**Exercise 1: User Directory with Lists and Keys**

**Question:**

1. Create a React application to display a list of users.
2. The application should contain three components: App, UserList, and UserItem.
   * App should hold an array of user objects with properties id, name, and age.
   * Pass the list of users to UserList as a prop.
   * UserList should iterate over the list using .map() and render each user using the UserItem component.
3. Each UserItem should display the user's name and age.
4. Use id as the key for each user.
5. Style each component with its own separate CSS file to make the app visually appealing.

**Solution:**

**Folder Structure**

react-list-keys

├── src

│ ├── App.js

│ ├── App.css

│ ├── components

│ │ ├── UserList.js

│ │ ├── UserList.css

│ │ ├── UserItem.js

│ │ ├── UserItem.css

│ ├── index.js

**App.js**

import React from "react";

import UserList from "./components/UserList";

import "./App.css";

function App() {

const users = [

{ id: 1, name: "Alice", age: 25 },

{ id: 2, name: "Bob", age: 30 },

{ id: 3, name: "Charlie", age: 35 },

];

return (

<div className="app">

<h1>User Directory</h1>

<UserList users={users} />

</div>

);

}

export default App;

**App.css**

.app {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

h1 {

color: #4a90e2;

}

**UserList.js**

import React from "react";

import UserItem from "./UserItem";

import "./UserList.css";

function UserList({ users }) {

return (

<div className="user-list">

{users.map((user) => (

<UserItem key={user.id} user={user} />

))}

</div>

);

}

export default UserList;

**UserList.css**

.user-list {

display: flex;

flex-direction: column;

gap: 10px;

margin: 20px auto;

max-width: 400px;

}

**UserItem.js**

import React from "react";

import "./UserItem.css";

function UserItem({ user }) {

return (

<div className="user-item">

<h2>{user.name}</h2>

<p>Age: {user.age}</p>

</div>

);

}

export default UserItem;

**UserItem.css**

.user-item {

background-color: #f4f4f4;

border: 1px solid #ccc;

border-radius: 8px;

padding: 10px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.user-item h2 {

color: #333;

margin: 0;

}

.user-item p {

color: #666;

margin: 5px 0 0;

}

**index.js**

import React from "react";

import ReactDOM from "react-dom";

import App from "./App";

ReactDOM.render(<App />, document.getElementById("root"));

**Exercise Task**

1. **Modify the List**:
   * Add more users with different attributes (e.g., location, profession).
   * Display the new attributes in the UserItem component.
2. **Add Styling**:
   * Update the CSS files to make the application visually appealing.
3. **Filter Feature**:
   * Add an input box in UserList to filter users by name.
4. **Extend Functionality**:
   * Include a button in UserItem to display more details (e.g., toggling visibility of additional user information).

**Exercise 2: Extended User Directory with Filter**

**Question:**

1. Extend the previous application to include a search feature.
2. Add an input field in the UserList component to filter users by name.
3. The filtered list should update in real-time as the user types in the input field.
4. Use a controlled input for managing the search term state.

**Solution:**

**UserList.js (Updated)**

import React, { useState } from "react";

import UserItem from "./UserItem";

import "./UserList.css";

function UserList({ users }) {

const [search, setSearch] = useState("");

const filteredUsers = users.filter((user) =>

user.name.toLowerCase().includes(search.toLowerCase())

);

return (

<div className="user-list">

<input

type="text"

placeholder="Search by name"

value={search}

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

className="search-input"

/>

{filteredUsers.map((user) => (

<UserItem key={user.id} user={user} />

))}

</div>

);

}

export default UserList;

**UserList.css (Updated)**

.search-input {

margin-bottom: 20px;

padding: 8px;

border: 1px solid #ccc;

border-radius: 5px;

width: 100%;

}

**Exercise 3: Toggle User Details**

**Question:**

1. Update the UserItem component to include a "Show Details" button.
2. When clicked, toggle the visibility of additional user information (e.g., email and profession).
3. The additional information should only be visible when the button is clicked.

**Solution:**

**UserItem.js (Updated)**

import React, { useState } from "react";

import "./UserItem.css";

function UserItem({ user }) {

const [showDetails, setShowDetails] = useState(false);

return (

<div className="user-item">

<h2>{user.name}</h2>

<p>Age: {user.age}</p>

<button onClick={() => setShowDetails(!showDetails)}>

{showDetails ? "Hide Details" : "Show Details"}

</button>

{showDetails && (

<div className="details">

<p>Email: {user.email || "N/A"}</p>

<p>Profession: {user.profession || "N/A"}</p>

</div>

)}

</div>

);

}

export default UserItem;

**UserItem.css (Updated)**

.details {

margin-top: 10px;

background-color: #eaeaea;

padding: 10px;

border-radius: 5px;

}

**Additional User Data in App.js**

const users = [

{ id: 1, name: "Alice", age: 25, email: "alice@outlook.com", profession: "Engineer" },

{ id: 2, name: "Bob", age: 30, email: "bob@gmail.com", profession: "Designer" },

{ id: 3, name: "Charlie", age: 35, email: "charlie@outlook.com", profession: "Manager" },

];

**Exercise 4: Add User Functionality**

**Question:**

1. Add functionality in the App component to allow the user to add a new user to the list.
2. Include a form with inputs for name and age.
3. The new user should be appended to the list and displayed in the UserList.

