**Facebook.js   
it is long code with log statements**import React, { useState, useEffect, useRef } from "react";

import FacebookLogin from "react-facebook-login";

import axios from "axios";

import "./Facebook.css";

import { Modal, Button, Form } from "react-bootstrap";

import { useNavigate } from "react-router-dom"; // useNavigate for navigation

import { toast } from "react-toastify"; // Import toast for notifications

export default function Facebook() {

  const [showModal, setShowModal] = useState(false);

  const [formData, setFormData] = useState({

    country\_code: "",

    phone: "",

    user\_name: "",

  });

  const [facebookData, setFacebookData] = useState(null);

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

  const [countries, setCountries] = useState([]); // State to store country data

  const [filteredCountries, setFilteredCountries] = useState([]);

  const [activeIndex, setActiveIndex] = useState(-1); // Track active index for keyboard navigation

  const navigate = useNavigate(); // Initialize useNavigate

  const suggestionRefs = useRef([]); //Ref to track suggesstions

  useEffect(() => {

    // Fetch country data from the API when the component mounts

    axios

      .get("https://api.silocloud.io/api/v1/public/countries")

      .then((response) => {

        console.log(response.data); // Log the API response

        // Use response.data.data.countries to get the array of countries

        const countryData = response.data.data.countries.map((country) => ({

          name: country.name, // Use country.name for the country name

          code: `+${country.phonecode}`, // Use country.phonecode for the phone code

        }));

        setCountries(countryData); // Update the state with the correct country data

      })

      .catch((error) => {

        console.log("Error fetching country data:", error);

      });

  }, []);

  const handleInputChange = (e) => {

    const { name, value } = e.target;

    setFormData({ ...formData, [name]: value });

    // Filter countries based on input

    const filtered = countries.filter(

      (country) =>

        country.code.startsWith(value) ||

        country.name.toLowerCase().startsWith(value.toLowerCase())

    );

    setFilteredCountries(filtered);

    setActiveIndex(-1); // Reset the active index on new input

  };

  const handleKeyDown = (e) => {

    if (e.key === "ArrowDown") {

      setActiveIndex((prevIndex) =>

        prevIndex < filteredCountries.length - 1 ? prevIndex + 1 : prevIndex

      );

    } else if (e.key === "ArrowUp") {

      setActiveIndex((prevIndex) =>

        prevIndex > 0 ? prevIndex - 1 : prevIndex

      );

    } else if (e.key === "Enter" && activeIndex >= 0) {

      handleCountrySelect(filteredCountries[activeIndex].code);

    }

  };

  useEffect(() => {

    // Scroll the active item into view

    if (activeIndex >= 0 && suggestionRefs.current[activeIndex]) {

      suggestionRefs.current[activeIndex].scrollIntoView({

        behavior: "smooth",

        block: "nearest",

      });

    }

  }, [activeIndex]);

  const handleCountrySelect = (code) => {

    setFormData({ ...formData, country\_code: code });

    setFilteredCountries([]); // Clear suggestions after selection

  };

  const handleClose = () => setShowModal(false);

  const handleShow = () => setShowModal(true);

  const validateForm = () => {

    const newErrors = {};

    if (!formData.country\_code) {

      newErrors.country\_code = "Country code is required.";

    } else if (!formData.country\_code.startsWith("+")) {

      newErrors.country\_code = "Country code should start with '+'";

    }

    if (!formData.phone) {

      newErrors.phone = "Phone number is required.";

    } else if (formData.phone.length !== 10) {

      newErrors.phone = "Phone number must be exactly 10 digits.";

    }

    if (!formData.user\_name) newErrors.user\_name = "Username is required.";

    return newErrors;

