart, Routable)
5. <ToyIndex> (Smart, Routable)
    a. <ToyList>
    b. <ToyPreview>
    c. <ToyFilter>
        i. By name (use debounce)
        ii. In stock (remember that there are 3 states here)
        iii. Toy label multiselect dropdown
        iv. Sort by: name / price / created

(git) commit your job: "Frontend basic functionality"

Part 3 – Beautiful miss-toy with SCSS

- Use a full SCSS architecture
- Convert your CSS to SCSS
    - Use nesting
    - Use variables
    - Use mixins
    - Use functions
- Make it look amazing on desktop, tablet and mobile

Part 4

Story

- We need shop owner (admin) to be able to manage the shop
- We need normal user to be able to add msgs about toys

Support authentication

- Use the provided user.service, auth.service to manage users (_id, fullname, username, password, isAdmin), have one admin user (isAdmin: true)
- Add a login page
- Only admin user should have the Edit/Delete/Add options.

Add msgs support

- Inside your toy, add a msgs array

```
const toy = {
    _id: "t101",
    name: "Talking Doll",
    price: 123,
    labels: ["Doll", "Battery Powered", "Baby"],
    createdAt: 1631031801011,
    inStock: true,
    msgs: [
        {
            id: 'm101',
            txt: 'Great toy, how much',
            by: {
                _id: 'u101',
                fullname: 'Puki Ga'
            }
        }
    ]
}
```

- In ToyDetails
    - show the current toy's msgs
    - Allow logged-in user to enter a msg

Part 5 – PWA

Add PWA support for your project

Tasks

- Update the manifest
- Bonus: offline support
    - Files are automatically cached by service worker
    - Cache data in localStorage