Slot3\_Excise1  
Code chương trình:

import { useState } from 'react'

import { Container, Row, Col, Table, Form, Button, Badge, Alert } from 'react-bootstrap'

import './App.css'

function App() {

  // Danh sách công ty

  const companies = [

    { name: "Company One", category: "Finance", start: 1981, end: 2004 },

    { name: "Company Two", category: "Retail", start: 1992, end: 2008 },

    { name: "Company Three", category: "Auto", start: 1999, end: 2007 },

    { name: "Company Four", category: "Retail", start: 1989, end: 2010 },

    { name: "Company Five", category: "Technology", start: 2009, end: 2014 },

    { name: "Company Six", category: "Finance", start: 1987, end: 2010 },

    { name: "Company Seven", category: "Auto", start: 1986, end: 1996 },

    { name: "Company Eight", category: "Technology", start: 2011, end: 2016 },

    { name: "Company Nine", category: "Retail", start: 1981, end: 1989 }

  ];

  // States cho các chức năng

  const [searchTerm, setSearchTerm] = useState('')

  const [sortBy, setSortBy] = useState('default')

  const [filterCategory, setFilterCategory] = useState('all')

  // Lấy danh sách categories duy nhất

  const categories = [...new Set(companies.map(company => company.category))]

  // Xử lý tìm kiếm và lọc

  const getFilteredAndSortedCompanies = () => {

    let filtered = companies

    // Tìm kiếm theo tên

    if (searchTerm) {

      filtered = filtered.filter(company =>

        company.name.toLowerCase().includes(searchTerm.toLowerCase())

      )

    }

    // Lọc theo category

    if (filterCategory !== 'all') {

      filtered = filtered.filter(company => company.category === filterCategory)

    }

    // Sắp xếp

    if (sortBy === 'yearAsc') {

      filtered = [...filtered].sort((a, b) => a.start - b.start)

    } else if (sortBy === 'yearDesc') {

      filtered = [...filtered].sort((a, b) => b.start - a.start)

    }

    return filtered

  }

  const filteredCompanies = getFilteredAndSortedCompanies()

  const getCategoryBadgeVariant = (category) => {

    switch(category) {

      case 'Technology': return 'primary'

      case 'Finance': return 'success'

      case 'Auto': return 'warning'

      case 'Retail': return 'info'

      default: return 'secondary'

    }

  }

  return (

    <Container className="py-4">

      <h1 className="text-center mb-4">Company Management System</h1>

      <Row className="mb-4">

        <Col md={3}>

          <Form.Group>

            <Form.Label>Tìm kiếm công ty</Form.Label>

            <Form.Control

              type="text"

              placeholder="Nhập tên công ty..."

              value={searchTerm}

              onChange={(e) => setSearchTerm(e.target.value)}

            />

          </Form.Group>

        </Col>

        <Col md={3}>

          <Form.Group>

            <Form.Label>Sắp xếp</Form.Label>

            <Form.Select

              value={sortBy}

              onChange={(e) => setSortBy(e.target.value)}

            >

              <option value="default">Mặc định</option>

              <option value="yearAsc">Năm tăng dần</option>

              <option value="yearDesc">Năm giảm dần</option>

            </Form.Select>

          </Form.Group>

        </Col>

        <Col md={3}>

          <Form.Group>

            <Form.Label>Lọc theo danh mục</Form.Label>

            <Form.Select

              value={filterCategory}

              onChange={(e) => setFilterCategory(e.target.value)}

            >

              <option value="all">Tất cả danh mục</option>

              {categories.map(category => (

                <option key={category} value={category}>{category}</option>

              ))}

            </Form.Select>

          </Form.Group>

        </Col>

        <Col md={3} className="d-flex align-items-end">

          <Button

            variant="outline-secondary"

            onClick={() => {

              setSearchTerm('')

              setSortBy('default')

              setFilterCategory('all')

            }}

            className="w-100"

          >

            Reset

          </Button>

        </Col>

      </Row>

      <Alert variant="light" className="mb-3">

        <strong>Tìm thấy {filteredCompanies.length} công ty</strong>

      </Alert>

      {filteredCompanies.length > 0 ? (

        <Table striped bordered hover responsive>

          <thead className="table-dark">

            <tr>

              <th>Tên Công Ty</th>

              <th>Danh Mục</th>

              <th>Năm Bắt Đầu</th>

              <th>Năm Kết Thúc</th>

              <th>Thời Gian Hoạt Động</th>

            </tr>

          </thead>

          <tbody>

            {filteredCompanies.map((company, index) => (

              <tr key={company.name}>

                <td className="fw-bold">{company.name}</td>

                <td>

                  <Badge bg={getCategoryBadgeVariant(company.category)}>

                    {company.category}

                  </Badge>

                </td>

                <td>{company.start}</td>

                <td>{company.end}</td>

                <td>

                  <Badge bg="dark">

                    {company.end - company.start} năm

                  </Badge>

                </td>

              </tr>

            ))}

          </tbody>

        </Table>

      ) : (

        <Alert variant="warning" className="text-center">

          <h4>No result</h4>

          <p className="mb-0">Không tìm thấy công ty nào phù hợp với tiêu chí tìm kiếm.</p>

        </Alert>

      )}

    </Container>

  )

}

export default App  
  
  
  
  
  
  
