Mandatory Exercise 1 - UX design

The deadline for this exercise is Friday, February 1, 08:59.

For this **mandatory exercise** you should work on **master branch only**.

Preparation

1. Create a new repository on GitHub called **mandatory-uxdesign1**.
2. Follow the instructions that GitHub gives you; Create a local repository and add a remote or clone the newly created repository.

Submission

When you submit the exercise in PingPong, before the deadline, you will enter a link to your repository, such as:

https://github.com/mygithubusername/mandatory-uxdesign1

The teacher will look in the **master branch**. If any commits are done to the branch after the deadline, the teacher will look at the last commit before the deadline.

You will get one of the grades **G** or **IG**.

The submission should contain the following source files

- **mds.css**
- **mds.js**
- **demo.html**

Instructions

In this exercise you will be creating a simple UI component library, implementing four form elements from Material Design. We will be calling the library “MD Simple”.

The elements to be implemented are

- Text field
  https://material.io/design/components/text-fields.html

- Switch
  https://material.io/design/components/selection-controls.html#switches
- Checkboxes
  https://material.io/design/components/selection-controls.html#checkboxes
- Radio buttons
  https://material.io/design/components/selection-controls.html#radio-buttons

CSS library

Create a CSS file named “mds.css”. The file should define styles for the follow components using BEM.

- **.mds-text-field**
- **.mds-switch**
- **.mds-checkbox**
- **.mds-radio**

Please refer to the documentation for information on how to create CSS components using BEM.

http://getbem.com/naming/

JavaScript Library

Create a JavaScript file named “mds.js” that defines an object called **mds**. The object should define the following methods

- **mds.textField(element)**
- **mds.switch(element)**
- **mds.checkbox(element)**
- **mds.radioButton(element)**

These methods instantiate the element, adding functionality if necessary. If no JavaScript is necessary, please leave the methods empty.

For this exercise only JavaScript code to control behavior of the text field label is necessary.

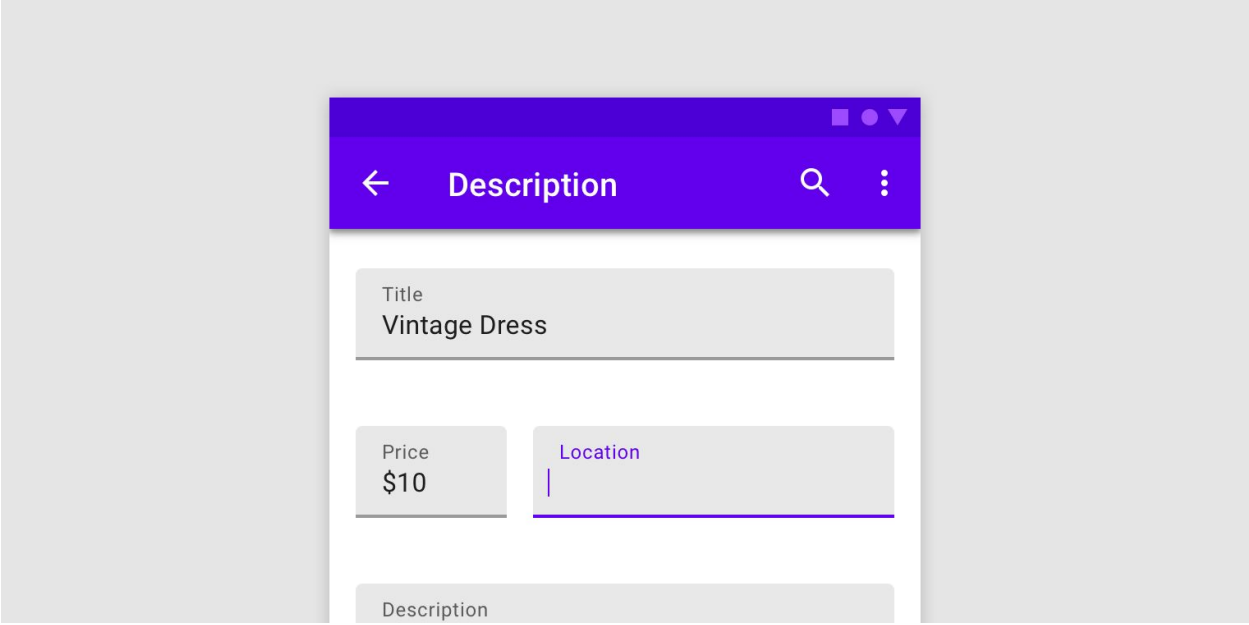
Optional: Return a function that can be used to remove all event listeners (if any) added by the method. This is to avoid memory leaks when using the components.

Demo page

Create a HTML file named “demo.html” that displays all the components. The page should at least contain a text field, a switch, a checkbox and two or more radio buttons.

Text field

Implement a Material Design textfield. You only need to implement the filled variant.



https://material.io/design/components/text-fields.html#

You will probably need an element for the placeholder text since it’s not possible to recreate the placeholder using the “placeholder” attribute on the “input” element.

Make sure that the behavior of the placeholder/label text is consistent with the Material Design Specification.

