Concepts in Focus

- User defined functions should be in the arrow function syntax
- Using Redirect Component in Callbacks or Event Listeners
- Using history.replace() in render() instead of Redirect Component
- Missing withRouter() to have history prop
- When request object keys are incorrect
- Not Updating the State when required

1. User defined functions should be in arrow function syntax

Mistake

In the below code snippet, we have defined the

`onChangeUsername` function as a function definition instead of an arrow function.

File: src/components/LoginForm/index.js

```jsx
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

class LoginForm extends Component {
  state = {
```

```
    username: '',
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }

  onChangeUsername(event){
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
  }

  onSubmitSuccess = jwtToken => {
    const {history} = this.props


    Cookies.set('jwt_token', jwtToken, {
      expires: 30,
    })

    history.replace('/')
  }

  onSubmitFailure = errorMsg => {
    this.setState({showSubmitError: true, errorMsg})
  }

  submitForm = async event => {
    event.preventDefault()
    const {username, password} = this.state
    const userDetails = {username, password}
    const url = 'https://apis.ccbp.in/login'
    const options = {
      method: 'POST',
```

```
      body: JSON.stringify(userDetails),
    }
    const response = await fetch(url, options)
    const data = await response.json()
    console.log(data.error_msg)
    if (response.ok === true) {
      this.onSubmitSuccess(data.jwt_token)
    } else {
      this.onSubmitFailure(data.error_msg)
    }
  }

  renderPasswordField = () => {
    const {password} = this.state
    return (
      <>
        <label className="input-label" htmlFor="password">
          PASSWORD
        </label>
        <input
          type="password"
          id="password"
          className="password-input-filed"
          value={password}
          onChange={this.onChangePassword}
          placeholder="Password"
        />
      </>
    )
  }

  renderUsernameField = () => {
    const {username} = this.state
    return (
      <>
```

```
        <label className="input-label" htmlFor="username">
          USERNAME
        </label>
        <input
          type="text"
          id="username"
          className="username-input-filed"
          value={username}
          onChange={this.onChangeUsername}
          placeholder="Username"
        />
      </>
    )
  }

  render() {
    const {showSubmitError, errorMsg} = this.state
    const jwtToken = Cookies.get('jwt_token')
    if (jwtToken !== undefined) {
      return <Redirect to="/" />
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
```

```
            className="login-website-logo-desktop-image"
            alt="website logo"
          />
          <div className="input-container">{this.renderUsernameField()}</div>
          <div className="input-container">{this.renderPasswordField()}</div>
          <button type="submit" className="login-button">
            Login
          </button>
          {showSubmitError && <p className="error-message">{errorMsg}</p>}
        </form>
      </div>
    )
  }
}

export default LoginForm
```

Collapse

Explanation:

- In Class Component, the function definition will execute in the browser context, so the value of `this` will be null.
- Component Life Cycle Methods(componentDidMount,render,constructor,componentWillUnmount) are extended from `React Component` . As they run in `React Context` , there is no need to write them in arrow functions.

Solution:

File: src/components/LoginForm/index.js

```jsx
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

```

JSX

```
class LoginForm extends Component {
  state = {
    username: '',
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }

  onChangeUsername = event => {
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
  }

  onSubmitSuccess = jwtToken => {
    const {history} = this.props


    Cookies.set('jwt_token', jwtToken, {
      expires: 30,
    })

    history.replace('/')
  }

  onSubmitFailure = errorMsg => {
    this.setState({showSubmitError: true, errorMsg})
  }

  submitForm = async event => {
    event.preventDefault()
    const {username, password} = this.state
    const userDetails = {username, password}
    const url = 'https://apis.ccbp.in/login'
```

```
    const options = {
      method: 'POST',
      body: JSON.stringify(userDetails),
    }
    const response = await fetch(url, options)
    const data = await response.json()
    console.log(data.error_msg)
    if (response.ok === true) {
      this.onSubmitSuccess(data.jwt_token)
    } else {
      this.onSubmitFailure(data.error_msg)
    }
  }

  renderPasswordField = () => {
    const {password} = this.state
    return (
      <>
        <label className="input-label" htmlFor="password">
          PASSWORD
        </label>
        <input
          type="password"
          id="password"
          className="password-input-filed"
          value={password}
          onChange={this.onChangePassword}
          placeholder="Password"
        />
      </>
    )
  }

  renderUsernameField = () => {
    const {username} = this.state
    return (
```

```
      <>
        <label className="input-label" htmlFor="username">
          USERNAME
        </label>
        <input
          type="text"
          id="username"
          className="username-input-filed"
          value={username}
          onChange={this.onChangeUsername}
          placeholder="Username"
        />
      </>
    )
  }

  render() {
    const {showSubmitError, errorMsg} = this.state
    const jwtToken = Cookies.get('jwt_token')
    if (jwtToken !== undefined) {
      return <Redirect to="/" />
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
```

```
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
              className="login-website-logo-desktop-image"
              alt="website logo"
            />
            <div className="input-container">{this.renderUsernameField()}</div>
            <div className="input-container">{this.renderPasswordField()}</div>
            <button type="submit" className="login-button">
              Login
            </button>
            {showSubmitError && <p className="error-message">{errorMsg}</p>}
          </form>
        </div>
      )
    }
}

export default LoginForm
```