  };

  const handleSubmit = () => {

    const validationErrors = validateForm(); // Validate form

    if (Object.keys(validationErrors).length > 0) {

      setErrors(validationErrors); // Set error messages

      return; // Stop form submission if there are errors

    }

    // Prepare the data to send to PHP

    const nameArr = facebookData.name.split(" ");

    const firstname = nameArr[0] ? nameArr[0] : null;

    const lastname = nameArr.length > 1 ? nameArr[1] : null;

    const dataToSend = {

      first\_name: firstname,

      last\_name: lastname,

      email: facebookData.email,

      user\_name: formData.user\_name,

      country\_code: formData.country\_code,

      phone: formData.phone,

    };

    // Store session data after successful form submission

    sessionStorage.setItem("userSession", JSON.stringify(dataToSend));

    // Send data to PHP

    axios

      .post("http://localhost/php-react/insert.php", dataToSend)

      .then((result) => {

        console.log(result.data);

        toast.success("Logged in successfully!!"); // Display toast message

        // Navigate to /data and pass the data as state

        navigate("/data", { state: dataToSend });

      })

      .catch((error) => {

        console.log("Error sending data", error);

      });

    handleClose(); // Close the modal after submission

  };

  const responseFacebook = (response) => {

    console.log("In responseFacebook callback");

    if (response.accessToken) {

      console.log("Logged in successfully", response);

      setFacebookData(response); // Save Facebook data

      if (response.email) {

        checkUserExist(response.email); // Pass the email directly

      } else {

        console.log("Email not found in response");

        handleShow(); // Show modal or handle case where email is missing

      }

    } else {

      console.log("Login Failed", response); // Handle login failure

    }

  };

  const checkUserExist = (email) => {

    console.log("Checking if user exists with email:", email);

    const dataToSend = {

      email: email,

    };

    axios

      .post("http://localhost/php-react/check\_user.php", dataToSend)

      .then((result) => {

        console.log(result.data);

        if (result.data.exists) {

          console.log("User exists:", result.data.userData);

          sessionStorage.setItem(

            "userSession",

            JSON.stringify(result.data.userData)

          );

          navigate("/data", { state: result.data.userData });

        } else {

          handleShow(); // Show modal for additional details

        }

      })

      .catch((error) => {

        console.log("Error sending data", error);

      });

  };

  return (

    <div className="container mt-5">

      <FacebookLogin

        appId="1937254353445888"

        autoLoad={true}

        fields="name,email"

        callback={responseFacebook}

        icon="fa-facebook"

        cssClass="fb-btn"

      />

      {/\* Modal for input \*/}

      <Modal show={showModal} onHide={handleClose}>

        <Modal.Header closeButton className="popup-form">

          <Modal.Title>Please Enter Additional Details!</Modal.Title>

        </Modal.Header>

        <Modal.Body className="popup-form-body">

          <Form className="popup-content">

            {/\* Country code \*/}

            <Form.Group className="mb-3">

              <Form.Label style={{ fontSize: "18px" }}>Country Code</Form.Label>

              <Form.Control

                type="text"

                placeholder="Enter country code"

                name="country\_code"

                value={formData.country\_code}

                onChange={handleInputChange}

                onKeyDown={handleKeyDown} // Capture key events

                className="ip-details"

                autoComplete="off"

              />

              {filteredCountries.length > 0 && (

                <div className="suggestions-wrapper">

                  <ul className="suggestions-list">

                    {filteredCountries.map((country, index) => (

                      <li

                        key={index}

                        ref={(el) => (suggestionRefs.current[index] = el)} // Set ref to each suggestion

                        onClick={() => handleCountrySelect(country.code)}

                        className={index === activeIndex ? "active" : ""}

                      >

                        {country.name} ({country.code})

                      </li>

                    ))}

                  </ul>

                  {/\* Close button (X symbol) \*/}

                  <span

                    className="close-suggestions"

                    onClick={() => setFilteredCountries([])} // Clear the filteredCountries to hide the list

                  >

                    &times;

                  </span>

                </div>

              )}

            </Form.Group>

            {/\* User name \*/}

            <Form.Group className="mb-3">

              <Form.Label style={{ fontSize: "18px" }}>User Name</Form.Label>

              <Form.Control

                type="text"

                placeholder="Enter user name"

                name="user\_name"

                value={formData.user\_name}

                onChange={handleInputChange}

                className="ip-details"

