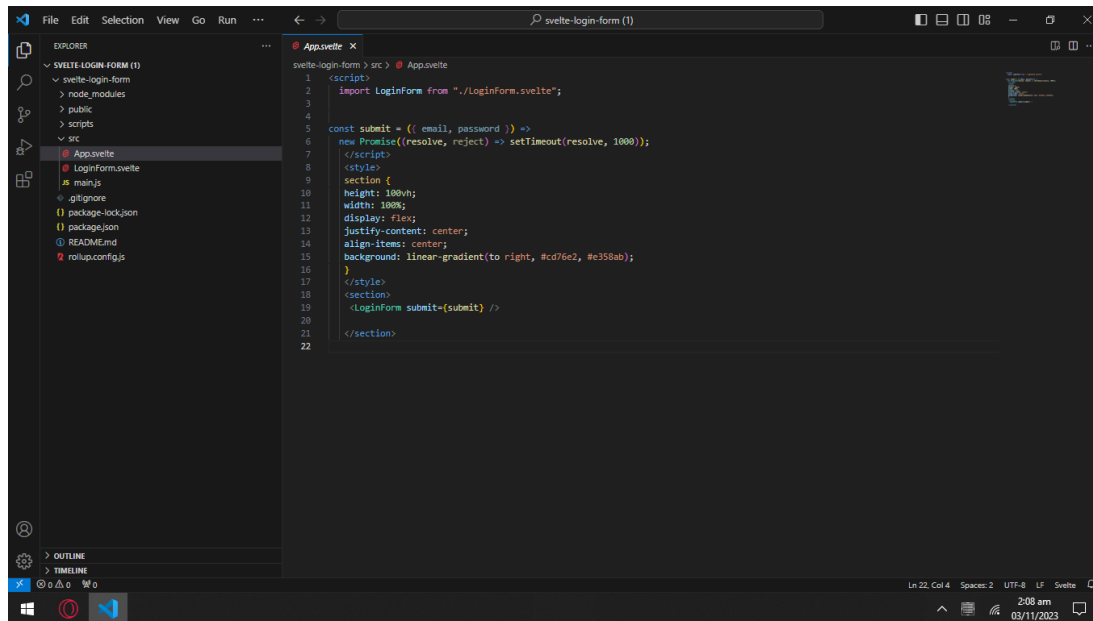


SVELTE

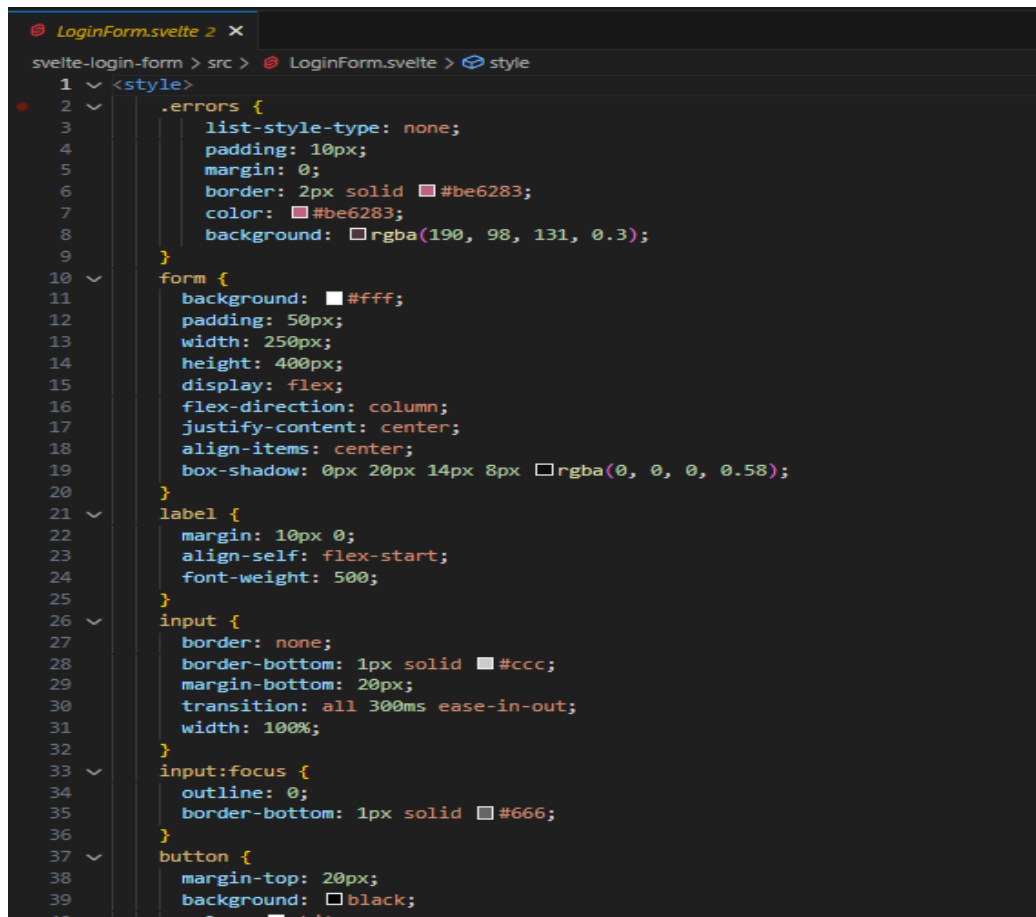
App.svelte



```
1 <script>
2   import LoginForm from './LoginForm.svelte';
3
4
5   const submit = ({ email, password }) =>
6     new Promise((resolve, reject) => setTimeout(resolve, 1000));
7
8   </script>
9   <style>
10    section {
11      height: 100vh;
12      width: 100%;
13      display: flex;
14      justify-content: center;
15      align-items: center;
16      background: linear-gradient(to right, #cd7be2, #e358ab);
17    }
18   </style>
19   <section>
20     <LoginForm submit={submit} />
21   </section>
22
```




LoginForm.svelte

<style>



```
1 <style>
2   .errors {
3     list-style-type: none;
4     padding: 10px;
5     margin: 0;
6     border: 2px solid #be6283;
7     color: #be6283;
8     background: rgba(190, 98, 131, 0.3);
9   }
10  form {
11    background: #fff;
12    padding: 50px;
13    width: 250px;
14    height: 400px;
15    display: flex;
16    flex-direction: column;
17    justify-content: center;
18    align-items: center;
19    box-shadow: 0px 20px 14px 8px rgba(0, 0, 0, 0.58);
20  }
21  label {
22    margin: 10px 0;
23    align-self: flex-start;
24    font-weight: 500;
25  }
26  input {
27    border: none;
28    border-bottom: 1px solid #ccc;
29    margin-bottom: 20px;
30    transition: all 300ms ease-in-out;
31    width: 100%;
32  }
33  input:focus {
34    outline: 0;
