User-Form

https://github.com/alienbrains/User-Form-Manager/

Src

App.js

```
import React from "react";
import Forms from "./Components/UserForm";

import "./App.css";

function App() {
 return <Forms />;
}

export default App;
```

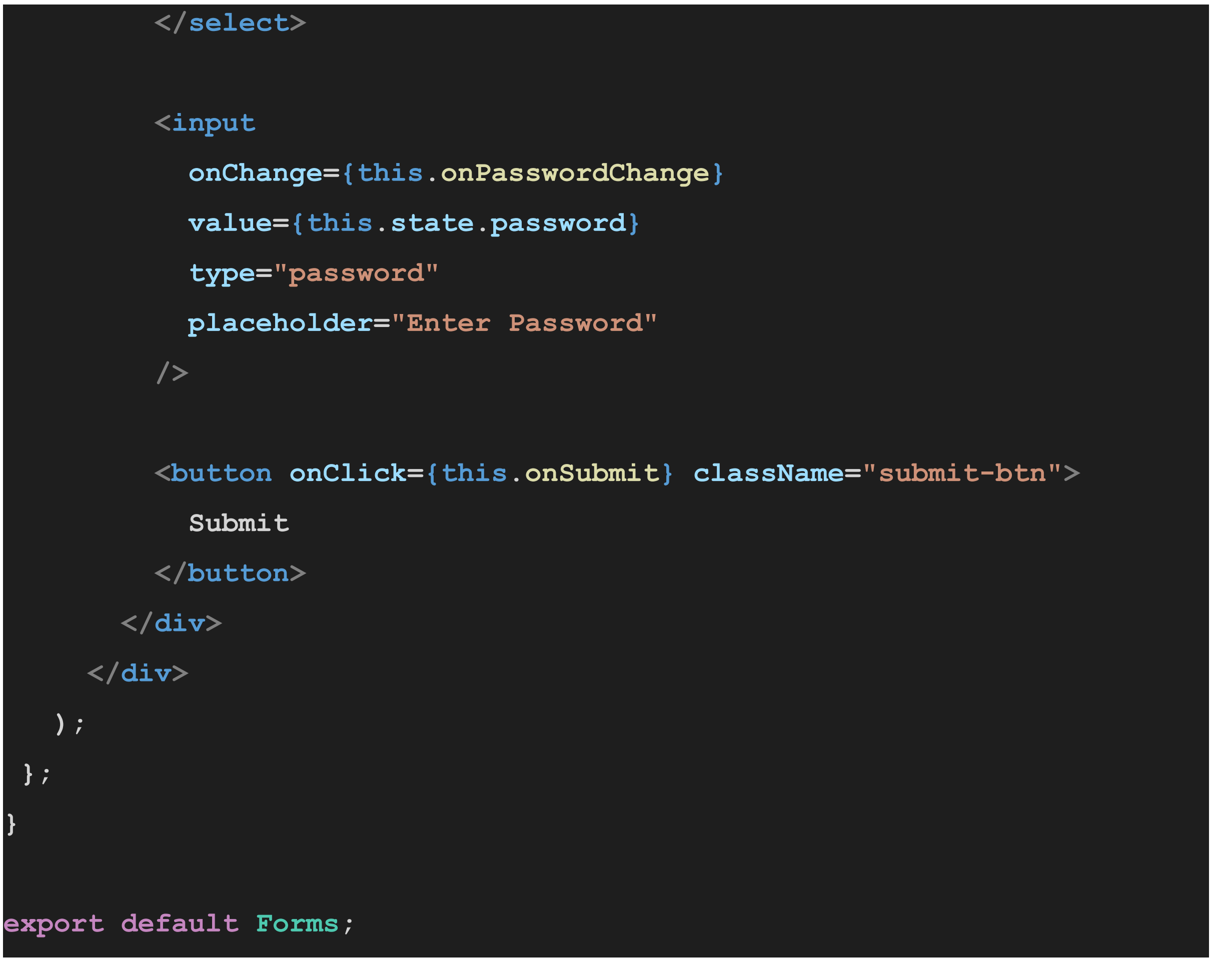
This is the initial App.js Component which is a functional component.

