Kết quả lab 6:

**📌 So sánh 3 bài khi dùng Redux vs không dùng Redux**

**1. Cart App (Giỏ hàng)**

🔹 **Không Redux**

* State cartItems nằm trong App.js.
* Khi ProductList muốn thêm sản phẩm vào giỏ → phải truyền hàm addToCart từ App.js xuống qua props.
* Nếu có nhiều cấp component trung gian → **prop drilling** (truyền props lòng vòng, dễ rối).

🔹 **Dùng Redux**

* cartSlice quản lý state giỏ hàng toàn cục.
* ProductList chỉ cần dispatch(addToCart(product)), không cần props.
* Cart chỉ cần useSelector lấy state.
* Dù ProductList và Cart không liên quan trực tiếp → vẫn dùng chung state mượt mà.

👉 **Lợi ích**: Không cần props qua nhiều tầng, dễ mở rộng khi app to hơn.

**2. Login App (Đăng nhập)**

🔹 **Không Redux**

* State user nằm trong App.js.
* LoginForm khi login thành công phải gọi hàm setUser của App.js.
* UserInfo cũng phải nhận user từ App.js qua props.
* Nếu nhiều component khác nhau (Navbar, Profile, Settings) cần thông tin user → truyền props rất phức tạp.

🔹 **Dùng Redux**

* authSlice lưu state { user, isLoggedIn } toàn cục.
* LoginForm chỉ cần dispatch(login(user)).
* UserInfo, Navbar, Profile chỉ cần useSelector lấy user.
* Không phụ thuộc vào vị trí component trong cây.

👉 **Lợi ích**: Quản lý session/login nhất quán, dễ dàng kiểm soát toàn bộ app.

**3. Filter App (Tìm kiếm & lọc sản phẩm)**

🔹 **Không Redux**

* State filters (search, category, sort) nằm trong App.js.
* FilterBar phải nhận và thay đổi state từ App.js.
* ProductList cũng phải nhận products đã lọc từ App.js.
* Nếu muốn thêm Pagination, Wishlist, v.v. → App.js phình to.

🔹 **Dùng Redux**

* productsSlice lưu state: { items, filters }.
* FilterBar chỉ cần dispatch(setFilter("brand", "Nike")).
* ProductList tự động re-render khi filters thay đổi vì lấy state từ Redux.
* Dễ thêm tính năng (pagination, sort, favorite) mà không cần sửa App.js.

👉 **Lợi ích**: Tách biệt logic filter ra Redux → App.js nhẹ hơn, component rõ ràng hơn.

