**Check\_user.php(runs the code successfully)**

<?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]);

}

?>

**Facebook.js (runs the code successfully)**

import React, { useState } 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

export default function Facebook() {

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

  const [formData, setFormData] = useState({

    country\_code: "",

    phone: "",

    user\_name: "",

    // password: ""

  });

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

  const [errors, setErrors] = useState({}); // State to track validation errors

  const navigate = useNavigate(); // Initialize useNavigate

  const handleInputChange = (e) => {

    const { name, value } = e.target;

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

    setErrors({ ...errors, [name]: "" }); // Clear error for specific field on change

  };

  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,

      };

      // Send data to PHP

      axios

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

          .then((result) => {

              console.log(result.data);

              alert(result.data.status + "\n" + result.data.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

          // Only call checkUserExist if facebookData is valid and contains an email

          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 // Use the email passed to this function

      };

      axios

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

          .then((result) => {

              console.log(result.data);

              if (result.data.exists) {

                  // User exists

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

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

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

              } else {

                  // User does not exist

                  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">

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

              <Form.Label>Country Code</Form.Label>

              <Form.Control

                type="text"

                placeholder="Enter country code"

                name="country\_code"

                value={formData.country\_code}

                onChange={handleInputChange}

                className="ip-details"

              />

              {errors.country\_code && (

                <small className="text-danger">{errors.country\_code}</small>

              )}

            </Form.Group>

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

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

              <Form.Label>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>

          </Form>

        </Modal.Body>

        <Modal.Footer>

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

            Close

          </Button>

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

            Submit

          </Button>

        </Modal.Footer>

      </Modal>

    </div>

  );

}