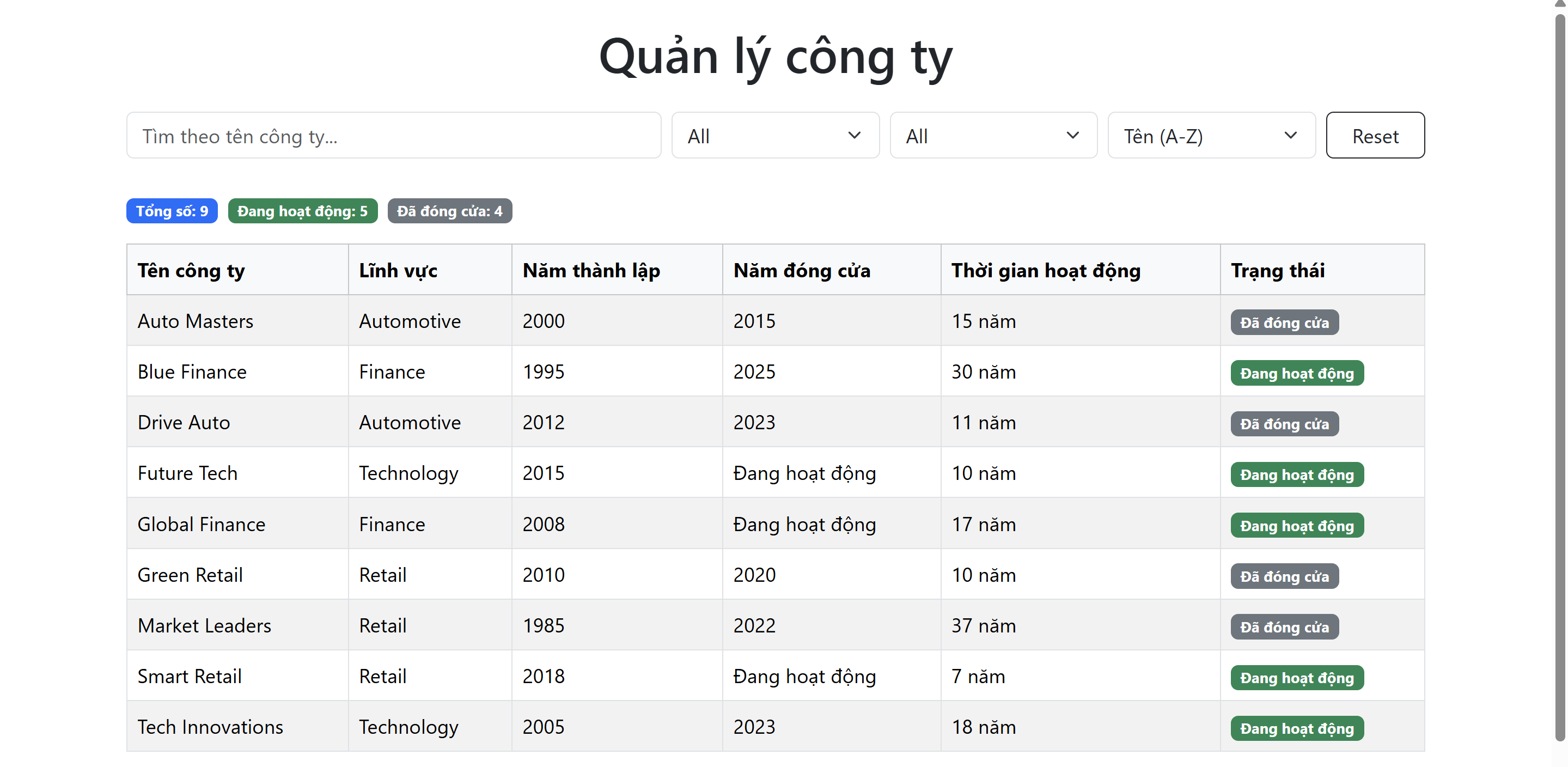
# EX1\_Slot3



## CompanyTable.js

import React, { useState, useEffect } from 'react';

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

// Component chính quản lý bảng công ty

const CompanyTable = () => {

  // Dữ liệu ban đầu của các công ty

  const initialCompanies = [

    { id: 1, name: "Tech Innovations", category: "Technology", start: 2005, end: 2023, status: "Active" },

    { id: 2, name: "Green Retail", category: "Retail", start: 2010, end: 2020, status: "Closed" },

    { id: 3, name: "Blue Finance", category: "Finance", start: 1995, end: 2025, status: "Active" },

    { id: 4, name: "Auto Masters", category: "Automotive", start: 2000, end: 2015, status: "Closed" },

    { id: 5, name: "Market Leaders", category: "Retail", start: 1985, end: 2022, status: "Closed" },

    { id: 6, name: "Future Tech", category: "Technology", start: 2015, end: null, status: "Active" },

    { id: 7, name: "Global Finance", category: "Finance", start: 2008, end: null, status: "Active" },

    { id: 8, name: "Drive Auto", category: "Automotive", start: 2012, end: 2023, status: "Closed" },

    { id: 9, name: "Smart Retail", category: "Retail", start: 2018, end: null, status: "Active" }

  ];

  // State quản lý danh sách công ty, tìm kiếm, sắp xếp, lọc

  const [companies, setCompanies] = useState(initialCompanies);

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

  const [sortOption, setSortOption] = useState('name-asc');

  const [filterCategory, setFilterCategory] = useState('All');

  const [filterStatus, setFilterStatus] = useState('All');

  // Lấy danh sách category và status duy nhất

  const categories = ['All', ...new Set(initialCompanies.map(company => company.category))];

  const statuses = ['All', ...new Set(initialCompanies.map(company => company.status))];

  // Hàm tính thời gian hoạt động

  const calculateDuration = (start, end) => {

    const endYear = end || new Date().getFullYear();

    return endYear - start;

  };

  // Hàm áp dụng các bộ lọc và sắp xếp

  const applyFilters = () => {

    let results = [...initialCompanies];

    // Lọc theo từ khóa tìm kiếm

    if (searchTerm) {

      results = results.filter(company =>

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

      );

    }

    // Lọc theo category

    if (filterCategory !== 'All') {

      results = results.filter(company => company.category === filterCategory);

    }

    // Lọc theo trạng thái

    if (filterStatus !== 'All') {

      results = results.filter(company => company.status === filterStatus);

    }

    // Sắp xếp kết quả

    results.sort((a, b) => {

      switch (sortOption) {

        case 'name-asc':

          return a.name.localeCompare(b.name);

        case 'name-desc':

          return b.name.localeCompare(a.name);

        case 'start-asc':

          return a.start - b.start;

        case 'start-desc':

          return b.start - a.start;

        case 'duration-asc':

          return calculateDuration(a.start, a.end) - calculateDuration(b.start, b.end);

        case 'duration-desc':

          return calculateDuration(b.start, b.end) - calculateDuration(a.start, a.end);

        default:

          return 0;

      }

    });

    setCompanies(results);

  };

  // Gọi applyFilters khi state thay đổi

  useEffect(() => {

    applyFilters();

  }, [searchTerm, sortOption, filterCategory, filterStatus]);

  // Hàm reset tất cả bộ lọc

  const resetFilters = () => {

    setSearchTerm('');

    setSortOption('name-asc');

    setFilterCategory('All');

    setFilterStatus('All');

  };

  return (

    <Container className="py-4">

      {/\* Tiêu đề trang \*/}

      <h1 className="text-center mb-4 text-dark">Quản lý công ty</h1>

      {/\* Khu vực bộ lọc \*/}

      <Row className="mb-4 g-2 align-items-end">

        <Col xs={12} md={5}>

          {/\* Ô tìm kiếm \*/}

          <Form.Control

            type="text"

            placeholder="Tìm theo tên công ty..."

            value={searchTerm}

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

            className="mb-2"

          />

        </Col>

        <Col xs={6} md={2}>

          {/\* Lọc theo category \*/}

          <Form.Select

            value={filterCategory}

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

            className="mb-2"

          >

            {categories.map(category => (

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

            ))}

          </Form.Select>

        </Col>

        <Col xs={6} md={2}>

          {/\* Lọc theo trạng thái \*/}

          <Form.Select

            value={filterStatus}

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

            className="mb-2"

          >

            {statuses.map(status => (

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

            ))}

          </Form.Select>

        </Col>

        <Col xs={6} md={2}>

          {/\* Sắp xếp \*/}

          <Form.Select

            value={sortOption}

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

            className="mb-2"

          >

            <option value="name-asc">Tên (A-Z)</option>

            <option value="name-desc">Tên (Z-A)</option>

            <option value="start-asc">Năm thành lập (Cũ nhất)</option>

            <option value="start-desc">Năm thành lập (Mới nhất)</option>

            <option value="duration-asc">Thời gian hoạt động (Ngắn nhất)</option>

            <option value="duration-desc">Thời gian hoạt động (Dài nhất)</option>

          </Form.Select>

        </Col>

        <Col xs={6} md={1}>

          {/\* Nút reset bộ lọc \*/}

          <Button variant="outline-dark" onClick={resetFilters} className="w-100 mb-2">

            Reset

          </Button>

        </Col>

      </Row>

      {/\* Hiển thị số liệu thống kê \*/}

      {companies.length > 0 && (

        <div className="mb-3 d-flex gap-2 flex-wrap">

          <Badge bg="primary" className="text-white">

            Tổng số: {companies.length}

          </Badge>

          <Badge bg="success">

            Đang hoạt động: {companies.filter(c => c.status === 'Active').length}

          </Badge>

          <Badge bg="secondary">

            Đã đóng cửa: {companies.filter(c => c.status === 'Closed').length}

          </Badge>

        </div>

      )}

      {/\* Bảng dữ liệu công ty \*/}

      {companies.length > 0 ? (

        <Table striped bordered hover responsive>

          <thead className="table-light">

            <tr>

              <th>Tên công ty</th>

              <th>Lĩnh vực</th>

              <th>Năm thành lập</th>

              <th>Năm đóng cửa</th>

              <th>Thời gian hoạt động</th>

              <th>Trạng thái</th>

            </tr>

          </thead>

          <tbody>

            {companies.map(company => (

              <tr key={company.id}>

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

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

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

                <td>{company.end || 'Đang hoạt động'}</td>

                <td>{calculateDuration(company.start, company.end)} năm</td>

                <td>

                  <Badge bg={company.status === 'Active' ? 'success' : 'secondary'}>

                    {company.status === 'Active' ? 'Đang hoạt động' : 'Đã đóng cửa'}

                  </Badge>

                </td>

              </tr>

            ))}

          </tbody>

        </Table>

      ) : (

        <div className="text-center py-4">

          <h4 className="text-muted">Không tìm thấy công ty nào phù hợp</h4>

          <Button variant="outline-dark" onClick={resetFilters} className="mt-2">

            Reset bộ lọc

          </Button>

        </div>

      )}

    </Container>

  );

};

export default CompanyTable;

## App.js

import React from 'react';

import CompanyTable from './CompanyTable';

function App() {

  return (

    <div className="App">

      <CompanyTable />

    </div>

  );

}

export default App;