

Project Development Phase

Sprint – 1

Source Code

Date	01 November 2022
Team ID	PNT2022TMID44305
Project Name	Personal Expense Tracker
Maximum Marks	8 Marks

App.js

```
import React from 'react';
import { BrowserRouter, Routes, Route, } from "react-router-dom";
import Home from './routes/home/Home';
import Login from './routes/forms/login/Login';
import Register from './routes/forms/register/Register';
const App = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route exact path="/login" element={<Login />} />
        <Route exact path="/register" element={<Register />} />
      </Routes>
    </BrowserRouter>
  );
};
export default App;
```

App.css

```
.App {
  text-align: center;
}
```

```
.App-logo {  
  height: 40vmin;  
  pointer-events: none;  
}  
  
@media (prefers-reduced-motion: no-preference) {  
  .App-logo {  
    animation: App-logo-spin infinite 20s linear;  
  }  
}  
  
.App-header {  
  background-color: #282c34;  
  min-height: 100vh;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  font-size: calc(10px + 2vmin);  
  color: white;  
}  
  
.App-link {  
  color: #61dafb;  
}  
  
@keyframes App-logo-spin {  
  from {  
    transform: rotate(0deg);  
  }  
  to {  
    transform: rotate(360deg);  
  }  
}
```

```
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import './index.css';
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

index.css

```
#root, body, html {
  height: 100%;
  margin: 0;
}
body {
  background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url(./assets/background-
img.jpeg);
  background-size: cover;
}
```

.routes/forms/formInput/FormInput.js

```
import React, { useState } from 'react';
import './formInput.css';
const FormInput = (props) => {
  const [focused, setFocused] = useState(false);
```

```

const { label, errorMessage, onChange, id, ...inputProps } = props;
const handleFocus = (e) => {
  setFocused(true);
};
return (
  <div className="formInput">
    <label className="labelField">{label}</label>
    <input
      {...inputProps}
      className="inputField"
      onChange={onChange}
      onBlur={handleFocus}
      onFocus={() => inputProps.name === "confirmPassword" && setFocused(true)}
      focused={focused.toString()} />
    <span className="errorMessage">{errorMessage}</span>
  </div>
);
};
export default FormInput;

```

.routes/forms/formInput/formInput.css

```

.inputField {
  padding: 15px;
  margin: 10px 0px;
  border-radius: 10px;
  border: 1px solid rgb(122 103 0);
}
.formInput {
  display: flex;
  flex-direction: column;

```

```

    width: 280px;
  }
  .labelField {
    font-size: 12px;
    color: rgb(50, 50, 0);
  }
  .errorMessage {
    font-size: 10px;
    padding: 3px;
    color: red;
    display: none;
  }
  .inputField:invalid:focus {
    border: 1px solid red;
  }
  .inputField:invalid:focus ~ span {
    display: block;
  }

```

.routes/forms/login/Login.js

```

import React, { useState } from 'react';
import { Link, useNavigate, } from 'react-router-dom';
import swal from 'sweetalert';
import FormInput from '../formInput/FormInput';
import './login.css';

```

```

const Login = () => {
  const navigate = useNavigate();
  const [values, setValues] = useState({
    username: "",

```

```

        password: "",
    });
    const { username, password } = values;
    const inputs = [
        {
            id: 1,
            name: "username",
            type: "text",
            placeholder: "Username",
            label: "Username",
            errorMessage: "Username should be 3-16 characters and shouldn't include any special
characters",
            required: true,
            pattern: "^[A-Za-z0-9]{3,16}$",
        },
        {
            id: 3,
            name: "password",
            type: "password",
            placeholder: "Password",
            label: "Password",
            errorMessage: "Password should be 8-20 characters and include at least 1 letter, 1
number and 1 special character",
            required: true,
            pattern: "^(?=.*[0-9])(?=.*[A-Za-z])(?=.*[!@#$%^&*])[a-zA-Z0-
9!@#$%^&*]{8,20}$",
        },
    ]
    const handleSubmit = (e) => {
        e.preventDefault();
        if (username === "aaaa" && password === "aaaa@1111") {

```

```

        swal("Successfull Login", "Welcome to Dashboard", "success");
        navigate(`/dashboard/${username}`);
    }
    else {
        swal("Wrong Credentials", "Please try again!", "error");
    }
};

const onChange = (e) => {
    setValues({ ...values, [e.target.name]: e.target.value });
};

return (
    <div className="login">
        <form className="loginForm" onSubmit={handleSubmit}>
            <h1 className="loginTitle">Login</h1>
            {inputs.map((input) => (
                <FormInput
                    key={input.id}
                    {...input}
                    value={values[input.name]} onChange={onChange}
                />
            ))}