35    border-bottom: 1px solid #666;
36  }
37  button {
38    margin-top: 20px;
39    background: black;
40    color: white;
41  }
42
```




```

39     background:  black;
40     color:  white;
41     padding: 10px 0;
42     width: 200px;
43     border-radius: 25px;
44     text-transform: uppercase;
45     font-weight: bold;
46     cursor: pointer;
47     transition: all 300ms ease-in-out;
48   }
49   button:hover {
50     transform: translateY(-2.5px);
51     box-shadow: 0px 1px 10px 0px  rgba(0, 0, 0, 0.58);
52   }
53   h1 {
54     margin: 10px 20px 30px 20px;
55     font-size: 40px;
56   }
57 </style>
58

```

<form>

```

60 <form on:submit|preventDefault={handleSubmit}>
61   {#if isSuccess}
62     <div class="success">
63       
64       <br />
65       You've been successfully logged in.
66     </div>
67   {:else}
68     <h1> </h1>
69
70     <label>Email</label>
71     <input name="email" placeholder="name@example.com" bind:value={email} />
72
73     <label>Password</label>
74     <input name="password" type="password" bind:value={password} />
75
76     <button type="submit">
77       {#if isLoading}Logging in...{:else}Log in  {:if}
78     </button>
79
80     {#if Object.keys(errors).length > 0}
81     <ul class="errors">
82       {#each Object.keys(errors) as field}
83       <li>{field}: {errors[field]}</li>
84     {/each}
85     </ul>
86   {:/if}
87 {/if}
88 </form>

