21.Handling Form submission

```
<form>
    <div className='new-expense__controls'>
        <div className="new-expense__control">
            <label>Title</label>
            <input type='text' onChange={titleChangeHandler} />
        </div>

        <div className="new-expense__control">
            <label>Amount</label>
            <input type="number" min="0.01" step="0.01" onChange=
        </div>

        <div className="new-expense__control">
            <label>Date</label>
            <input type="date" min="2020-01-01" max="2023-12-31"
        </div>
    </div>

    <div className="new-expense__actions">
        <button type="submit" >Add Expense</button>
    </div>
</form>
)
```

- To handle the form submission, we could add a event listener 'click' for that submit button
- But that's not the best practice
- By default, whenever a button of type submit is pressed automatically the form element emits an event of 'Submit'
- So we can use event listener for form instead of using it in the button

```
21        return (
22            <form onSubmit={}>
23                <div className='new-expense__control
24                    <div className="new-expense__con
25                        <label>Title</label>
26                        <input type='text' onChange=
```

```
20
21    const submitHandler=()=>{}
22
23    return (
24        <form onSubmit={submitHandler}>
25            <div className='new-expense__controls'>
26                <div className="new-expense__control">
27                    <label>Title</label>
28                    <input type='text' onChange={title
```

- **One important point to keep in mind is that if we submit the form, our browser by default sends a new request(POST/GET) to the server, but here we want to handle these values manually in react**
- **so to avoid this we could use preventdefault to avoid sending a request to the server**
-

```
20                              (parameter) event: an
21      const submitHandler=(event)=>{
22          event.preventDefault();
23      }
24
25      return (
26          <form onSubmit={submitHandler}>
```

  - **Here we are passing in the event parameter sent by the onSubmit props**
  - **Then we are using the preventDefault() function to avoid the request being send to the server**
  - **After this we could do whatever we want to do with these values**

```
21      const submitHandler=(event)=>{
22          event.preventDefault();
23
24          const expenseData={
25              title:enteredTitle,
26              amount:enteredAmount,
27              date:new Date(enteredDate)
28          }
29          console.log(expenseData);
30      }
```

  - **Here we are creating a new object and printing the object**

-