TASK

User Registration Form

Create a user registration form with validation.

You are building a user registration form for a new web application. The form should collect the following information from the user:

First Name
Last Name
Email
Password
Confirm Password

Create a form with the necessary input fields.
1.Add basic validation for required fields and email format.
2.Ensure that the password and confirm password fields match.
3.Display error messages for invalid inputs.
4.Implement a success message upon successful form submission.

Booking Form

Create a booking form for scheduling appointments.

You are developing a booking form for scheduling appointments at a clinic. The form should collect the following information:

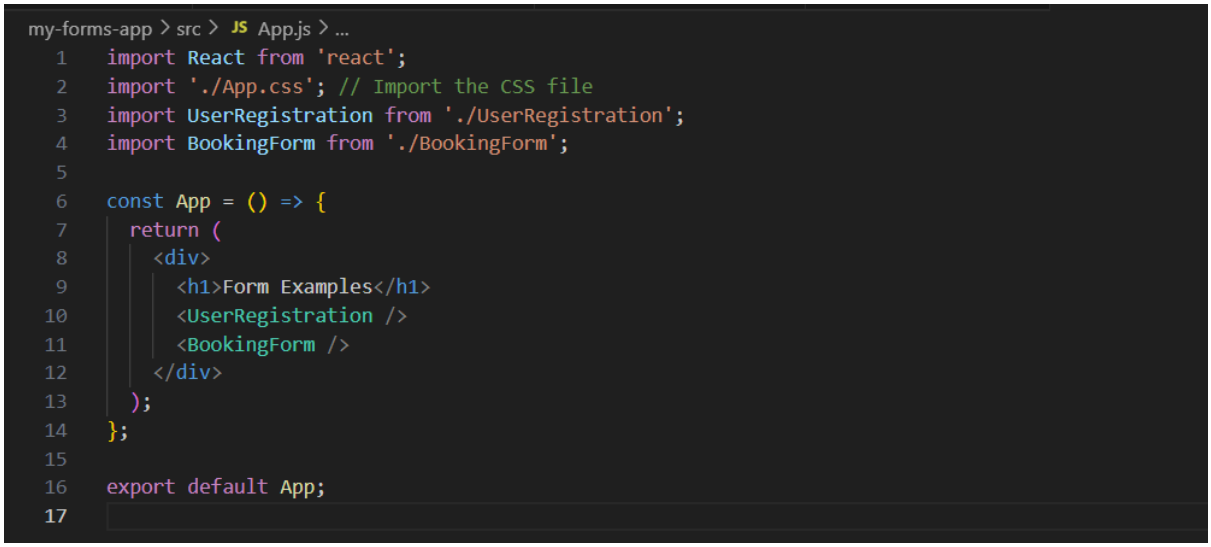
Full Name
Email
Phone Number
Appointment Date
Appointment Time

1.Create a form with the necessary input fields.
2.Add basic validation for required fields and email format.
3.Display a confirmation message upon successful submission.
4.Add date and time pickers for selecting the appointment date and time.

Solution –

App.js

```
import React from 'react';
import './App.css'; // Import the CSS file
import UserRegistration from './UserRegistration';
import BookingForm from './BookingForm';

const App = () => {
  return (
    <div>
      <h1>Form Examples</h1>
      <UserRegistration />
      <BookingForm />
    </div>
  );
};

export default App;
```



