Slot3\_Ex2  
import { useState, useMemo } from 'react'

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

import { persons } from './person.js'

import './App.css'

function App() {

  // State cho các tab

  const [activeTab, setActiveTab] = useState(1)

  // State cho yêu cầu 1: Sort first name

  const [sortOrder, setSortOrder] = useState('asc') // 'asc' hoặc 'desc'

  // State cho yêu cầu 2: Filter theo age và skill

  const [minAge, setMinAge] = useState('')

  const [maxAge, setMaxAge] = useState('')

  const [selectedSkill, setSelectedSkill] = useState('')

  // State cho yêu cầu 4: Search và sort đa tiêu chí

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

  // Lấy tất cả skills unique cho dropdown

  const allSkills = useMemo(() => {

    const skillSet = new Set()

    persons.forEach(person => {

      person.skills.forEach(skill => skillSet.add(skill))

    })

    return Array.from(skillSet).sort()

  }, [])

  // Yêu cầu 1: Sort danh sách theo firstName

  const sortedPersons = useMemo(() => {

    return [...persons].sort((a, b) => {

      if (sortOrder === 'asc') {

        return a.firstName.localeCompare(b.firstName)

      } else {

        return b.firstName.localeCompare(a.firstName)

      }

    })

  }, [sortOrder])

  // Yêu cầu 2: Filter theo age và skill

  const filteredPersons = useMemo(() => {

    return persons.filter(person => {

      const { age, skills } = person

      // Filter theo age

      const minAgeNum = minAge ? parseInt(minAge) : 0

      const maxAgeNum = maxAge ? parseInt(maxAge) : 100

      const ageMatch = age >= minAgeNum && age <= maxAgeNum

      // Filter theo skill

      const skillMatch = selectedSkill ? skills.includes(selectedSkill) : true

      return ageMatch && skillMatch

    })

  }, [minAge, maxAge, selectedSkill])

  // Yêu cầu 3: Ranking skills bằng reduce

  const skillRanking = useMemo(() => {

    const skillCount = persons.reduce((acc, person) => {

      person.skills.forEach(skill => {

        acc[skill] = (acc[skill] || 0) + 1

      })

      return acc

    }, {})

    return Object.entries(skillCount)

      .map(([skill, count]) => ({ skill, count }))

      .sort((a, b) => b.count - a.count)

  }, [])

  // Yêu cầu 4: Search và sort đa tiêu chí

  const searchAndSortedPersons = useMemo(() => {

    // Filter theo search term

    const filtered = persons.filter(person => {

      const fullName = `${person.firstName} ${person.lastName}`.toLowerCase()

      return fullName.includes(searchTerm.toLowerCase())

    })

    // Sort đa tiêu chí: isActive (true trước), age tăng dần, lastName A→Z

    return filtered.sort((a, b) => {

      // Ưu tiên isActive (true trước)

      if (a.isActive !== b.isActive) {

        return b.isActive - a.isActive

      }

      // Sau đó age tăng dần

      if (a.age !== b.age) {

        return a.age - b.age

      }

      // Cuối cùng lastName A→Z

      return a.lastName.localeCompare(b.lastName)

    })

  }, [searchTerm])

  // Yêu cầu 4: Statistics bằng reduce

  const statistics = useMemo(() => {

    return searchAndSortedPersons.reduce((acc, person) => {

      return {

        totalPeople: acc.totalPeople + 1,

        totalAge: acc.totalAge + person.age,

        activeCount: acc.activeCount + (person.isActive ? 1 : 0)

      }

    }, { totalPeople: 0, totalAge: 0, activeCount: 0 })

  }, [searchAndSortedPersons])

  const averageAge = statistics.totalPeople > 0 ? (statistics.totalAge / statistics.totalPeople).toFixed(1) : 0

  return (

    <Container className="mt-4">

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

      {/\* Navigation Tabs \*/}

      <Row className="mb-4">

        <Col>

          <div className="d-flex gap-2">

            <Button

              variant={activeTab === 1 ? "primary" : "outline-primary"}

              onClick={() => setActiveTab(1)}

            >

              1. Danh sách & Sort

            </Button>

            <Button

              variant={activeTab === 2 ? "primary" : "outline-primary"}

              onClick={() => setActiveTab(2)}

            >

              2. Lọc theo tuổi & skill

            </Button>

            <Button

              variant={activeTab === 3 ? "primary" : "outline-primary"}

              onClick={() => setActiveTab(3)}

            >

              3. Ranking skills

            </Button>

            <Button

              variant={activeTab === 4 ? "primary" : "outline-primary"}

              onClick={() => setActiveTab(4)}

            >

              4. Tìm kiếm & thống kê

            </Button>

          </div>

        </Col>

      </Row>

      {/\* Yêu cầu 1: Hiển thị danh sách với Sort \*/}

      {activeTab === 1 && (

        <Card>

          <Card.Header>

            <h3>1. Danh sách với Sort First Name</h3>

            <Button

              variant="info"

              onClick={() => setSortOrder(sortOrder === 'asc' ? 'desc' : 'asc')}

            >

              Sort First Name: {sortOrder === 'asc' ? 'A→Z' : 'Z→A'}

            </Button>

          </Card.Header>

          <Card.Body>

            <ul className="list-group">

              {sortedPersons.map(person => (

                <li key={person.id} className="list-group-item">

                  <strong>{person.firstName} {person.lastName}</strong> -

                  Age: {person.age} -

                  City: {person.city} -

                  Skills: {person.skills.join(', ')}

                </li>

              ))}

            </ul>

          </Card.Body>

        </Card>

      )}

      {/\* Yêu cầu 2: Lọc theo age và skill \*/}

      {activeTab === 2 && (

        <Card>

          <Card.Header>

            <h3>2. Lọc theo khoảng độ tuổi và skill</h3>

          </Card.Header>

          <Card.Body>

            <Row className="mb-3">

              <Col md={3}>

                <Form.Group>

                  <Form.Label>Min Age</Form.Label>

                  <Form.Control

                    type="number"

                    value={minAge}

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

                    placeholder="Tuổi tối thiểu"

                  />

                </Form.Group>

              </Col>

              <Col md={3}>

                <Form.Group>

                  <Form.Label>Max Age</Form.Label>

                  <Form.Control

                    type="number"

                    value={maxAge}

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

                    placeholder="Tuổi tối đa"

                  />

                </Form.Group>

              </Col>

              <Col md={3}>

                <Form.Group>

                  <Form.Label>Skill</Form.Label>

                  <Form.Select

                    value={selectedSkill}

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

                  >

                    <option value="">Tất cả skills</option>

                    {allSkills.map(skill => (

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

                    ))}

                  </Form.Select>

                </Form.Group>

              </Col>

            </Row>

            {filteredPersons.length > 0 ? (

              <ul className="list-group">

                {filteredPersons.map(person => (

                  <li key={person.id} className="list-group-item">

                    {person.firstName} - {person.lastName} - Skills: {person.skills.join(', ')}

                  </li>

                ))}

              </ul>

            ) : (

              <Alert variant="warning">No found.</Alert>

            )}

          </Card.Body>

        </Card>

      )}

      {/\* Yêu cầu 3: Ranking skills \*/}

      {activeTab === 3 && (

        <Card>

          <Card.Header>

            <h3>3. Bảng xếp hạng skill</h3>

          </Card.Header>

          <Card.Body>

            <Table striped bordered hover>

              <thead>

                <tr>

                  <th>Skill</th>

                  <th>Count</th>

                </tr>

              </thead>

              <tbody>

                {skillRanking.map((item, index) => (

                  <tr key={item.skill}>

                    <td className={index === 0 ? "fw-bold" : ""}>

                      {item.skill}

                      {index === 0 && <Badge bg="warning" className="ms-2">TOP 1</Badge>}

                    </td>

                    <td className={index === 0 ? "fw-bold" : ""}>

                      {item.count}

                    </td>

                  </tr>

                ))}

              </tbody>

            </Table>

          </Card.Body>

        </Card>

      )}

      {/\* Yêu cầu 4: Tìm kiếm và thống kê \*/}

      {activeTab === 4 && (

        <Card>

          <Card.Header>

            <h3>4. Tìm kiếm theo tên và thống kê</h3>

          </Card.Header>

          <Card.Body>

            <Row className="mb-3">

              <Col md={6}>

                <Form.Group>

                  <Form.Label>Search by Name</Form.Label>

                  <Form.Control

                    type="text"

                    value={searchTerm}

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

                    placeholder="Nhập tên để tìm kiếm..."

                  />

                </Form.Group>

              </Col>

            </Row>

            <Row>

              <Col md={8}>

                <h5>Kết quả (Sort: Active → Age ↑ → LastName A→Z)</h5>

                <ul className="list-group">

                  {searchAndSortedPersons.map(person => (

                    <li key={person.id} className="list-group-item d-flex justify-content-between align-items-center">

                      <span>

                        {person.firstName} {person.lastName} - Age: {person.age}

                      </span>

                      <Badge bg={person.isActive ? "success" : "secondary"}>

                        {person.isActive ? "Active" : "Inactive"}

                      </Badge>

                    </li>

                  ))}

                </ul>

              </Col>

              <Col md={4}>

                <Card className="bg-light">

                  <Card.Header>

                    <h5>Statistics</h5>

                  </Card.Header>

                  <Card.Body>

                    <p><strong>Tổng người:</strong> {statistics.totalPeople}</p>

                    <p><strong>Tuổi trung bình:</strong> {averageAge}</p>

                    <p><strong>Số người active:</strong> {statistics.activeCount}</p>

                  </Card.Body>

                </Card>

              </Col>

            </Row>

          </Card.Body>

        </Card>

      )}

    </Container>

  )

}

export default App

  
  
  