Collapse

2. Using Redirect Component in Callbacks or Event Listeners

Mistake:

As the

`Redirect` Component returns JSX, it should not be used in event handlers.

File: src/components/Header/index.js

```
import {Link, withRouter} from 'react-router-dom'
```

JSX

```jsx

import Cookies from 'js-cookie'

import './index.css'

const Header = props => {
  const onClickLogout = () => {
    Cookies.remove('jwt_token')
    return <Redirect to="/login" />
  }

  return (
    <nav className="nav-header">
      <div className="nav-content">
        <div className="nav-bar-mobile-logo-container">
          <img
            className="website-logo"
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            alt="website logo"
          />

          <button type="button" className="nav-mobile-btn">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-log-out-img.png"
              alt="nav logout"
              className="nav-bar-image"
              onClick={onClickLogout}
            />
          </button>
        </div>

        <div className="nav-bar-large-container">
          <img
            className="website-logo"
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            alt="website logo"
```

```
        />
        <ul className="nav-menu">
          <li className="nav-menu-item">
            <Link to="/" className="nav-link">
              Home
            </Link>
          </li>

          <li className="nav-menu-item">
            <Link to="/products" className="nav-link">
              Products
            </Link>
          </li>

          <li className="nav-menu-item">
            <Link to="/cart" className="nav-link">
              Cart
            </Link>
          </li>
        </ul>
        <button
          type="button"
          className="logout-desktop-btn"
          onClick={onClickLogout}
        >
          Logout
        </button>
      </div>
    </div>
    <div className="nav-menu-mobile">
      <ul className="nav-menu-list-mobile">
        <li className="nav-menu-item-mobile">
          <Link to="/" className="nav-link">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-home-icon.png"
              alt="nav home"
```

```
            className="nav-bar-image"
          />
        </Link>
      </li>

      <li className="nav-menu-item-mobile">
        <Link to="/products" className="nav-link">
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-products-icon.png"
            alt="nav products"
            className="nav-bar-image"
          />
        </Link>
      </li>
      <li className="nav-menu-item-mobile">
        <Link to="/cart" className="nav-link">
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-cart-icon.png"
            alt="nav cart"
            className="nav-bar-image"
          />
        </Link>
      </li>
    </ul>
  </div>
</nav>
  )
}

export default withRouter(Header)
```

Collapse

Explanation:

- `Redirect` Component returns JSX and it is used to render the UI in the specific Route.
- To navigate to the specific Routes in event handlers or callbacks used the `history` object.

Solution:

File: src/components/Header/index.js

JSX

```jsx
import {Link, withRouter} from 'react-router-dom'

import Cookies from 'js-cookie'

import './index.css'

const Header = props => {
  const onClickLogout = () => {
    const {history} = props
    Cookies.remove('jwt_token')
    history.replace('/login')
  }

  return (
    <nav className="nav-header">
      <div className="nav-content">
        <div className="nav-bar-mobile-logo-container">
          <img
            className="website-logo"
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            alt="website logo"
          />

          <button type="button" className="nav-mobile-btn">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-log-out-img.png"
              alt="nav logout"
              className="nav-bar-image"
```

```
          className="nav-bar-image"
          onClick={onClickLogout}
        />
      </button>
    </div>

    <div className="nav-bar-large-container">
      <img
        className="website-logo"
        src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
        alt="website logo"
      />
      <ul className="nav-menu">
        <li className="nav-menu-item">
          <Link to="/" className="nav-link">
            Home
          </Link>
        </li>

        <li className="nav-menu-item">
          <Link to="/products" className="nav-link">
            Products
          </Link>
        </li>

        <li className="nav-menu-item">
          <Link to="/cart" className="nav-link">
            Cart
          </Link>
        </li>
      </ul>
      <button
        type="button"
        className="logout-desktop-btn"
        onClick={onClickLogout}
      >
```

```
          Logout
        </button>
      </div>
    </div>
    <div className="nav-menu-mobile">
      <ul className="nav-menu-list-mobile">
        <li className="nav-menu-item-mobile">
          <Link to="/" className="nav-link">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-home-icon.png"
              alt="nav home"
              className="nav-bar-image"
            />
          </Link>
        </li>

        <li className="nav-menu-item-mobile">
          <Link to="/products" className="nav-link">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-products-icon.png"
              alt="nav products"
              className="nav-bar-image"
            />
          </Link>
        </li>
        <li className="nav-menu-item-mobile">
          <Link to="/cart" className="nav-link">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-cart-icon.png"
              alt="nav cart"
              className="nav-bar-image"
            />
          </Link>
        </li>
      </ul>
    </div>
```

```
      </nav>
    )
  }

export default withRouter(Header)
```

Collapse

3. Using history.replace() in render() instead of Redirect Component

Mistake:

In the

`render` method, we have used `history.replace()` instead of `Redirect` Component and history prop is used in event handlers or callbacks to navigate to specific routes.

File: src/components/LoginForm/index.js

JSX

```
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

class LoginForm extends Component {
  state = {
    username: '',
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }
```

```
  onChangeUsername = event => {
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
  }

  onSubmitSuccess = jwtToken => {
    const {history} = this.props


    Cookies.set('jwt_token', jwtToken, {
      expires: 30,
    })

    history.replace('/')
  }

  onSubmitFailure = errorMsg => {
    this.setState({showSubmitError: true, errorMsg})
  }

  submitForm = async event => {
    event.preventDefault()
    const {username, password} = this.state
    const userDetails = {username, password}
    const url = 'https://apis.ccbp.in/login'
    const options = {
      method: 'POST',
      body: JSON.stringify(userDetails),
    }
    const response = await fetch(url, options)
    const data = await response.json()
    console.log(data.error_msg)
```

```
    if (response.ok === true) {
      this.onSubmitSuccess(data.jwt_token)
    } else {
      this.onSubmitFailure(data.error_msg)
    }
  }

  renderPasswordField = () => {
    const {password} = this.state
    return (
      <>
        <label className="input-label" htmlFor="password">
          PASSWORD
        </label>
        <input
          type="password"
          id="password"
          className="password-input-filed"
          value={password}
          onChange={this.onChangePassword}
          placeholder="Password"
        />
      </>
    )
  }

  renderUsernameField = () => {
    const {username} = this.state
    return (
      <>
        <label className="input-label" htmlFor="username">
          USERNAME
        </label>
        <input
          type="text"
          id="username"
```

```
          className="username-input-filed"
          value={username}
          onChange={this.onChangeUsername}
          placeholder="Username"
        />
      </>
    )
  }

  render() {
    const {showSubmitError, errorMsg} = this.state
    const jwtToken = Cookies.get('jwt_token')
    const {history} = this.props
    if (jwtToken !== undefined) {
      history.replace('/')
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            className="login-website-logo-desktop-image"
            alt="website logo"
          />
          <div className="input-container">{this.renderUsernameField()}</div>
          <div className="input-container">{this.renderPasswordField()}</div>
```

```
          <button type="submit" className="login-button">
            Login
          </button>
          {showSubmitError && <p className="error-message">{errorMsg}</p>}
        </form>
      </div>
    )
  }
}

export default LoginForm
```

Collapse

Explanation:

In

`render()` JSX is displayed and `Redirect` Component returns JSX. So it's a best practice to use `Redirect Component` while navigating in render().