UserRegistration.js

```
import React, { useState } from 'react';

const UserRegistration = () => {
  const [formData, setFormData] = useState({
    firstName: '',
    lastName: '',
    email: '',
    password: '',
    confirmPassword: ''
  });

  const [errors, setErrors] = useState({});
  const [successMessage, setSuccessMessage] = useState('');

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData((prevData) => ({
```

```
      ...prevData,
      [name]: value
    }));
  };

  const validate = () => {
    const newErrors = {};
    const { firstName, lastName, email, password, confirmPassword } =
formData;

    // Required fields validation
    if (!firstName) newErrors.firstName = 'First name is required';
    if (!lastName) newErrors.lastName = 'Last name is required';
    if (!email) newErrors.email = 'Email is required';
    if (!password) newErrors.password = 'Password is required';
    if (!confirmPassword) newErrors.confirmPassword = 'Confirm password is
required';

    // Email format validation
    if (email && !/\S+@\S+\.\S+/.test(email)) {
      newErrors.email = 'Email is invalid';
    }

    // Password and confirm password match validation
    if (password && confirmPassword && password !== confirmPassword) {
      newErrors.confirmPassword = 'Passwords do not match';
    }
```

```
  return newErrors;
};

const handleSubmit = (e) => {
  e.preventDefault();
  const validationErrors = validate();
  if (Object.keys(validationErrors).length === 0) {
    setSuccessMessage('Registration successful!');
    setErrors({});
  } else {
    setErrors(validationErrors);
    setSuccessMessage('');
  }
};

return (
  <div>
    <h2>User Registration</h2>
    <form onSubmit={handleSubmit}>
      <div>
        <label>First Name</label>
        <input
          type="text"
          name="firstName"
          value={formData.firstName}
          onChange={handleChange}
        />
```

```
  {errors.firstName && <span>{errors.firstName}</span>}
</div>

<div>
  <label>Last Name</label>
  <input
    type="text"
    name="lastName"
    value={formData.lastName}
    onChange={handleChange}
  />
  {errors.lastName && <span>{errors.lastName}</span>}
</div>

<div>
  <label>Email</label>
  <input
    type="email"
    name="email"
    value={formData.email}
    onChange={handleChange}
  />
  {errors.email && <span>{errors.email}</span>}
</div>

<div>
  <label>Password</label>
```

```
          <input
            type="password"
            name="password"
            value={formData.password}
            onChange={handleChange}
          />
          {errors.password && <span>{errors.password}</span>}
        </div>

        <div>
          <label>Confirm Password</label>
          <input
            type="password"
            name="confirmPassword"
            value={formData.confirmPassword}
            onChange={handleChange}
          />
          {errors.confirmPassword && <span>{errors.confirmPassword}</span>}
        </div>

        <button type="submit">Register</button>
        {successMessage && <p>{successMessage}</p>}
      </form>
    </div>
  );
};
```

export default UserRegistration;

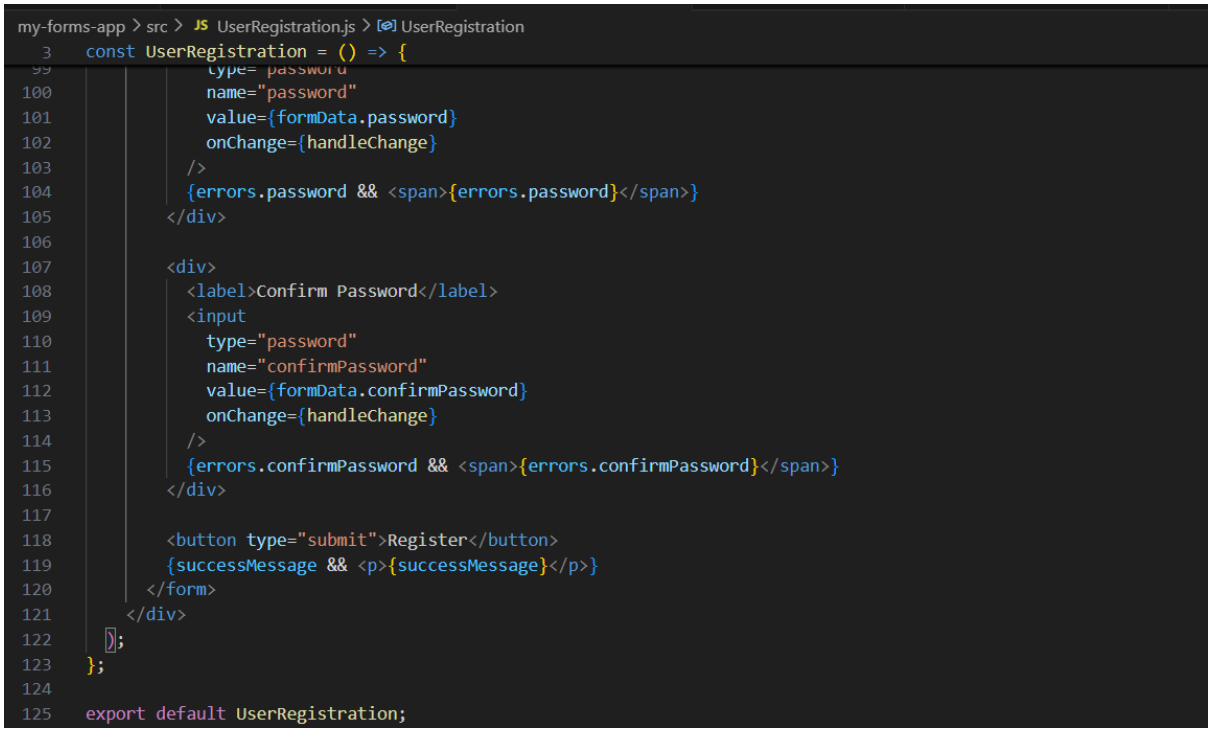
my-forms-app > src > UserRegistration.js > UserRegistration > handleChange > setFormData() callback > [name]

```
import React, { useState } from 'react';

const UserRegistration = () => {
  const [formData, setFormData] = useState({
    firstName: '',
    lastName: '',
    email: '',
    password: '',
    confirmPassword: ''
  });

  const [errors, setErrors] = useState({});
  const [successMessage, setSuccessMessage] = useState('');

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData((prevData) => ({
      ...prevData,
      [name]: value
    }));
  };

  const validate = () => {
    const newErrors = {};
    const { firstName, lastName, email, password, confirmPassword } = formData;

    // Required fields validation
    if (!firstName) newErrors.firstName = 'First name is required';
```

```
    // Required fields validation
    if (!firstName) newErrors.firstName = 'First name is required';
    if (!lastName) newErrors.lastName = 'Last name is required';
    if (!email) newErrors.email = 'Email is required';
    if (!password) newErrors.password = 'Password is required';
    if (!confirmPassword) newErrors.confirmPassword = 'Confirm password is required';

    // Email format validation
    if (email && !/\S+@\S+\.\S+/.test(email)) {
      newErrors.email = 'Email is invalid';
    }

    // Password and confirm password match validation
    if (password && confirmPassword && password !== confirmPassword) {
      newErrors.confirmPassword = 'Passwords do not match';
    }

    return newErrors;
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const validationErrors = validate();
    if (Object.keys(validationErrors).length === 0) {
      setSuccessMessage('Registration successful!');
```

```
      setSuccessMessage('Registration successful!');
      setErrors({});
    } else {
      setErrors(validationErrors);
      setSuccessMessage('');
    }
  };

  return (
    <div>
      <h2>User Registration</h2>
      <form onSubmit={handleSubmit}>
        <div>
          <label>First Name</label>
          <input
            type="text"
            name="firstName"
            value={formData.firstName}
            onChange={handleChange}
          />
          {errors.firstName && <span>{errors.firstName}</span>}
        </div>

        <div>
          <label>Last Name</label>
```

```
        <div>
          <label>Last Name</label>
          <input
            type="text"
            name="lastName"
            value={formData.lastName}
            onChange={handleChange}
          />
          {errors.lastName && <span>{errors.lastName}</span>}
        </div>

        <div>
          <label>Email</label>
          <input
            type="email"
            name="email"
            value={formData.email}
            onChange={handleChange}
          />
          {errors.email && <span>{errors.email}</span>}
        </div>

        <div>
          <label>Password</label>
          <input
            type="password"
```



BookingForm.js

```
import React, { useState } from 'react';


const BookingForm = () => {
  const [formData, setFormData] = useState({
    fullName: '',
    email: '',
    phoneNumber: '',
    appointmentDate: '',
    appointmentTime: ''
  });

  const [errors, setErrors] = useState({});
  const [confirmationMessage, setConfirmationMessage] = useState('');
```

```
  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData((prevData) => ({
      ...prevData,
      [name]: value
    }));
  };

  const validate = () => {
    const newErrors = {};
    const { fullName, email, phoneNumber, appointmentDate, appointmentTime
} = formData;

    // Required fields validation
    if (!fullName) newErrors.fullName = 'Full name is required';
    if (!email) newErrors.email = 'Email is required';
    if (!phoneNumber) newErrors.phoneNumber = 'Phone number is required';
    if (!appointmentDate) newErrors.appointmentDate = 'Appointment date is
required';
    if (!appointmentTime) newErrors.appointmentTime = 'Appointment time is
required';

    // Email format validation
    if (email && !/\S+@\S+\.\S+/.test(email)) {
      newErrors.email = 'Email is invalid';
    }

    return newErrors;
```

```
};

const handleSubmit = (e) => {
  e.preventDefault();
  const validationErrors = validate();
  if (Object.keys(validationErrors).length === 0) {
    setConfirmationMessage('Appointment booked successfully!');
    setErrors({});
  } else {
    setErrors(validationErrors);
    setConfirmationMessage('');
  }
};

return (
  <div>
    <h2>Booking Form</h2>
    <form onSubmit={handleSubmit}>
      <div>
        <label>Full Name</label>
        <input
          type="text"
          name="fullName"
          value={formData.fullName}
          onChange={handleChange}
        />
        {errors.fullName && <span>{errors.fullName}</span>}
```

```
</div>

<div>
  <label>Email</label>
  <input
    type="email"
    name="email"
    value={formData.email}
    onChange={handleChange}
  />
  {errors.email && <span>{errors.email}</span>}
</div>

<div>
  <label>Phone Number</label>
  <input
    type="text"
    name="phoneNumber"
    value={formData.phoneNumber}
    onChange={handleChange}
  />
  {errors.phoneNumber && <span>{errors.phoneNumber}</span>}
</div>

<div>
  <label>Appointment Date</label>
  <input
```

```
          type="date"
          name="appointmentDate"
          value={formData.appointmentDate}
          onChange={handleChange}
        />
        {errors.appointmentDate && <span>{errors.appointmentDate}</span>}
      </div>

      <div>
        <label>Appointment Time</label>
        <input
          type="time"
          name="appointmentTime"
          value={formData.appointmentTime}
          onChange={handleChange}
        />
        {errors.appointmentTime && <span>{errors.appointmentTime}</span>}
      </div>

      <button type="submit">Book Appointment</button>
      {confirmationMessage && <p>{confirmationMessage}</p>}
    </form>
  </div>
 );
};

export default BookingForm;
```

my-forms-app > src > JS BookingForm.js > BookingForm

```jsx
import React, { useState } from 'react';

const BookingForm = () => {
  const [formData, setFormData] = useState({
    fullName: '',
    email: '',
    phoneNumber: '',
    appointmentDate: '',
    appointmentTime: ''
  });

  const [errors, setErrors] = useState({});
  const [confirmationMessage, setConfirmationMessage] = useState('');

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData((prevData) => ({
      ...prevData,
      [name]: value
    }));
  };

  const validate = () => {
    const newErrors = {};
    const { fullName, email, phoneNumber, appointmentDate, appointmentTime } = formData;

    // Required fields validation
    if (!fullName) newErrors.fullName = 'Full name is required';
```

my-forms-app > src > JS BookingForm.js > ...

```jsx
const BookingForm = () => {
  const validate = () => {
    if (!email) newErrors.email = 'Email is required';
    if (!phoneNumber) newErrors.phoneNumber = 'Phone number is required';
    if (!appointmentDate) newErrors.appointmentDate = 'Appointment date is required';
    if (!appointmentTime) newErrors.appointmentTime = 'Appointment time is required';

    // Email format validation
    if (email && !/\S+@\S+\.\S+/.test(email)) {
      newErrors.email = 'Email is invalid';
    }

    return newErrors;
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const validationErrors = validate();
    if (Object.keys(validationErrors).length === 0) {
      setConfirmationMessage('Appointment booked successfully!');
      setErrors({});
    } else {
      setErrors(validationErrors);
      setConfirmationMessage('');
    }
  };
```

```
54    return (
55      <div>
56        <h2>Booking Form</h2>
57        <form onSubmit={handleSubmit}>
58          <div>
59            <label>Full Name</label>
60            <input
61              type="text"
62              name="fullName"
63              value={formData.fullName}
64              onChange={handleChange}
65            />
66            {errors.fullName && <span>{errors.fullName}</span>}
67          </div>
68
69          <div>
70            <label>Email</label>
71            <input
72              type="email"
73              name="email"
74              value={formData.email}
75              onChange={handleChange}
76            />
77            {errors.email && <span>{errors.email}</span>}
78          </div>
79
```

my-forms-app > src > JS BookingForm.js > ...

```
3   const BookingForm = () => {
80          <div>
81            <label>Phone Number</label>
82            <input
83              type="text"
84              name="phoneNumber"
85              value={formData.phoneNumber}
86              onChange={handleChange}
87            />
88            {errors.phoneNumber && <span>{errors.phoneNumber}</span>}
89          </div>
90
91          <div>
92            <label>Appointment Date</label>
93            <input
94              type="date"
95              name="appointmentDate"
96              value={formData.appointmentDate}
97              onChange={handleChange}
98            />
99            {errors.appointmentDate && <span>{errors.appointmentDate}</span>}
100         </div>
101
102         <div>
103           <label>Appointment Time</label>
104           <input
105             type="time"
```

```
            type="time"
            name="appointmentTime"
            value={formData.appointmentTime}
            onChange={handleChange}
          />
          {errors.appointmentTime && <span>{errors.appointmentTime}</span>}
        </div>

        <button type="submit">Book Appointment</button>
        {confirmationMessage && <p>{confirmationMessage}</p>}
      </form>
    </div>
  );
};

export default BookingForm;
```

OUTPUT-