The following is mandatory

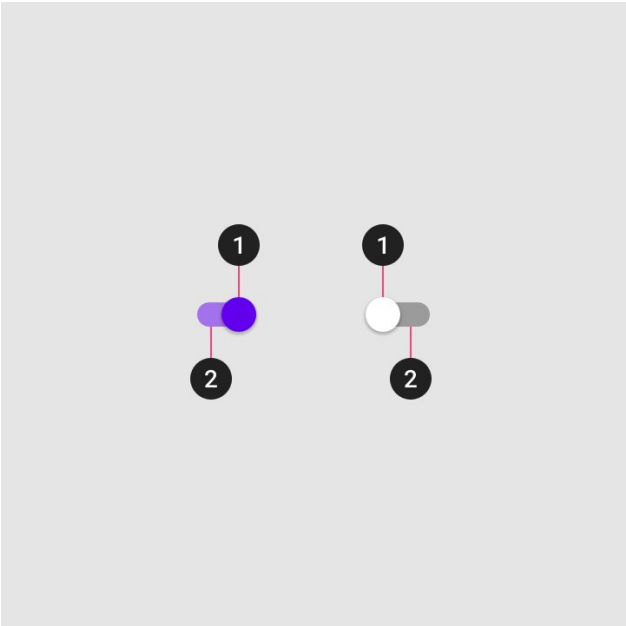
- Implement the text field in accordance with the Spec
- Animated placeholder/label text
- Animated underline
- Disabled state

The following is optional

- Helper text
- Error message
- Icons
- Character counter
- Multi-line text field
- Textarea

Switch

Implement a Material Design switch.



https://material.io/design/components/selection-controls.html#switches

The following is mandatory

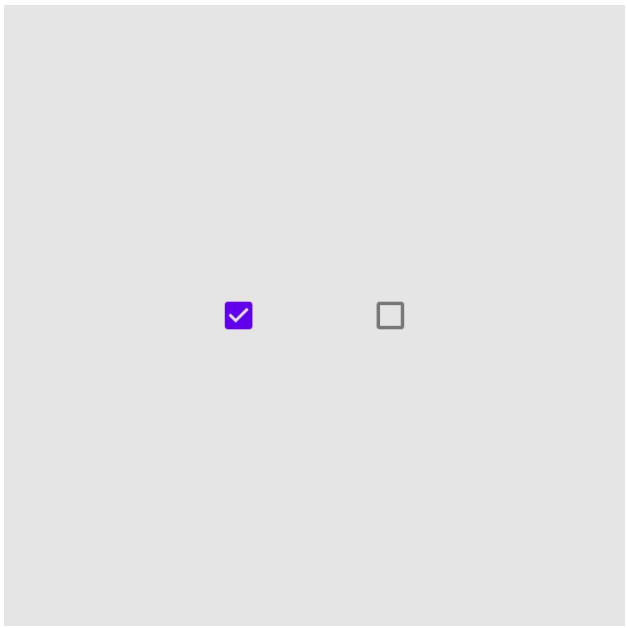
- Implement the switch in accordance with the Spec
- Animated switch toggle
- Disabled state

The following is optional

- Make the switch draggable/swipeable
- Display processing status

Checkbox

Implement a Material Design checkbox.



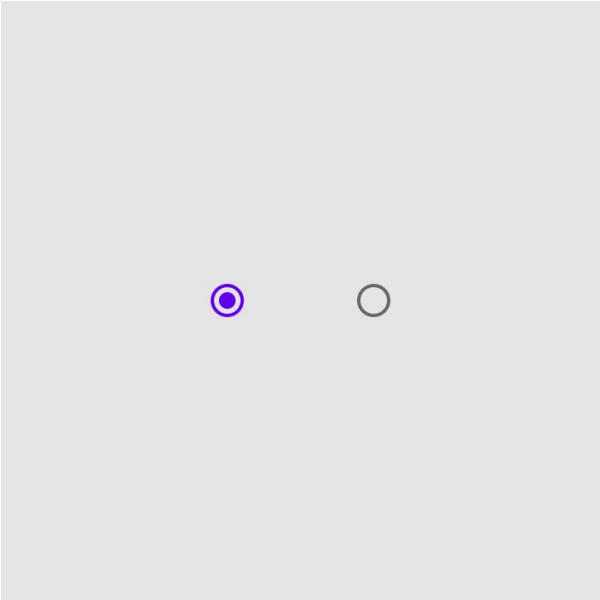
https://material.io/design/components/selection-controls.html#checkboxes

The following is mandatory

- Implement a checkbox in accordance with the spec
- Disabled state

Radio button

Implement a Material Design radio button



https://material.io/design/components/selection-controls.html#radio-buttons

The following is mandatory

- Implement a radio button in accordance with the Spec
- Disabled state

Tips

- Use native form elements as the basis for your components. E.g. use a native checkbox for the switch and checkbox components
- Create BEM “elements” when necessary
- Use “BEM” modifiers for different component states
- Make sure the order and structure of your elements makes sense
- Only use JavaScript when it’s absolutely necessary
- Ask questions if anything is unclear
- Test your components thoroughly
- Good luck!