**Solution: Add User Functionality**

**App.js (Updated)**

import React, { useState } from "react";

import UserList from "./UserList";

import "./App.css";

function App() {

const [users, setUsers] = useState([

{ id: 1, name: "Alice", age: 25 },

{ id: 2, name: "Bob", age: 30 },

{ id: 3, name: "Charlie", age: 35 },

]);

const [name, setName] = useState("");

const [age, setAge] = useState("");

const addUser = (e) => {

e.preventDefault();

if (name && age) {

setUsers([...users, { id: Date.now(), name, age: Number(age) }]);

setName("");

setAge("");

}

};

return (

<div className="app">

<h1>User Directory</h1>

<form onSubmit={addUser} className="user-form">

<input

type="text"

placeholder="Name"

value={name}

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

required

/>

<input

type="number"

placeholder="Age"

value={age}

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

required

/>

<button type="submit">Add User</button>

</form>

<UserList users={users} />

</div>

);

}

export default App;

**App.css**

.app {

font-family: Arial, sans-serif;

padding: 20px;

}

.user-form {

margin-bottom: 20px;

display: flex;

gap: 10px;

}

.user-form input {

padding: 8px;

border: 1px solid #ccc;

border-radius: 5px;

}

.user-form button {

padding: 8px 12px;

background-color: #007bff;

color: #fff;

border: none;

border-radius: 5px;

cursor: pointer;

}

.user-form button:hover {

background-color: #0056b3;

}

**Exercise 5: Delete User Functionality**

**Question:**

1. Add a "Delete" button to each UserItem.
2. Clicking the button should remove the user from the list.
3. The updated list should reflect immediately.

**Solution: Delete User Functionality**

**UserItem.js (Updated)**

import React from "react";

import "./UserItem.css";

function UserItem({ user, onDelete }) {

return (

<div className="user-item">

<h2>{user.name}</h2>

<p>Age: {user.age}</p>

<button onClick={() => onDelete(user.id)} className="delete-button">

Delete

</button>

</div>

);

}

export default UserItem;

**UserItem.css (Updated)**

.user-item {

border: 1px solid #ccc;

padding: 10px;

margin-bottom: 10px;

border-radius: 5px;

}

.delete-button {

background-color: #dc3545;

color: #fff;

border: none;

padding: 8px 12px;

border-radius: 5px;

cursor: pointer;

}

.delete-button:hover {

background-color: #c82333;

}

**UserList.js (Updated)**

import React from "react";

import UserItem from "./UserItem";

import "./UserList.css";

function UserList({ users, onDelete }) {

return (

<div className="user-list">

{users.map((user) => (

<UserItem key={user.id} user={user} onDelete={onDelete} />

))}

</div>

);

}

export default UserList;

**UserList.css**

.user-list {

display: flex;

flex-direction: column;

gap: 10px;

}

**App.js (Final Version for Both Add and Delete)**

javascript

Copy code

import React, { useState } from "react";

import UserList from "./UserList";

import "./App.css";

function App() {

const [users, setUsers] = useState([

{ id: 1, name: "Alice", age: 25 },

{ id: 2, name: "Bob", age: 30 },

{ id: 3, name: "Charlie", age: 35 },

]);

const [name, setName] = useState("");

const [age, setAge] = useState("");

const addUser = (e) => {

e.preventDefault();

if (name && age) {

setUsers([...users, { id: Date.now(), name, age: Number(age) }]);

setName("");

setAge("");

}

};

const deleteUser = (id) => {

setUsers(users.filter((user) => user.id !== id));

};

return (

<div className="app">

<h1>User Directory</h1>

<form onSubmit={addUser} className="user-form">

<input

type="text"

placeholder="Name"

value={name}

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

required

/>

<input

type="number"

placeholder="Age"

value={age}

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

required

/>

<button type="submit">Add User</button>

</form>

<UserList users={users} onDelete={deleteUser} />

</div>

);

}

export default App;

**Exercise 6: Toggle User Details**

**Question**

Create a React application with the following requirements:

1. Display a list of users with their names.
2. Include a button for each user that toggles the visibility of their details (age and email).
3. Use at least three components:
   * App to manage the state.
   * **UserList** to display the list of users.
   * **UserItem** to handle individual user details.
4. Add CSS for each component in separate files.

**Solution**

**App.js**

import React, { useState } from "react";

import UserList from "./UserList";

import "./App.css";

function App() {

const [users, setUsers] = useState([

{ id: 1, name: "Alice", age: 25, email: "alice@example.com" },

{ id: 2, name: "Bob", age: 30, email: "bob@example.com" },

{ id: 3, name: "Charlie", age: 35, email: "charlie@example.com" },

]);

return (

<div className="app">

<h1>User List</h1>

<UserList users={users} />

</div>

);

}

export default App;

**App.css**

.app {

font-family: Arial, sans-serif;

padding: 20px;

text-align: center;

}

**UserList.js**

import React from "react";

import UserItem from "./UserItem";

import "./UserList.css";

function UserList({ users }) {

return (

<div className="user-list">

{users.map((user) => (

<UserItem key={user.id} user={user} />

))}

</div>

);

}

export default UserList;

**UserList.css**

.user-list {

display: flex;

flex-direction: column;

gap: 15px;

align-items: center;

}

**UserItem.js**

import React, { useState } from "react";

import "./UserItem.css";

function UserItem({ user }) {

const [showDetails, setShowDetails] = useState(false);

return (

<div className="user-item">

<h2>{user.name}</h2>

<button

className="toggle-button"

onClick={() => setShowDetails((prev) => !prev)}

>

{showDetails ? "Hide Details" : "Show Details"}

</button>

{showDetails && (

<div className="user-details">

<p>Age: {user.age}</p>

<p>Email: {user.email}</p>

</div>

)}

</div>

);