Solution:

File: src/components/LoginForm/index.js

```jsx
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

class LoginForm extends Component {
  state = {
    username: '',
```

```
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }

  onChangeUsername = event => {
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
  }

  onSubmitSuccess = jwtToken => {
    const {history} = this.props


    Cookies.set('jwt_token', jwtToken, {
      expires: 30,
    })

    history.replace('/')
  }

  onSubmitFailure = errorMsg => {
    this.setState({showSubmitError: true, errorMsg})
  }

  submitForm = async event => {
    event.preventDefault()
    const {username, password} = this.state
    const userDetails = {username, password}
    const url = 'https://apis.ccbp.in/login'
    const options = {
      method: 'POST',
      body: JSON.stringify(userDetails),
```

```
    }
    const response = await fetch(url, options)
    const data = await response.json()
    console.log(data.error_msg)
    if (response.ok === true) {
      this.onSubmitSuccess(data.jwt_token)
    } else {
      this.onSubmitFailure(data.error_msg)
    }
  }

  renderPasswordField = () => {
    const {password} = this.state
    return (
      <>
        <label className="input-label" htmlFor="password">
          PASSWORD
        </label>
        <input
          type="password"
          id="password"
          className="password-input-filed"
          value={password}
          onChange={this.onChangePassword}
          placeholder="Password"
        />
      </>
    )
  }

  renderUsernameField = () => {
    const {username} = this.state
    return (
      <>
        <label className="input-label" htmlFor="username">
          USERNAME
```

```
        </label>
        <input
          type="text"
          id="username"
          className="username-input-filed"
          value={username}
          onChange={this.onChangeUsername}
          placeholder="Username"
        />
      </>
    )
  }

  render() {
    const {showSubmitError, errorMsg} = this.state
    const jwtToken = Cookies.get('jwt_token')
    if (jwtToken !== undefined) {
     return <Redirect to="/" />
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            className="login-website-logo-desktop-image"
            alt="website logo"
```

```
          alt="website logo"
          />
          <div className="input-container">{this.renderUsernameField()}</div>
          <div className="input-container">{this.renderPasswordField()}</div>
          <button type="submit" className="login-button">
            Login
          </button>
          {showSubmitError && <p className="error-message">{errorMsg}</p>}
        </form>
      </div>
    )
  }
}

export default LoginForm
```

Collapse

4. Missing withRouter() to have history prop

Mistake:

The

`history` prop will be available for only Route Components. To use the `history` prop in other components, it should be wrapped with `withRouter()` .

File: src/components/Header/index.js

```jsx
import {Link} from 'react-router-dom'

import Cookies from 'js-cookie'
```

JSX

```
import './index.css'

const Header = props => {
  const onClickLogout = () => {
    const {history} = props
    Cookies.remove('jwt_token')
    history.replace('/login')
  }

  return (
    <nav className="nav-header">
      <div className="nav-content">
        <div className="nav-bar-mobile-logo-container">
          <img
            className="website-logo"
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            alt="website logo"
          />

          <button type="button" className="nav-mobile-btn">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-log-out-img.png"
              alt="nav logout"
              className="nav-bar-image"
              onClick={onClickLogout}
            />
          </button>
        </div>

        <div className="nav-bar-large-container">
          <img
            className="website-logo"
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            alt="website logo"
          />
```

```
          <ul className="nav-menu">
            <li className="nav-menu-item">
              <Link to="/" className="nav-link">
                Home
              </Link>
            </li>

            <li className="nav-menu-item">
              <Link to="/products" className="nav-link">
                Products
              </Link>
            </li>

            <li className="nav-menu-item">
              <Link to="/cart" className="nav-link">
                Cart
              </Link>
            </li>
          </ul>
          <button
            type="button"
            className="logout-desktop-btn"
            onClick={onClickLogout}
          >
            Logout
          </button>
        </div>
      </div>
      <div className="nav-menu-mobile">
        <ul className="nav-menu-list-mobile">
          <li className="nav-menu-item-mobile">
            <Link to="/" className="nav-link">
              <img
                src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-home-icon.png"
                alt="nav home"
                className="nav-bar-image"
```

```
            />
          </Link>
        </li>

        <li className="nav-menu-item-mobile">
          <Link to="/products" className="nav-link">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-products-icon.png"
              alt="nav products"
              className="nav-bar-image"
            />
          </Link>
        </li>
        <li className="nav-menu-item-mobile">
          <Link to="/cart" className="nav-link">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-cart-icon.png"
              alt="nav cart"
              className="nav-bar-image"
            />
          </Link>
        </li>
      </ul>
    </div>
  </nav>
 )
}

export default Header
```