            <button className="loginBtn" >Login</button>

            <div>
                <p style={{ fontSize: "14px", marginBottom: "30px" }}>New user? <Link
to="/register" className="registerLink">Click to Register</Link></p>
            </div>
        </form>
    </div>
);
};

```

```
export default Login;
```

.routes/forms/login/login.css

```
.login {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  height: 100vh;  
}  
.loginForm {  
  background-color: rgb(244 228 140);  
  padding: 0px 60px;  
  border-radius: 10px;  
}  
.loginTitle {  
  color: rgb(139 120 2);  
  text-align: center;  
  margin-top: 30px;  
}  
.loginBtn {  
  width: 100%;  
  height: 50px;  
  padding: 10px;  
  border: none;  
  background-color: rgb(139 120 2);  
  color: white;  
  border-radius: 10px;  
  font-weight: 600;  
  font-size: 18px;  
  cursor: pointer;
```



```
margin-top: 15px;
margin-bottom: 15px;
}
.loginBtn:hover {
  background-color: rgb(246 235 172);
  color: rgb(139 120 2);
  border: 2px solid rgb(139 120 2);
}
.registerLink {
  text-decoration: none;
  margin-left: 5px;
  color: rgb(72 62 1);
  font-weight: 500;
}
```

.routes/forms/register/Register.js

```
import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import FormInput from '../formInput/FormInput';
import './register.css';
```

```
const Register = () => {

  const navigate = useNavigate();

  const [values, setValues] = useState({
    username: "",
    email: "",
    password: "",
    confirmPassword: "",
```

```
});
```

```
const { username } = values;
```

```
const inputs = [  
  {  
    id: 1,  
    name: "username",  
    type: "text",  
    placeholder: "Username",  
    label: "Username",  
    errorMessage: "Username should be 3-16 characters and shouldn't include any special  
characters",  
    required: true,  
    pattern: "^[A-Za-z0-9]{3,16}$",  
  },  
  {  
    id: 2,  
    name: "email",  
    type: "email",  
    placeholder: "Email",  
    label: "Email",  
    errorMessage: "It should be a valid email address",  
    required: true,  
  },  
  
  {  
    id: 3,  
    name: "password",  
    type: "password",  
    placeholder: "Password",
```

```

      label: "Password",

      errorMessage: "Password should be 8-20 characters and include at least 1 letter, 1
number and 1 special character",

      required: true,

      pattern: "^(?=.*[0-9])(?=.*[A-Za-z])(?=.*[@#$%^&*])[a-zA-Z0-
9!@#$%^&*]{8,20}$",
    },
    {
      id: 4,
      name: "confirmPassword",
      type: "password",
      placeholder: "Confirm Password",
      label: "Confirm Password",
      errorMessage: "Passwords don't match",
      required: true,
      pattern: values.password,
    },
  ],
]

```

```

const handleSubmit = (e) => {
  e.preventDefault();
  navigate(`/dashboard/${username}`);
};

```

```

const onChange = (e) => {
  setValues({ ...values, [e.target.name]: e.target.value });
};

```

```

return (
  <div className="register">

```

```

    <form className="registerForm" onSubmit={handleSubmit}>
      <h1 className="registerTitle">Register</h1>
      {inputs.map((input) => (
        <FormInput
          key={input.id}
          {...input}
          value={values[input.name]} onChange={onChange}
        />
      ))}

      <button className="registerBtn">Submit</button>

      <div>
        <p style={{ fontSize: "14px", marginBottom: "30px" }}>Already have an
account? <Link to="/login" className="loginLink">Click to Login</Link></p>
      </div>
    </form>
  </div>
);
};

```

```
export default Register;
```

.routes/forms/register/register.css

```

.register {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
}

```

```
.registerForm {
```

```
background-color: rgb(244 228 140);  
padding: 0px 60px;  
border-radius: 10px;  
}
```

```
.registerTitle {  
    color: rgb(139 120 2);  
    text-align: center;  
    margin-top: 30px;  
}
```

```
.registerBtn {  
    width: 100%;  
    height: 50px;  
    padding: 10px;  
    border: none;  
    background-color: rgb(139 120 2);  
    color: white;  
    border-radius: 10px;  
    font-weight: bold;  
    font-size: 18px;  
    cursor: pointer;  
    margin-top: 15px;  
    margin-bottom: 15px;  
}
```

```
.registerBtn:hover {  
    background-color: rgb(246 235 172);  
    color: rgb(139 120 2);  
}
```

```
border: 2px solid rgb(139 120 2);  
}
```

```
.loginLink {  
  text-decoration: none;  
  margin-left: 5px;  
  color: rgb(72 62 1);  
  font-weight: 500;  
}
```