}

export default UserItem;

**UserItem.css**

.user-item {

border: 1px solid #ccc;

padding: 15px;

border-radius: 8px;

text-align: center;

width: 300px;

background-color: #f9f9f9;

}

.toggle-button {

margin-top: 10px;

padding: 8px 12px;

border: none;

background-color: #007bff;

color: white;

border-radius: 5px;

cursor: pointer;

}

.toggle-button:hover {

background-color: #0056b3;

}

.user-details {

margin-top: 10px;

background-color: #f0f8ff;

padding: 10px;

border-radius: 5px;

border: 1px solid #ccc;

}

**Exercise 7: Product Catalog with Categories**

**Question**

Create a React application with the following requirements:

Display a list of **product categories**.

When a category is clicked, show a list of **products** for that category.

Allow users to click on a **product** to view its details.

**Components:**

**CategoryList** to display product categories.

**ProductList** to display products within a category.

**ProductDetails** to display details of the selected product.

Include a separate CSS file for each component.

**Solution**

**App.js**

import React, { useState } from "react";

import CategoryList from "./CategoryList";

import ProductList from "./ProductList";

import ProductDetails from "./ProductDetails";

import "./App.css";

function App() {

const [categories] = useState(["Electronics", "Clothing", "Books"]);

const [products] = useState({

Electronics: [

{ id: 1, name: "Laptop", price: "$1200", description: "High-performance laptop" },

{ id: 2, name: "Smartphone", price: "$800", description: "Latest smartphone" },

],

Clothing: [

{ id: 3, name: "T-Shirt", price: "$20", description: "Cotton T-Shirt" },

{ id: 4, name: "Jeans", price: "$50", description: "Denim Jeans" },

],

Books: [

{ id: 5, name: "React Guide", price: "$30", description: "Comprehensive React guide" },

{ id: 6, name: "JavaScript Basics", price: "$25", description: "Learn JavaScript easily" },

],

});

const [selectedCategory, setSelectedCategory] = useState(null);

const [selectedProduct, setSelectedProduct] = useState(null);

return (

<div className="app">

<h1>Product Catalog</h1>

<CategoryList categories={categories} onCategorySelect={setSelectedCategory} />

{selectedCategory && (

<ProductList

products={products[selectedCategory]}

onProductSelect={setSelectedProduct}

/>

)}

{selectedProduct && <ProductDetails product={selectedProduct} />}

</div>

);

}

export default App;

**App.css**

.app {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

**CategoryList.js**

import React from "react";

import "./CategoryList.css";

function CategoryList({ categories, onCategorySelect }) {

return (

<div className="category-list">

<h2>Categories</h2>

{categories.map((category) => (

<button

key={category}

className="category-button"

onClick={() => onCategorySelect(category)}

>

{category}

</button>

))}

</div>

);

}

export default CategoryList;

**CategoryList.css**

.category-list {

margin: 20px 0;

}

.category-button {

margin: 5px;

padding: 10px 15px;

border: none;

background-color: #007bff;

color: white;

border-radius: 5px;

cursor: pointer;

}

.category-button:hover {

background-color: #0056b3;

}

**ProductList.js**

import React from "react";

import "./ProductList.css";

function ProductList({ products, onProductSelect }) {

return (

<div className="product-list">

<h2>Products</h2>

{products.map((product) => (

<div

key={product.id}

className="product-item"

onClick={() => onProductSelect(product)}

>

{product.name} - {product.price}

</div>

))}

</div>

);

}

export default ProductList;

**ProductList.css**

.product-list {

margin: 20px 0;

}

.product-item {

padding: 10px;

border: 1px solid #ccc;

margin: 5px 0;

cursor: pointer;

border-radius: 5px;

}

.product-item:hover {

background-color: #f0f8ff;

}

**ProductDetails.js**

import React from "react";

import "./ProductDetails.css";

function ProductDetails({ product }) {

return (

<div className="product-details">

<h2>Product Details</h2>

<p>

<strong>Name:</strong> {product.name}

</p>

<p>

<strong>Price:</strong> {product.price}

</p>

<p>

<strong>Description:</strong> {product.description}

</p>

</div>

);

}

export default ProductDetails;

**ProductDetails.css**

.product-details {

margin-top: 20px;

padding: 15px;

border: 1px solid #ccc;

border-radius: 5px;

background-color: #f9f9f9;

text-align: left;

display: inline-block;

}

**Exercise 8: Task Management Dashboard**

**Question**

Build a Task Management Dashboard in React with the following functionality:

1. Display multiple task lists (e.g., "To Do," "In Progress," "Completed").
2. Each task list should show tasks under it.
3. Allow the user to add new tasks to a list.
4. Implement functionality to move tasks between lists (drag-and-drop not required; use buttons).
5. Include a detailed view of a selected task, where the user can update its description or delete it.
6. Use at least 10 components:
   * App to manage the global state.
   * TaskList for individual task lists.
   * TaskItem for each task.
   * TaskDetails for viewing/updating a task.
   * AddTaskForm for adding new tasks.
   * Additional components to organize and display the UI as needed.
7. Each component should have its own CSS file.