```

<script>

```
90 <script>
91   export let submit;
92   let email = "";
93   let password = "";
94   let isLoading = false;
95   let isSuccess = false;
96   let errors = {};
97
98
99   const handleSubmit = () => {
100     errors = {};
101     if (email.length === 0) {
102       errors.email = "Field should not be empty";
103     }
104     if (password.length === 0) {
105       errors.password = "Field should not be empty";
106     }
107     if (Object.keys(errors).length === 0) {
108       isLoading = true;
109       submit({ email, password })
110         .then(() => {
111           isSuccess = true;
112           isLoading = false;
113         })
114         .catch(err => {
115           errors.server = err;
116           isLoading = false;
117         });
118     }
119   };
120 </script>
121
```

Source Code:

App.svelte

```
<script>
    import LoginForm from "./LoginForm.svelte";

const submit = ({ email, password }) =>
    new Promise((resolve, reject) => setTimeout(resolve, 1000));
</script>
<style>
    section {
        height: 100vh;
        width: 100%;
        display: flex;
        justify-content: center;
        align-items: center;
        background: linear-gradient(to right, #cd76e2, #e358ab);
    }
</style>
<section>
    <LoginForm submit={submit} />

</section>
```

LoginForm.svelte

```
<style>
<style>
    .errors {
        list-style-type: none;
        padding: 10px;
        margin: 0;
        border: 2px solid #be6283;
        color: #be6283;
        background: rgba(190, 98, 131, 0.3);
    }
    form {
        background: #fff;
        padding: 50px;
        width: 250px;
        height: 400px;
        display: flex;
        flex-direction: column;
```

```
    justify-content: center;
    align-items: center;
    box-shadow: 0px 20px 14px 8px rgba(0, 0, 0, 0.58);
}
label {
    margin: 10px 0;
    align-self: flex-start;
    font-weight: 500;
}
input {
    border: none;
    border-bottom: 1px solid #ccc;
    margin-bottom: 20px;
    transition: all 300ms ease-in-out;
    width: 100%;
}
input:focus {
    outline: 0;
    border-bottom: 1px solid #666;
}
button {
    margin-top: 20px;
    background: black;
    color: white;
    padding: 10px 0;
    width: 200px;
    border-radius: 25px;
    text-transform: uppercase;
    font-weight: bold;
    cursor: pointer;
    transition: all 300ms ease-in-out;
}
button:hover {
    transform: translateY(-2.5px);
    box-shadow: 0px 1px 10px 0px rgba(0, 0, 0, 0.58);
}
h1 {
    margin: 10px 20px 30px 20px;
    font-size: 40px;
}
</style>
```

<form>

```
<form on:submit|preventDefault={handleSubmit}>
  {#if isSuccess}
    <div class="success">
      
      <br />
      You've been successfully logged in.
    </div>
  {:else}
    <h1>  </h1>

    <label>Email</label>
    <input name="email" placeholder="name@example.com" bind:value={email} />

    <label>Password</label>
    <input name="password" type="password" bind:value={password} />

    <button type="submit">
      {#if isLoading}Logging in...{:else}Log in  {:if}
    </button>

    {#if Object.keys(errors).length > 0}
      <ul class="errors">
        {#each Object.keys(errors) as field}
          <li>{field}: {errors[field]}</li>
        {/each}
      </ul>
    {:if}
  {:if}
</form>
```

<script>

```
<script>
  export let submit;
  let email = "";
  let password = "";
  let isLoading = false;
  let isSuccess = false;
  let errors = {};

  const handleSubmit = () => {
    errors = {};
    if (email.length === 0) {
```

```
        errors.email = "Field should not be empty";
    }
    if (password.length === 0) {
        errors.password = "Field should not be empty";
    }
    if (Object.keys(errors).length === 0) {
        isLoading = true;
        submit({ email, password })
            .then(() => {
                isSuccess = true;
                isLoading = false;
            })
            .catch(err => {
                errors.server = err;
                isLoading = false;
            });
    }
};
</script>
```

Output:

