1.

Question 1

What of the next input types doesn’t have a controlled version when they are used in React?

<input type=”text” />

<input type=”file” />

<textarea />

2.

Question 2

What are some of the features of controlled components? Select all that apply

Conditionally disabling the submit button.

Validating all values in the client side when a submission occurs in the form, before calling the server endpoint.

Enforcing a specific input format.

3.

Question 3

How do you get the value of an input when its state is handled by the DOM (Uncontrolled)? Select all that apply.

Using a ref via useRef hook, assigning it to the input and then reading the input value when the submission happens via ref.current.value.

Using a combination of useEffect and useRef hooks, where a ref is used on the uncontrolled input and then its value can be read on useEffect after a re-render cycle happens.

Using local state and initializing it to an empty string. Then, reading the input value from the event object when the submission happens and finally setting the local state with that value.

4.

Question 4

What happens when you click on the submit button in the below code snippet?

<form onSubmit={() => alert("Submitting")}>

<input type="text" value={text} onChange={e => setText(e.target.value)} />

<input type="button" value="Submit" />

</form>

The onSubmit callback is executed and an alert is shown with the text "Submitting".

An error is thrown.

Nothing happens when the button is clicked.

5.

Question 5

What is missing in the below code for the select component to work properly?

<select onChange={handleChange}>

<option value="grapefruit">Grapefruit</option>

<option value="lime">Lime</option>

<option value="coconut">Coconut</option>

<option value="mango">Mango</option>

</select>

Each option tag should be accompanied by a label tag.

Each option tag should have an onChange handler.

The select tag is missing a value prop.