**Solution**

**App.js**

import React, { useState } from "react";

import TaskList from "./TaskList";

import TaskDetails from "./TaskDetails";

import AddTaskForm from "./AddTaskForm";

import "./App.css";

function App() {

const [tasks, setTasks] = useState({

todo: [

{ id: 1, title: "Learn React", description: "Basics of React" },

{ id: 2, title: "Read a book", description: "Finish reading 'Clean Code'" },

],

inProgress: [

{ id: 3, title: "Build an App", description: "Work on a side project" },

],

completed: [

{ id: 4, title: "Exercise", description: "Morning workout" },

],

});

const [selectedTask, setSelectedTask] = useState(null);

const handleAddTask = (list, task) => {

setTasks((prev) => ({

...prev,

[list]: [...prev[list], task],

}));

};

const handleMoveTask = (fromList, toList, task) => {

setTasks((prev) => ({

...prev,

[fromList]: prev[fromList].filter((t) => t.id !== task.id),

[toList]: [...prev[toList], task],

}));

};

const handleUpdateTask = (list, updatedTask) => {

setTasks((prev) => ({

...prev,

[list]: prev[list].map((task) =>

task.id === updatedTask.id ? updatedTask : task

),

}));

setSelectedTask(null);

};

const handleDeleteTask = (list, taskId) => {

setTasks((prev) => ({

...prev,

[list]: prev[list].filter((task) => task.id !== taskId),

}));

setSelectedTask(null);

};

return (

<div className="app">

<h1>Task Management Dashboard</h1>

<div className="task-lists">

{["todo", "inProgress", "completed"].map((list) => (

<TaskList

key={list}

title={list}

tasks={tasks[list]}

onMoveTask={handleMoveTask}

onSelectTask={(task) => setSelectedTask({ ...task, list })}

onAddTask={(task) => handleAddTask(list, task)}

/>

))}

</div>

{selectedTask && (

<TaskDetails

task={selectedTask}

onUpdateTask={(updatedTask) =>

handleUpdateTask(selectedTask.list, updatedTask)

}

onDeleteTask={() => handleDeleteTask(selectedTask.list, selectedTask.id)}

/>

)}

</div>

);

}

export default App;

**App.css**

.app {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

.task-lists {

display: flex;

justify-content: space-around;

margin-top: 20px;

}

**TaskList.js**

import React from "react";

import TaskItem from "./TaskItem";

import AddTaskForm from "./AddTaskForm";

import "./TaskList.css";

function TaskList({ title, tasks, onMoveTask, onSelectTask, onAddTask }) {

return (

<div className="task-list">

<h2>{title}</h2>

<div>

{tasks.map((task) => (

<TaskItem

key={task.id}

task={task}

onMoveTask={onMoveTask}

onSelectTask={onSelectTask}

currentList={title}

/>

))}

</div>

<AddTaskForm onAddTask={onAddTask} />

</div>

);

}

export default TaskList;

**TaskList.css**

.task-list {

width: 30%;

padding: 15px;

border: 1px solid #ccc;

border-radius: 5px;

background-color: #f9f9f9;

}

**TaskItem.js**

import React from "react";

import "./TaskItem.css";

function TaskItem({ task, onMoveTask, onSelectTask, currentList }) {

const otherLists = {

todo: ["inProgress", "completed"],

inProgress: ["todo", "completed"],

completed: ["todo", "inProgress"],

}[currentList];

return (

<div className="task-item">

<h4 onClick={() => onSelectTask(task)}>{task.title}</h4>

<div>

{otherLists.map((list) => (

<button key={list} onClick={() => onMoveTask(currentList, list, task)}>

Move to {list}

</button>

))}

</div>

</div>

);

}

export default TaskItem;

**TaskItem.css**

.task-item {

padding: 10px;

margin: 10px 0;

border: 1px solid #ddd;

border-radius: 5px;

background-color: #fff;

cursor: pointer;

}

.task-item h4:hover {

text-decoration: underline;

}

.task-item button {

margin: 5px;

padding: 5px 10px;

border: none;

background-color: #007bff;

color: white;

border-radius: 5px;

cursor: pointer;

}

.task-item button:hover {

background-color: #0056b3;

}

**Additional Components:**

Include AddTaskForm.js and TaskDetails.js, along with their corresponding CSS files. Let me know if you'd like to see these components in detail!

Here are the additional components: AddTaskForm.js and TaskDetails.js, along with their respective CSS files.

**AddTaskForm.js**

import React, { useState } from "react";

import "./AddTaskForm.css";

function AddTaskForm({ onAddTask }) {

const [taskTitle, setTaskTitle] = useState("");

const [taskDescription, setTaskDescription] = useState("");

const handleAdd = (e) => {

e.preventDefault();

if (!taskTitle.trim() || !taskDescription.trim()) return;

const newTask = {

id: Date.now(),

title: taskTitle,

description: taskDescription,

};

onAddTask(newTask);

setTaskTitle("");

setTaskDescription("");

};

return (

<form className="add-task-form" onSubmit={handleAdd}>

<input

type="text"

placeholder="Task Title"

value={taskTitle}

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

/>

<textarea

placeholder="Task Description"

value={taskDescription}

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

/>

<button type="submit">Add Task</button>

</form>

);

}

export default AddTaskForm;

**AddTaskForm.css**

.add-task-form {

margin-top: 10px;

display: flex;

flex-direction: column;

gap: 10px;

}

.add-task-form input,

.add-task-form textarea {

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

.add-task-form button {

padding: 10px;

background-color: #28a745;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.add-task-form button:hover {

background-color: #218838;