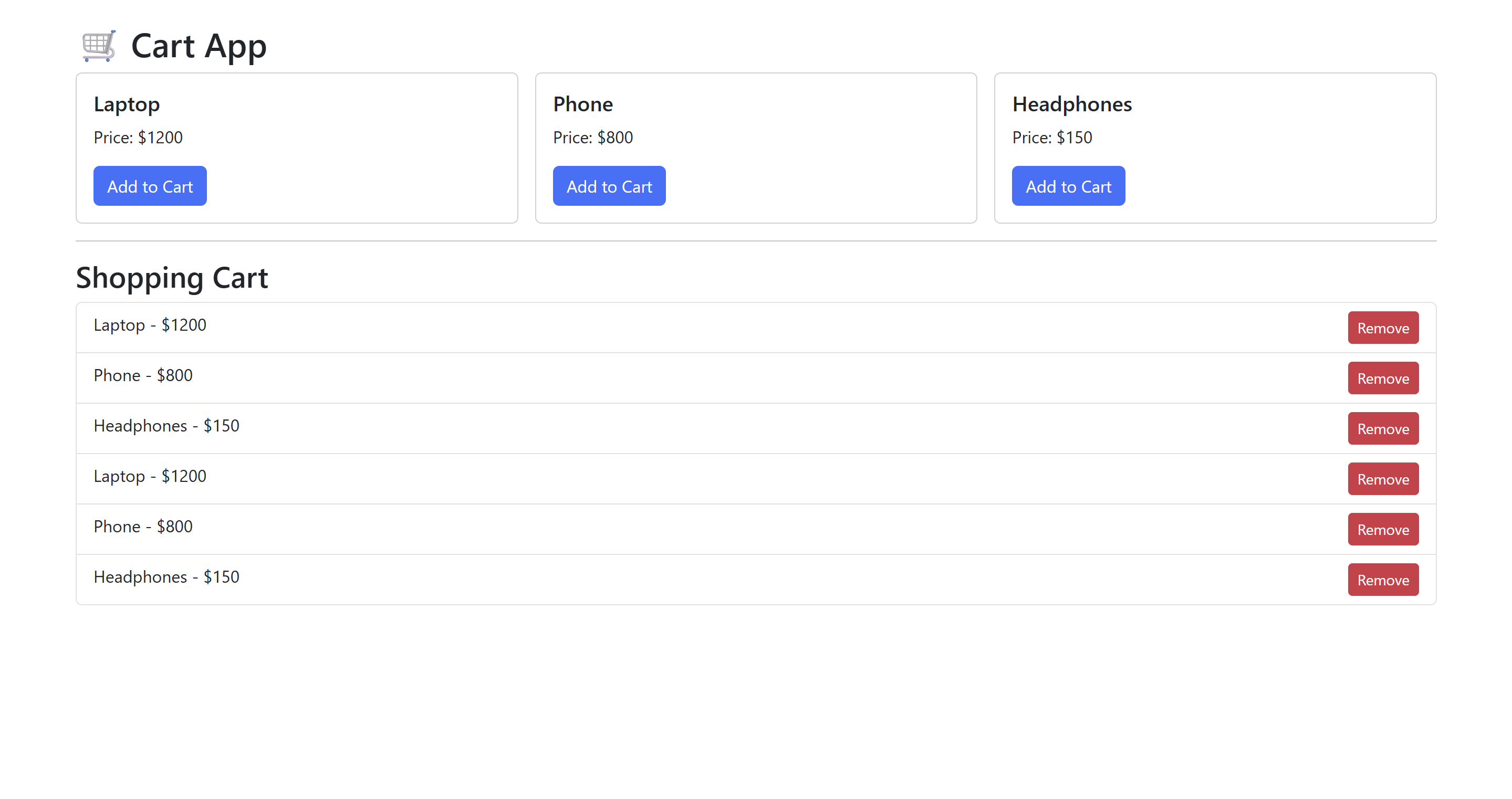
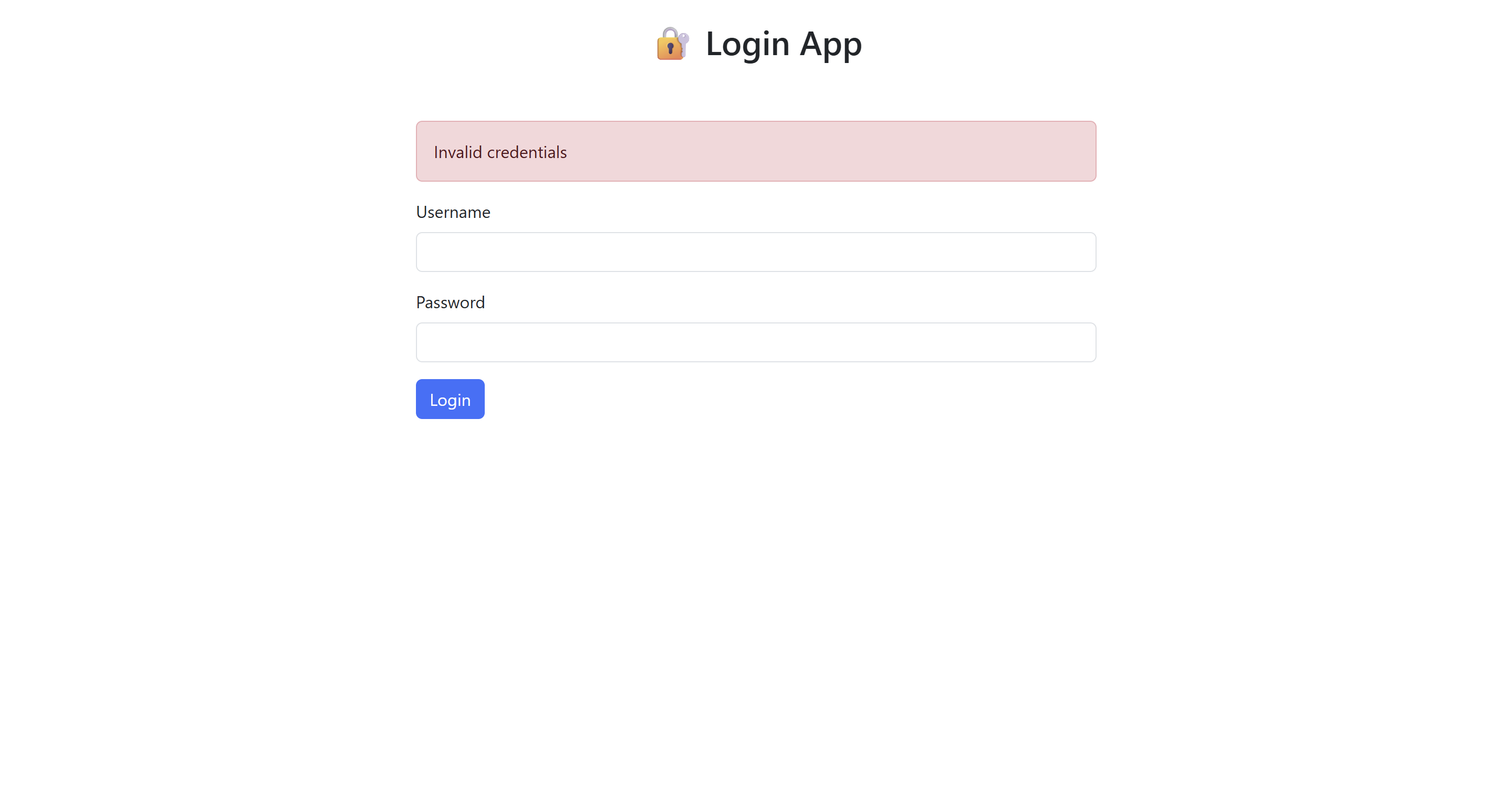
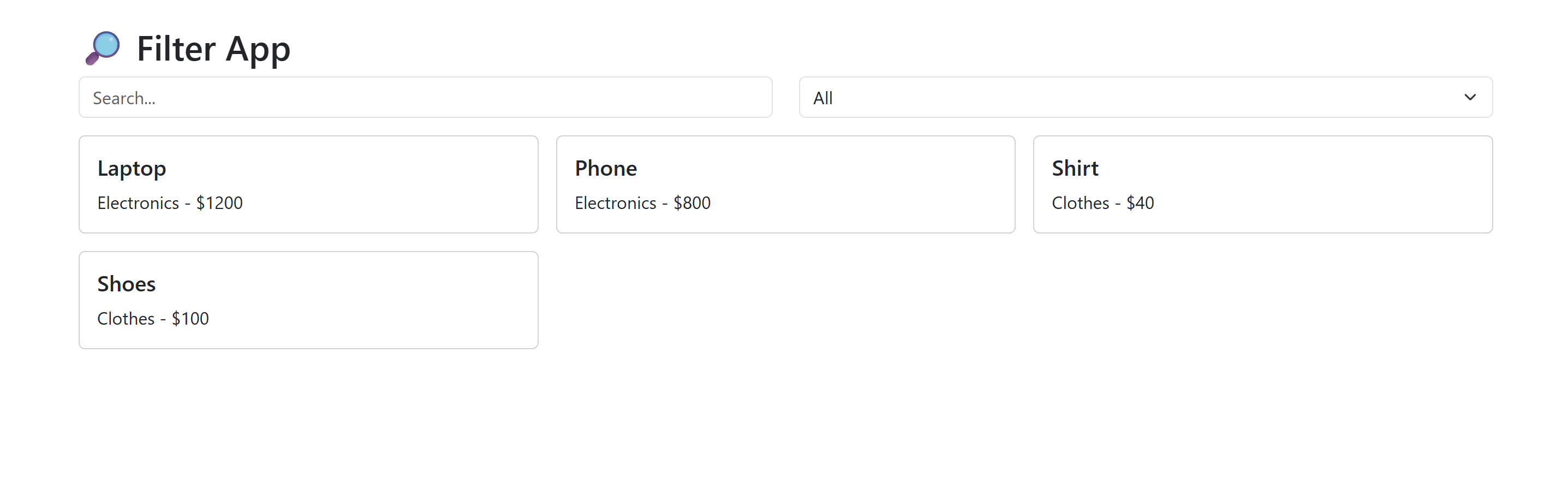
**🎯 Kết luận**

**Khi app nhỏ** (ít component, ít state) → không Redux thì code đơn giản, dễ viết.  
**Khi app lớn** (nhiều component, nhiều state chia sẻ) → Redux cực kỳ hữu ích vì:

1. Tránh **prop drilling** (truyền props vòng vòng).
2. Giúp **state management rõ ràng** (dùng action + reducer).
3. **Dễ scale** khi app phức tạp (giỏ hàng, login, filter, profile, settings...).
4. Dễ debug (Redux DevTools cho bạn thấy toàn bộ lịch sử state).

👉 Tóm lại:

* **Cart App**: Redux giúp đồng bộ giỏ hàng ở mọi chỗ.
* **Login App**: Redux giúp quản lý trạng thái đăng nhập toàn cục.
* **Filter App**: Redux giúp tách logic lọc/sort ra khỏi App, dễ mở rộng.

1. Kết quả:  
   - Cart App: 
   * Login App:  
     
   * Filter App:  
     
   * Tổng hợp 3 bài vào 1 dự án:  
     