Collapse

Solution:

File: src/components/Header/index.js

JSX

```jsx
import {Link, withRouter} from 'react-router-dom'

import Cookies from 'js-cookie'

import './index.css'

const Header = props => {
  const onClickLogout = () => {
    const {history} = props
    Cookies.remove('jwt_token')
    history.replace('/login')
  }

  return (
    <nav className="nav-header">
      <div className="nav-content">
        <div className="nav-bar-mobile-logo-container">
          <img
            className="website-logo"
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            alt="website logo"
          />

          <button type="button" className="nav-mobile-btn">
            <img
              src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-log-out-img.png"
              alt="nav logout"
              className="nav-bar-image"
              onClick={onClickLogout}
            />
          </button>
        </div>

        <div className="nav-bar-large-container">
```

```jsx
        <div className="nav-bar-large-container">
          <img
            className="website-logo"
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            alt="website logo"
          />
          <ul className="nav-menu">
            <li className="nav-menu-item">
              <Link to="/" className="nav-link">
                Home
              </Link>
            </li>

            <li className="nav-menu-item">
              <Link to="/products" className="nav-link">
                Products
              </Link>
            </li>

            <li className="nav-menu-item">
              <Link to="/cart" className="nav-link">
                Cart
              </Link>
            </li>
          </ul>
          <button
            type="button"
            className="logout-desktop-btn"
            onClick={onClickLogout}
          >
            Logout
          </button>
        </div>
      </div>
      <div className="nav-menu-mobile">
        <ul className="nav-menu-list-mobile">
```

```jsx
          <li className="nav-menu-item-mobile">
            <Link to="/" className="nav-link">
              <img
                src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-home-icon.png"
                alt="nav home"
                className="nav-bar-image"
              />
            </Link>
          </li>

          <li className="nav-menu-item-mobile">
            <Link to="/products" className="nav-link">
              <img
                src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-products-icon.png"
                alt="nav products"
                className="nav-bar-image"
              />
            </Link>
          </li>
          <li className="nav-menu-item-mobile">
            <Link to="/cart" className="nav-link">
              <img
                src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-cart-icon.png"
                alt="nav cart"
                className="nav-bar-image"
              />
            </Link>
          </li>
        </ul>
      </div>
    </nav>
  )
}

export default withRouter(Header)
```

Collapse

5. When request object keys are incorrect

Mistake:

For example below in the request object, we need to send

`username` and `password` but `userName` is sent instead of the `username` .

File: src/components/LoginForm/index.js

JSX

```jsx
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

class LoginForm extends Component {
  state = {
    userName: '',
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }

  onChangeUsername = event => {
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
  }
```

```
22
23  onSubmitSuccess = jwtToken => {
24    const {history} = this.props
25
26
27    Cookies.set('jwt_token', jwtToken, {
28      expires: 30,
29    })
30
31    history.replace('/')
32  }
33
34  onSubmitFailure = errorMsg => {
35    this.setState({showSubmitError: true, errorMsg})
36  }
37
38  submitForm = async event => {
39    event.preventDefault()
40    const {userName, password} = this.state
41    const userDetails = {userName, password}
42    const url = 'https://apis.ccbp.in/login'
43    const options = {
44      method: 'POST',
45      body: JSON.stringify(userDetails),
46    }
47    const response = await fetch(url, options)
48    const data = await response.json()
49    console.log(data.error_msg)
50    if (response.ok === true) {
51      this.onSubmitSuccess(data.jwt_token)
52    } else {
53      this.onSubmitFailure(data.error_msg)
54    }
55  }
56
57  renderPasswordField = () => {
```

```
    const {password} = this.state
    return (
      <>
        <label className="input-label" htmlFor="password">
          PASSWORD
        </label>
        <input
          type="password"
          id="password"
          className="password-input-filed"
          value={password}
          onChange={this.onChangePassword}
          placeholder="Password"
        />
      </>
    )
  }

  renderUsernameField = () => {
    const {username} = this.state
    return (
      <>
        <label className="input-label" htmlFor="username">
          USERNAME
        </label>
        <input
          type="text"
          id="username"
          className="username-input-filed"
          value={username}
          onChange={this.onChangeUsername}
          placeholder="Username"
        />
      </>
    )
```

```
  }

  render() {
    const {showSubmitError, errorMsg} = this.state
    const jwtToken = Cookies.get('jwt_token')
    if (jwtToken !== undefined) {
      return <Redirect to="/" />
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            className="login-website-logo-desktop-image"
            alt="website logo"
          />
          <div className="input-container">{this.renderUsernameField()}</div>
          <div className="input-container">{this.renderPasswordField()}</div>
          <button type="submit" className="login-button">
            Login
          </button>
          {showSubmitError && <p className="error-message">{errorMsg}</p>}
        </form>
      </div>
    )
  }
```

```
}

export default LoginForm
```