}

**TaskDetails.js**

import React, { useState } from "react";

import "./TaskDetails.css";

function TaskDetails({ task, onUpdateTask, onDeleteTask }) {

const [isEditing, setIsEditing] = useState(false);

const [updatedTitle, setUpdatedTitle] = useState(task.title);

const [updatedDescription, setUpdatedDescription] = useState(task.description);

const handleSave = () => {

onUpdateTask({

...task,

title: updatedTitle,

description: updatedDescription,

});

setIsEditing(false);

};

return (

<div className="task-details">

<h2>Task Details</h2>

{isEditing ? (

<>

<input

type="text"

value={updatedTitle}

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

/>

<textarea

value={updatedDescription}

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

/>

<button onClick={handleSave}>Save</button>

</>

) : (

<>

<h3>{task.title}</h3>

<p>{task.description}</p>

<button onClick={() => setIsEditing(true)}>Edit</button>

</>

)}

<button onClick={onDeleteTask} className="delete-btn">

Delete

</button>

</div>

);

}

export default TaskDetails;

**TaskDetails.css**

.task-details {

margin-top: 20px;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

background-color: #f8f9fa;

text-align: left;

}

.task-details h2 {

margin-bottom: 15px;

}

.task-details h3 {

margin: 10px 0;

}

.task-details textarea,

.task-details input {

width: 100%;

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ddd;

border-radius: 5px;

}

.task-details button {

margin-right: 10px;

padding: 10px 15px;

border: none;

border-radius: 5px;

cursor: pointer;

}

.task-details button:hover {

opacity: 0.9;

}

.task-details .delete-btn {

background-color: #dc3545;

color: white;

}

.task-details .delete-btn:hover {

background-color: #c82333;

}

Now your **Task Management Dashboard** includes the AddTaskForm for adding tasks and TaskDetails for viewing, editing, and deleting tasks. Let me know if you need more features or further refinements!

**Exercise 9: Project Management Dashboard**

Build a React application that simulates a **Project Management Dashboard**. The application should include the following features:

1. A list of projects with details such as name, description, and assigned team members.
2. A form to add new projects.
3. Task management within each project, including adding, editing, and deleting tasks.
4. Display project details with a list of associated tasks and team members.
5. Assign team members to projects from a predefined list.
6. Use at least **10 components**, each with a dedicated CSS file.

**Directory Structure**

src/

├── components/

│ ├── AddProjectForm.js

│ ├── AddTaskForm.js

│ ├── AssignTeamMembers.js

│ ├── Navbar.js

│ ├── ProjectCard.js

│ ├── ProjectDetails.js

│ ├── ProjectList.js

│ ├── TaskDetails.js

│ ├── TeamMemberCard.js

│ ├── TeamMemberList.js

├── css/

│ ├── AddProjectForm.css

│ ├── AddTaskForm.css

│ ├── AssignTeamMembers.css

│ ├── Navbar.css

│ ├── ProjectCard.css

│ ├── ProjectDetails.css

│ ├── ProjectList.css

│ ├── TaskDetails.css

│ ├── TeamMemberCard.css

│ ├── TeamMemberList.css

├── App.js

├── App.css

**Solution:**

**1. AddProjectForm.js**

import React, { useState } from "react";

import "../css/AddProjectForm.css";

function AddProjectForm({ onAddProject }) {

const [projectName, setProjectName] = useState("");

const [description, setDescription] = useState("");

const handleSubmit = (e) => {

e.preventDefault();

const newProject = {

id: Date.now(),

name: projectName,

description,

tasks: [],

teamMembers: [],

};

onAddProject(newProject);

setProjectName("");

setDescription("");

};

return (

<form className="add-project-form" onSubmit={handleSubmit}>

<input

type="text"

placeholder="Project Name"

value={projectName}

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

/>

<textarea

placeholder="Description"

value={description}

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

/>

<button type="submit">Add Project</button>

</form>

);

}

export default AddProjectForm;

**AddProjectForm.css**

.add-project-form {

display: flex;

flex-direction: column;

gap: 10px;

}

.add-project-form input,

.add-project-form textarea {

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

}

.add-project-form button {

padding: 10px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

**2. ProjectList.js**

import React from "react";

import ProjectCard from "./ProjectCard";

import "../css/ProjectList.css";

function ProjectList({ projects, teamMembers, onUpdateProject }) {

return (

<div className="project-list">

{projects.map((project) => (

<ProjectCard

key={project.id}

project={project}

teamMembers={teamMembers}

onUpdateProject={onUpdateProject}

/>

))}

</div>

);

}

export default ProjectList;

**ProjectList.css**

.project-list {

display: flex;

flex-wrap: wrap;

gap: 20px;

}

**3. ProjectCard.js**

import React from "react";

import "../css/ProjectCard.css";

function ProjectCard({ project, onUpdateProject }) {

return (

<div className="project-card">

<h3>{project.name}</h3>

<p>{project.description}</p>

<button onClick={() => onUpdateProject(project)}>View Details</button>

</div>

);

}

export default ProjectCard;

**ProjectCard.css**

.project-card {

border: 1px solid #ddd;

border-radius: 5px;

padding: 10px;

width: 200px;

background-color: white;

}

.project-card h3 {

margin: 0 0 10px;

}

.project-card p {

color: #555;

}

.project-card button {

background-color: #007bff;

color: white;

border: none;

padding: 8px;

border-radius: 5px;

cursor: pointer;

}

**4. AddTaskForm.js**

import React, { useState } from "react";

import "../css/AddTaskForm.css";

function AddTaskForm({ onAddTask }) {

const [taskName, setTaskName] = useState("");

const handleSubmit = (e) => {

e.preventDefault();

const newTask = { id: Date.now(), name: taskName };

onAddTask(newTask);

setTaskName("");

};

return (

<form className="add-task-form" onSubmit={handleSubmit}>

<input

type="text"

placeholder="Task Name"

value={taskName}

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

/>

<button type="submit">Add Task</button>

</form>

);

}

export default AddTaskForm;

**AddTaskForm.css**

.add-task-form {

display: flex;

gap: 10px;

}

.add-task-form input {

padding: 8px;

border: 1px solid #ddd;

border-radius: 5px;

}

.add-task-form button {

background-color: #28a745;

color: white;

border: none;

padding: 8px;

border-radius: 5px;

cursor: pointer;

}

**5. TaskDetails.js**

import React from "react";

import "../css/TaskDetails.css";

function TaskDetails({ task, onDeleteTask }) {

return (

<div className="task-details">

<span>{task.name}</span>

<button onClick={() => onDeleteTask(task.id)}>Delete</button>

</div>

);

}

export default TaskDetails;

**TaskDetails.css**

.task-details {

display: flex;

justify-content: space-between;

padding: 10px;

border-bottom: 1px solid #ddd;

}

.task-details button {

background-color: #dc3545;

color: white;

border: none;

padding: 5px;

border-radius: 5px;

cursor: pointer;

}

**6. Navbar.js**

import React from "react";

import "../css/Navbar.css";

function Navbar() {

return (

<nav className="navbar">

<h1>Project Management Dashboard</h1>

<ul>

<li>Home</li>

<li>Projects</li>

<li>Team</li>

</ul>

</nav>

);

}

export default Navbar;

**Navbar.css**

.navbar {

background-color: #007bff;

color: white;

padding: 15px;

display: flex;

justify-content: space-between;

align-items: center;

}

.navbar h1 {

margin: 0;

}

.navbar ul {

list-style: none;

display: flex;

gap: 20px;

margin: 0;

padding: 0;

}

.navbar ul li {

cursor: pointer;

font-size: 18px;

}

**7. ProjectDetails.js**

import React from "react";

import TaskDetails from "./TaskDetails";

import AddTaskForm from "./AddTaskForm";

import "../css/ProjectDetails.css";

function ProjectDetails({ project, onAddTask, onDeleteTask }) {

return (

<div className="project-details">

<h2>{project.name}</h2>

<p>{project.description}</p>

<h3>Tasks</h3>

<AddTaskForm onAddTask={(task) => onAddTask(project.id, task)} />

<div className="task-list">

{project.tasks.map((task) => (

<TaskDetails

key={task.id}

task={task}

onDeleteTask={(taskId) => onDeleteTask(project.id, taskId)}

/>

))}

</div>

</div>

);

}

export default ProjectDetails;

**ProjectDetails.css**

.project-details {

padding: 20px;

background-color: #f8f9fa;

border-radius: 5px;

}

.project-details h2 {

margin: 0 0 10px;

}

.project-details h3 {

margin: 20px 0 10px;

}

.task-list {

margin-top: 10px;

}

**8. AssignTeamMembers.js**

import React, { useState } from "react";

import "../css/AssignTeamMembers.css";

function AssignTeamMembers({ project, teamMembers, onAssignMember }) {

const [selectedMember, setSelectedMember] = useState("");

const handleAssign = () => {

if (selectedMember) {

onAssignMember(project.id, selectedMember);

setSelectedMember("");

}

};

return (

<div className="assign-team-members">

<h3>Assign Team Members</h3>

<select

value={selectedMember}

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

>

<option value="">Select Member</option>

{teamMembers.map((member) => (

<option key={member.id} value={member.id}>

{member.name}

</option>

))}

</select>

<button onClick={handleAssign}>Assign</button>

</div>

);

}

export default AssignTeamMembers;

**AssignTeamMembers.css**

.assign-team-members {

margin-top: 20px;

}

.assign-team-members select {

padding: 8px;

border: 1px solid #ddd;

border-radius: 5px;

margin-right: 10px;

}

.assign-team-members button {

background-color: #28a745;

color: white;

border: none;

padding: 8px 12px;

border-radius: 5px;

cursor: pointer;

}

**9. TeamMemberCard.js**

import React from "react";

import "../css/TeamMemberCard.css";

function TeamMemberCard({ member }) {

return (

<div className="team-member-card">

<h4>{member.name}</h4>

<p>Role: {member.role}</p>

</div>

);

}

export default TeamMemberCard;

**TeamMemberCard.css**

.team-member-card {

border: 1px solid #ddd;

padding: 10px;

border-radius: 5px;

background-color: #f9f9f9;

}

.team-member-card h4 {

margin: 0 0 5px;

}

.team-member-card p {

margin: 0;

color: #555;

}

**10. TeamMemberList.js**

import React from "react";

import TeamMemberCard from "./TeamMemberCard";

import "../css/TeamMemberList.css";

function TeamMemberList({ members }) {

return (

<div className="team-member-list">

{members.map((member) => (

<TeamMemberCard key={member.id} member={member} />

))}

</div>

);

}

export default TeamMemberList;

**TeamMemberList.css**

.team-member-list {

display: flex;

gap: 15px;

flex-wrap: wrap;