              />

              {errors.user\_name && (

                <small className="text-danger">{errors.user\_name}</small>

              )}

            </Form.Group>

            {/\* Phone number \*/}

            <Form.Group className="mb-3">

              <Form.Label style={{ fontSize: "18px" }}>Phone Number</Form.Label>

              <Form.Control

                type="tel"

                placeholder="Enter phone number"

                name="phone"

                value={formData.phone}

                onChange={handleInputChange}

                className="ip-details"

              />

              {errors.phone && (

                <small className="text-danger">{errors.phone}</small>

              )}

            </Form.Group>

          </Form>

        </Modal.Body>

        <Modal.Footer>

          <Button variant="secondary" onClick={handleClose}>

            Close

          </Button>

          <Button variant="primary" onClick={handleSubmit}>

            Submit

          </Button>

        </Modal.Footer>

      </Modal>

    </div>

  );

}

**insert.php**<?php

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: POST");

header("Access-Control-Allow-Headers: Content-Type");

header("Content-Type: application/json; charset=UTF-8");

header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");

$data = json\_decode(file\_get\_contents("php://input"));

$first\_name = $data->first\_name;

$last\_name = $data->last\_name;

$user\_name = $data->user\_name;

$email = $data->email;

$country\_code = $data->country\_code;

$phone = $data->phone;

// $password = $data->password;

// Establish the database connection

$conn = mysqli\_connect("localhost", "root", "");

mysqli\_select\_db($conn, "fb-login");

// Prepare the SQL insert statement

$sql = "INSERT INTO registration (first\_name, last\_name, user\_name, email, country\_code, phone)

        VALUES ('$first\_name', '$last\_name', '$user\_name', '$email', '$country\_code', '$phone')";

$result = mysqli\_query($conn, $sql);

if ($result) {

    // // Fetch all emails from the result set

    // while ($row = mysqli\_fetch\_assoc($result)) {

    //     $emails[] = $row['email'];

    // }

    $response = [

        'status' => 'Valid',

        'message' => 'Successfully Created account',

    ];

} else {

    $response = [

        'status' => 'Invalid',

        'message' => 'Can\'t create your account here.

                        Try using another Email id',

        'email' => $email,

    ];

}

echo json\_encode($response);

mysqli\_close($conn);

?>

**Check\_user.php**<?php

header("Access-Control-Allow-Origin: http://localhost:3000"); // Allow your React app's origin

header("Access-Control-Allow-Methods: POST, OPTIONS"); // Allow POST and OPTIONS methods

header("Access-Control-Allow-Headers: Content-Type"); // Allow specific headers

 // Database configuration

    $host = 'localhost'; // Database host, usually 'localhost'

    $user = 'root'; // Your database username

    $password = ''; // Your database password

    $dbname = 'fb-login'; // Your database name

    // Create connection

    $conn = new mysqli($host, $user, $password, $dbname);

    // Check connection

    if ($conn->connect\_error) {

        die("Connection failed: " . $conn->connect\_error);

    }

// Check for preflight requests (OPTIONS)

if ($\_SERVER['REQUEST\_METHOD'] == 'OPTIONS') {

    exit; // Exit for preflight requests

}

// Get the POST data

$data = json\_decode(file\_get\_contents("php://input"));

// Check if email is set

if (isset($data->email)) {

    $email = $data->email;

    // Query to check if the user exists

    $query = "SELECT \* FROM registration WHERE email = '$email' LIMIT 1";

    $result = mysqli\_query($conn, $query);

    if (mysqli\_num\_rows($result) > 0) {

        // User exists

        $user = mysqli\_fetch\_assoc($result);

        echo json\_encode([

            "exists" => true,

            "userData" => $user

        ]);

    } else {

        // User does not exist

        echo json\_encode([

            "exists" => false

        ]);

    }

} else {

    echo json\_encode(["exists" => false]);

}

?>