Components

Components/UserForm/index.js

```
import React from "react";

import "./style.css";

class Forms extends React.Component {
 constructor(props) {
   super(props);
   this.state = {
     name: "",
     email: "",
     phoneNumber: "",
     password: "",
     gender: ""
   };
 }

 //  executed when something changes in name input field
 onNameChange = (e) => {
   this.setState({
     name: e.target.value
   });
 };

 //  executed when something changes in email input field
 onEmailChange = (e) => {
   this.setState({
     email: e.target.value
   });
 };

 //  executed when something changes in phone input field
```

```
onPhoneChange = (e) => {
  this.setState({
    phoneNumber: e.target.value
  });
};

//  executed when something changes in password input field
onPasswordChange = (e) => {
  this.setState({
    password: e.target.value
  });
};

//  executed when something changes in gender select field
onGenderChange = (e) => {
  this.setState({
    gender: e.target.value
  });
};

//  executed when user click on submit button to submit the form
onSubmit = () => {
  const data = `
  Name: ${this.state.name},
  Email: ${this.state.email},
  Phone: ${this.state.phoneNumber},
  Gender: ${this.state.gender},
  Password: ${this.state.password},
  `;

  window.alert(data);
  this.clearForm();
};

//  clears the form
clearForm = () => {
  this.setState({
    name: "",
    email: "",
    phoneNumber: "",
    password: "",
    gender: ""
  });
};

render = () => {
  return (
    <div className="container">
      <div className="form-container">
        <p className="title">-------- User Form -----------</p>
        <input
          onChange={this.onNameChange}
          value={this.state.name}
          type="text"
          placeholder="Enter name"
        />
        <input
          onChange={this.onEmailChange}
          value={this.state.email}
          type="email"
          placeholder="Enter Email"
        />
        <input
          onChange={this.onPhoneChange}
          value={this.state.phoneNumber}
          type="text"
          placeholder="Enter Phone number"
        />

        <select onChange={this.onGenderChange}
value={this.state.gender}>
          <option value="">Choose Gender</option>
          <option value="male">Male</option>
          <option value="female">Female</option>
```



```
          </select>

          <input
            onChange={this.onPasswordChange}
            value={this.state.password}
            type="password"
            placeholder="Enter Password"
          />

          <button onClick={this.onSubmit} className="submit-btn">
            Submit
          </button>
        </div>
      </div>
    );
  };
}

export default Forms;
```

In the **Forms** component there are 4 <input /> tags.
The first one is for the user name, the second one is for the email address, the third one is for the phone number and the fourth one is for the password.

The first <input /> is of type = “text” , which means it accepts any text input. When we start typing on this input field, the **onChange** function is invoked which calls the function **this.onNameChange.**
this.onNameChange receives an object event (which is ‘e’ in this code). The name which we type gets stored in the key **event.target.value** , we store this **event.target.value** in the state variable **name** using the **setState** function.

The second <input /> is of type = “email” , which means it accepts any text input which is like an email address. When we start typing on this input field, the **onChange** function is invoked which calls the function **this.onEmailChange.**
this.onEmailChange receives an object event (which is ‘e’ in this code). The email which we type gets stored in the key **event.target.value** , we store this **event.target.value** in the state variable **email** using the **setState** function.

The third <input /> is of type = “text” , which means it accepts any text input. When we start typing on this input field, the **onChange** function is invoked which calls the function **this.onPhoneChange.**
this.onPhoneChange receives an object event (which is ‘e’ in this code). The phone number which we type gets stored in the key **event.target.value** , we store this **event.target.value** in the state variable **phone** using the **setState** function.

The fourth <input /> is of type = “password” , which means it accepts any text input as a password and doesn’t show what you have typed. When we start typing on this input field, the **onChange** function is invoked which calls the function **this.onPasswordChange.**
this.onPasswordChange receives an object event (which is ‘e’ in this code). The password which we type gets stored in the key **event.target.value** , we store this **event.target.value** in the state variable **password** using the **setState** function.

The <select /> tag provides two options to select from it Male and Female.
When we select an option the **onChange** function is invoked which calls the function **this.onGenderChange.**
this.onGenderChange receives an object event (which is ‘e’ in this code). The option which we select gets stored in the key **event.target.value** , we store this **event.target.value** in the state variable **gender** using the **setState** function.

The <button /> tag is a submit button. When we click on the button the function **this.onSubmit** is called which shows all the data, we have entered, as an alert in the window. After that from there we call the function **this.clearForm** which resets the state values.