}

**App.js**

import React, { useState } from "react";

import Navbar from "./components/Navbar";

import ProjectList from "./components/ProjectList";

import AddProjectForm from "./components/AddProjectForm";

import "./App.css";

function App() {

const [projects, setProjects] = useState([]);

const [teamMembers] = useState([

{ id: 1, name: "Alice", role: "Developer" },

{ id: 2, name: "Bob", role: "Designer" },

{ id: 3, name: "Charlie", role: "Tester" },

]);

const addProject = (project) => setProjects([...projects, project]);

const updateProject = (updatedProject) => {

setProjects((prev) =>

prev.map((project) =>

project.id === updatedProject.id ? updatedProject : project

)

);

};

return (

<div className="app">

<Navbar />

<main>

<AddProjectForm onAddProject={addProject} />

<ProjectList

projects={projects}

teamMembers={teamMembers}

onUpdateProject={updateProject}

/>

</main>

</div>

);

}

export default App;

**App.css**

.app {

font-family: Arial, sans-serif;

padding: 20px;

background-color: #f2f2f2;

}

main {

margin-top: 20px;

}

**Exercise 10: Human Resource Management System (HRMS)**

Build a React application that simulates a **Human Resource Management System (HRMS)**. The application should include the following features:

**HRMS App Example**

This HRMS app will include the following components:

* **Navbar**: For navigation between different sections.
* **EmployeeList**: Displays a list of employees.
* **AddEmployeeForm**: A form to add new employees.
* **EmployeeDetails**: Displays the details of an employee.
* **DepartmentList**: Displays a list of departments.
* **AddDepartmentForm**: A form to add new departments.
* **AttendanceList**: Displays employee attendance.
* **AddAttendanceForm**: A form to record attendance.
* **EmployeeCard**: A card for each employee in the list.
* **DepartmentCard**: A card for each department in the list.
* **HRDashboard**: Main dashboard showing an overview of employees, departments, and attendance.

Here’s a more complete HRMS example with **10 components**, each following the **Lists and Keys** concept in React. Along with the components, I'll include their corresponding **CSS files** for styling. The structure is as follows:

**Project Structure:**

hrms-app/

├── public/

│ ├── index.html

│ └── favicon.ico

├── src/

│ ├── components/

│ │ ├── EmployeeList.js

│ │ ├── DepartmentList.js

│ │ ├── AttendanceList.js

│ │ ├── SalaryList.js

│ │ ├── TaskList.js

│ │ ├── LeaveList.js

│ │ ├── ProjectList.js

│ │ ├── TrainingList.js

│ │ ├── DepartmentHeads.js

│ │ ├── JobTitleList.js

│ ├── css/

│ │ ├── EmployeeList.css

│ │ ├── DepartmentList.css

│ │ ├── AttendanceList.css

│ │ ├── SalaryList.css

│ │ ├── TaskList.css

│ │ ├── LeaveList.css

│ │ ├── ProjectList.css

│ │ ├── TrainingList.css

│ │ ├── DepartmentHeads.css

│ │ ├── JobTitleList.css

│ ├── App.js

│ ├── index.js

│ ├── index.css

└── package.json

**1. EmployeeList.js**

// src/components/EmployeeList.js

import React from 'react';

import './EmployeeList.css';

const EmployeeList = ({ employees }) => {

return (

<div className="employee-list">

<h2>Employee List</h2>

<ul>

{employees.map(employee => (

<li key={employee.id} className="employee-item">

{employee.name} - {employee.position}

</li>

))}

</ul>

</div>

);

};

export default EmployeeList;

**EmployeeList.css**

/\* src/css/EmployeeList.css \*/

.employee-list {

padding: 20px;

background-color: #f4f4f4;

}

.employee-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.employee-item:hover {

background-color: #e0e0e0;

}

**2. DepartmentList.js**

// src/components/DepartmentList.js

import React from 'react';

import './DepartmentList.css';

const DepartmentList = ({ departments }) => {

return (

<div className="department-list">

<h2>Department List</h2>

<ul>

{departments.map(department => (

<li key={department.id} className="department-item">

{department.name}

</li>

))}

</ul>

</div>

);

};

export default DepartmentList;

**DepartmentList.css**

/\* src/css/DepartmentList.css \*/

.department-list {

padding: 20px;

background-color: #fafafa;

}

.department-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.department-item:hover {

background-color: #e9e9e9;

}

**3. AttendanceList.js**

// src/components/AttendanceList.js

import React from 'react';

import './AttendanceList.css';

const AttendanceList = ({ attendanceRecords }) => {

return (

<div className="attendance-list">

<h2>Attendance Records</h2>

<ul>

{attendanceRecords.map(record => (

<li key={record.id} className="attendance-item">

{record.employeeName} - {record.date} - {record.status}

</li>

))}

</ul>

</div>

);

};

export default AttendanceList;

**AttendanceList.css**

/\* src/css/AttendanceList.css \*/

.attendance-list {

padding: 20px;

background-color: #eef3f7;

}

.attendance-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.attendance-item:hover {

background-color: #f0f0f0;

}

**4. SalaryList.js**

// src/components/SalaryList.js

import React from 'react';

import './SalaryList.css';

const SalaryList = ({ salaries }) => {

return (

<div className="salary-list">

<h2>Salary List</h2>

<ul>

{salaries.map(salary => (

<li key={salary.id} className="salary-item">

{salary.employeeName} - ${salary.amount}

</li>

))}

</ul>

</div>

);

};

export default SalaryList;

**SalaryList.css**

/\* src/css/SalaryList.css \*/

.salary-list {

padding: 20px;

background-color: #f2f2f2;

}

.salary-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.salary-item:hover {

background-color: #e9e9e9;

}

**5. TaskList.js**

// src/components/TaskList.js

import React from 'react';

import './TaskList.css';

const TaskList = ({ tasks }) => {

return (

<div className="task-list">

<h2>Task List</h2>

<ul>

{tasks.map(task => (

<li key={task.id} className="task-item">

{task.title} - {task.status}

</li>

))}

</ul>

</div>

);

};

export default TaskList;

**TaskList.css**

/\* src/css/TaskList.css \*/

.task-list {

padding: 20px;

background-color: #f9f9f9;

}

.task-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.task-item:hover {

background-color: #f0f0f0;

}

**6. LeaveList.js**

// src/components/LeaveList.js

import React from 'react';

import './LeaveList.css';

const LeaveList = ({ leaves }) => {

return (

<div className="leave-list">

<h2>Leave List</h2>

<ul>

{leaves.map(leave => (

<li key={leave.id} className="leave-item">

{leave.employeeName} - {leave.date} - {leave.type}

</li>

))}

</ul>

</div>

);

};

export default LeaveList;

**LeaveList.css**

/\* src/css/LeaveList.css \*/

.leave-list {

padding: 20px;

background-color: #f6f6f6;

}

.leave-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.leave-item:hover {

background-color: #e8e8e8;

}

**7. ProjectList.js**

// src/components/ProjectList.js

import React from 'react';

import './ProjectList.css';

const ProjectList = ({ projects }) => {

return (

<div className="project-list">

<h2>Project List</h2>

<ul>

{projects.map(project => (

<li key={project.id} className="project-item">

{project.name} - {project.status}

</li>

))}

</ul>

</div>

);

};

export default ProjectList;

**ProjectList.css**

/\* src/css/ProjectList.css \*/

.project-list {

padding: 20px;

background-color: #ececec;

}

.project-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.project-item:hover {

background-color: #dfdfdf;

}

**8. TrainingList.js**

// src/components/TrainingList.js

import React from 'react';

import './TrainingList.css';

const TrainingList = ({ trainings }) => {

return (

<div className="training-list">

<h2>Training List</h2>

<ul>

{trainings.map(training => (

<li key={training.id} className="training-item">

{training.name} - {training.date}

</li>

))}

</ul>

</div>

);

};

export default TrainingList;

**TrainingList.css**

/\* src/css/TrainingList.css \*/

.training-list {

padding: 20px;

background-color: #d7d7d7;

}

.training-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.training-item:hover {

background-color: #cccccc;

}

**9. DepartmentHeads.js**

// src/components/DepartmentHeads.js

import React from 'react';

import './DepartmentHeads.css';

const DepartmentHeads = ({ departmentHeads }) => {

return (

<div className="department-heads">

<h2>Department Heads</h2>

<ul>

{departmentHeads.map(head => (

<li key={head.id} className="head-item">

{head.name} - {head.department}

</li>

))}

</ul>

</div>

);

};

export default DepartmentHeads;

**DepartmentHeads.css**

/\* src/css/DepartmentHeads.css \*/

.department-heads {

padding: 20px;

background-color: #d1e7f5;

}

.head-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.head-item:hover {

background-color: #c6d8e2;

}

**10. JobTitleList.js**

// src/components/JobTitleList.js

import React from 'react';

import './JobTitleList.css';

const JobTitleList = ({ jobTitles }) => {

return (

<div className="job-title-list">

<h2>Job Title List</h2>

<ul>

{jobTitles.map(title => (

<li key={title.id} className="job-title-item">

{title.name}

</li>

))}

</ul>

</div>

);

};

export default JobTitleList;

**JobTitleList.css**

/\* src/css/JobTitleList.css \*/

.job-title-list {

padding: 20px;

background-color: #f0f0f0;

}

.job-title-item {

background-color: #fff;

padding: 10px;

margin: 5px 0;

border-radius: 5px;

border: 1px solid #ddd;

}

.job-title-item:hover {

background-color: #e0e0e0;

}

**App.js**

// src/App.js

import React from "react";

import EmployeeList from "./components/EmployeeList";

import DepartmentList from "./components/DepartmentList";

import AttendanceList from "./components/AttendanceList";

import SalaryList from "./components/SalaryList";

import TaskList from "./components/TaskList";

import LeaveList from "./components/LeaveList";

import ProjectList from "./components/ProjectList";

import TrainingList from "./components/TrainingList";

import DepartmentHeads from "./components/DepartmentHeads";

import JobTitleList from "./components/JobTitleList";

const employees = [

{ id: 1, name: "John Doe", position: "Software Engineer" },

{ id: 2, name: "Jane Smith", position: "Product Manager" },

{ id: 3, name: "Bob Johnson", position: "UX Designer" },

];

const departments = [

{ id: 1, name: "Engineering" },

{ id: 2, name: "Marketing" },

{ id: 3, name: "Sales" },

];

// Further mock data for other lists

const App = () => {

return (

<div className="app">

<EmployeeList employees={employees} />

<DepartmentList departments={departments} />

<AttendanceList attendanceRecords={[]} />

<SalaryList salaries={[]} />

<TaskList tasks={[]} />

<LeaveList leaves={[]} />

<ProjectList projects={[]} />

<TrainingList trainings={[]} />

<DepartmentHeads departmentHeads={[]} />

<JobTitleList jobTitles={[]} />

</div>

);

};

export default App;