Collapse

Solution:

File: src/components/LoginForm/index.js

JSX

```jsx
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

class LoginForm extends Component {
  state = {
    username: '',
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }

  onChangeUsername = event => {
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
  }

  onSubmitSuccess = jwtToken => {
    const {history} = this.props
```

```
    Cookies.set('jwt_token', jwtToken, {
      expires: 30,
    })

    history.replace('/')
  }

  onSubmitFailure = errorMsg => {
    this.setState({showSubmitError: true, errorMsg})
  }

  submitForm = async event => {
    event.preventDefault()
    const {username, password} = this.state
    const userDetails = {username, password}
    const url = 'https://apis.ccbp.in/login'
    const options = {
      method: 'POST',
      body: JSON.stringify(userDetails),
    }
    const response = await fetch(url, options)
    const data = await response.json()
    console.log(data.error_msg)
    if (response.ok === true) {
      this.onSubmitSuccess(data.jwt_token)
    } else {
      this.onSubmitFailure(data.error_msg)
    }
  }

  renderPasswordField = () => {
    const {password} = this.state
    return (
      <>
```

```
      <label className="input-label" htmlFor="password">
        PASSWORD
      </label>
      <input
        type="password"
        id="password"
        className="password-input-filed"
        value={password}
        onChange={this.onChangePassword}
        placeholder="Password"
      />
    </>
  )
}

renderUsernameField = () => {
  const {username} = this.state
  return (
    <>
      <label className="input-label" htmlFor="username">
        USERNAME
      </label>
      <input
        type="text"
        id="username"
        className="username-input-filed"
        value={username}
        onChange={this.onChangeUsername}
        placeholder="Username"
      />
    </>
  )
}

render() {
  const {showSubmitError, errorMsg} = this.state
```

```
    const jwtToken = Cookies.get('jwt_token')
    if (jwtToken !== undefined) {
      return <Redirect to="/" />
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            className="login-website-logo-desktop-image"
            alt="website logo"
          />
          <div className="input-container">{this.renderUsernameField()}</div>
          <div className="input-container">{this.renderPasswordField()}</div>
          <button type="submit" className="login-button">
            Login
          </button>
          {showSubmitError && <p className="error-message">{errorMsg}</p>}
        </form>
      </div>
    )
  }
}

export default LoginForm
```

Collapse

6. Not Updating the State when required

Mistake:

On login failure, the

`showSubmitError` state should be updated to display the error message

File: src/components/LoginForm/index.js

JSX

```jsx
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

class LoginForm extends Component {
  state = {
    userName: '',
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }

  onChangeUsername = event => {
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
```

```
  }

  onSubmitSuccess = jwtToken => {
    const {history} = this.props


    Cookies.set('jwt_token', jwtToken, {
      expires: 30,
    })

    history.replace('/')
  }

  onSubmitFailure = errorMsg => {
    this.setState({errorMsg})
  }

  submitForm = async event => {
    event.preventDefault()
    const {userName, password} = this.state
    const userDetails = {userName, password}
    const url = 'https://apis.ccbp.in/login'
    const options = {
      method: 'POST',
      body: JSON.stringify(userDetails),
    }
    const response = await fetch(url, options)
    const data = await response.json()
    console.log(data.error_msg)
    if (response.ok === true) {
      this.onSubmitSuccess(data.jwt_token)
    } else {
      this.onSubmitFailure(data.error_msg)
    }
  }
```

```
  renderPasswordField = () => {
    const {password} = this.state
    return (
      <>
        <label className="input-label" htmlFor="password">
          PASSWORD
        </label>
        <input
          type="password"
          id="password"
          className="password-input-filed"
          value={password}
          onChange={this.onChangePassword}
          placeholder="Password"
        />
      </>
    )
  }

  renderUsernameField = () => {
    const {username} = this.state
    return (
      <>
        <label className="input-label" htmlFor="username">
          USERNAME
        </label>
        <input
          type="text"
          id="username"
          className="username-input-filed"
          value={username}
          onChange={this.onChangeUsername}
          placeholder="Username"
        />
      </>
```

```
    )
  }

  render() {
    const {showSubmitError, errorMsg} = this.state
    const jwtToken = Cookies.get('jwt_token')
    if (jwtToken !== undefined) {
      return <Redirect to="/" />
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            className="login-website-logo-desktop-image"
            alt="website logo"
          />
          <div className="input-container">{this.renderUsernameField()}</div>
          <div className="input-container">{this.renderPasswordField()}</div>
          <button type="submit" className="login-button">
            Login
          </button>
          {showSubmitError && <p className="error-message">{errorMsg}</p>}
        </form>
      </div>
    )
```

```
  }
}

export default LoginForm
```

Collapse

Solution:

File: src/components/LoginForm/index.js

JSX

```jsx
import {Component} from 'react'
import Cookies from 'js-cookie'
import {Redirect} from 'react-router-dom'

import './index.css'

class LoginForm extends Component {
  state = {
    username: '',
    password: '',
    showSubmitError: false,
    errorMsg: '',
  }

  onChangeUsername = event => {
    this.setState({username: event.target.value})
  }

  onChangePassword = event => {
    this.setState({password: event.target.value})
  }

  onSubmitSuccess = jwtToken => {
```

```
    const {history} = this.props


    Cookies.set('jwt_token', jwtToken, {
      expires: 30,
    })

    history.replace('/')
  }

  onSubmitFailure = errorMsg => {
    this.setState({showSubmitError: true, errorMsg})
  }

  submitForm = async event => {
    event.preventDefault()
    const {username, password} = this.state
    const userDetails = {username, password}
    const url = 'https://apis.ccbp.in/login'
    const options = {
      method: 'POST',
      body: JSON.stringify(userDetails),
    }
    const response = await fetch(url, options)
    const data = await response.json()
    console.log(data.error_msg)
    if (response.ok === true) {
      this.onSubmitSuccess(data.jwt_token)
    } else {
      this.onSubmitFailure(data.error_msg)
    }
  }

  renderPasswordField = () => {
    const {password} = this.state
    return (
```

```
      <>
        <label className="input-label" htmlFor="password">
          PASSWORD
        </label>
        <input
          type="password"
          id="password"
          className="password-input-filed"
          value={password}
          onChange={this.onChangePassword}
          placeholder="Password"
        />
      </>
    )
  }

  renderUsernameField = () => {
    const {username} = this.state
    return (
      <>
        <label className="input-label" htmlFor="username">
          USERNAME
        </label>
        <input
          type="text"
          id="username"
          className="username-input-filed"
          value={username}
          onChange={this.onChangeUsername}
          placeholder="Username"
        />
      </>
    )
  }

  render() {
```

```
    const {showSubmitError, errorMsg} = this.state
    const jwtToken = Cookies.get('jwt_token')
    if (jwtToken !== undefined) {
      return <Redirect to="/" />
    }
    return (
      <div className="login-form-container">
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
          className="login-website-logo-mobile-image"
          alt="website logo"
        />
        <img
          src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-login-img.png"
          className="login-image"
          alt="website login"
        />
        <form className="form-container" onSubmit={this.submitForm}>
          <img
            src="https://assets.ccbp.in/frontend/react-js/nxt-trendz-logo-img.png"
            className="login-website-logo-desktop-image"
            alt="website logo"
          />
          <div className="input-container">{this.renderUsernameField()}</div>
          <div className="input-container">{this.renderPasswordField()}</div>
          <button type="submit" className="login-button">
            Login
          </button>
          {showSubmitError && <p className="error-message">{errorMsg}</p>}
        </form>
      </div>
    )
  }
}

export default LoginForm
```

MARKED AS COMPLETE

Submit Feedback