**Báo cáo thực tập viết niên luận**

**Đề tài: Xây dựng ứng dụng Todo-List bằng React**

Sinh viên thực hiện: Nguyễn Phước Thành  
Mã sinh viên: 22T1020438  
Lớp: K46F  
Giảng viên hướng dẫn: Trần Nguyên Phong

## **1. Giới thiệu đề tài**

### **1.1. Lý do chọn đề tài**

Ứng dụng To-Do List là một trong những ứng dụng cơ bản nhưng giúp sinh viên làm quen với **ReactJS** – thư viện JavaScript phổ biến để xây dựng giao diện người dùng.  
Việc thực hiện ứng dụng này không chỉ giúp nắm vững **các khái niệm về component, props, state, hook** trong React mà còn biết cách **lưu dữ liệu cục bộ (Local Storage)**, cải thiện giao diện với **Tailwind CSS** và nâng cao trải nghiệm người dùng với **React-Toastify**.

### **1.2. Mục tiêu**

* Xây dựng ứng dụng quản lý công việc (CRUD).
* Lưu trữ dữ liệu trên Local Storage.
* Cải thiện giao diện và UX với Tailwind CSS.
* Thêm thông báo tương tác khi người dùng thao tác.

## **2. Yêu cầu chức năng**

### **2.1. Tính năng bắt buộc**

* **Thêm công việc** mới.
* **Xóa công việc**.
* **Đánh dấu hoàn thành / chưa hoàn thành**.
* **Lọc công việc** theo trạng thái (Tất cả / Đang làm / Hoàn thành) – nếu có thời gian.
* **Lưu trữ dữ liệu** bằng Local Storage.

### **2.2. Tính năng nâng cao**

* Giao diện đẹp mắt, responsive.
* Thông báo toast khi thêm hoặc xóa công việc.
* Hiệu ứng hover, active, bóng đổ.

## **3. Công nghệ sử dụng**

* **ReactJS** (Create React App).
* **Tailwind CSS** (thư viện CSS utility-first).
* **React-Toastify** (thông báo).
* **Local Storage API** (lưu dữ liệu cục bộ).
* **JavaScript ES6+**.

## **4. Các bước thực hiện theo tuần**

### **Tuần 1: Cài đặt môi trường**

* Cài đặt Node.js, npm.
* Cài đặt VS Code và tiện ích hỗ trợ.
* Khởi tạo project React với create-react-app.

### **Tuần 2: Phân tích yêu cầu & thiết kế giao diện**

* Xác định chức năng CRUD và các component: App, TodoForm, TodoList, TodoItem, Filter.
* Vẽ phác thảo UI bằng draw.io.
* Lập sơ đồ luồng dữ liệu giữa các component.

### **Tuần 3: Xây dựng các component tĩnh**

* Tạo các component React chứa markup cơ bản.
* Hiển thị danh sách mẫu và form nhập công việc.

### **Tuần 4: Thêm logic với useState**

* Quản lý danh sách công việc bằng state.
* Thêm chức năng thêm, xóa, đánh dấu hoàn thành.

### **Tuần 5: Lưu dữ liệu với Local Storage**

* Load dữ liệu từ Local Storage khi khởi động app.
* Lưu dữ liệu vào Local Storage khi danh sách thay đổi.
* Đảm bảo công việc không mất khi reload trang.

### **Tuần 6: Cải thiện giao diện & UX**

* Tích hợp Tailwind CSS.
* Thêm hiệu ứng hover, active, màu sắc dễ nhìn.
* Bổ sung thông báo với React-Toastify khi thêm hoặc xóa công việc.

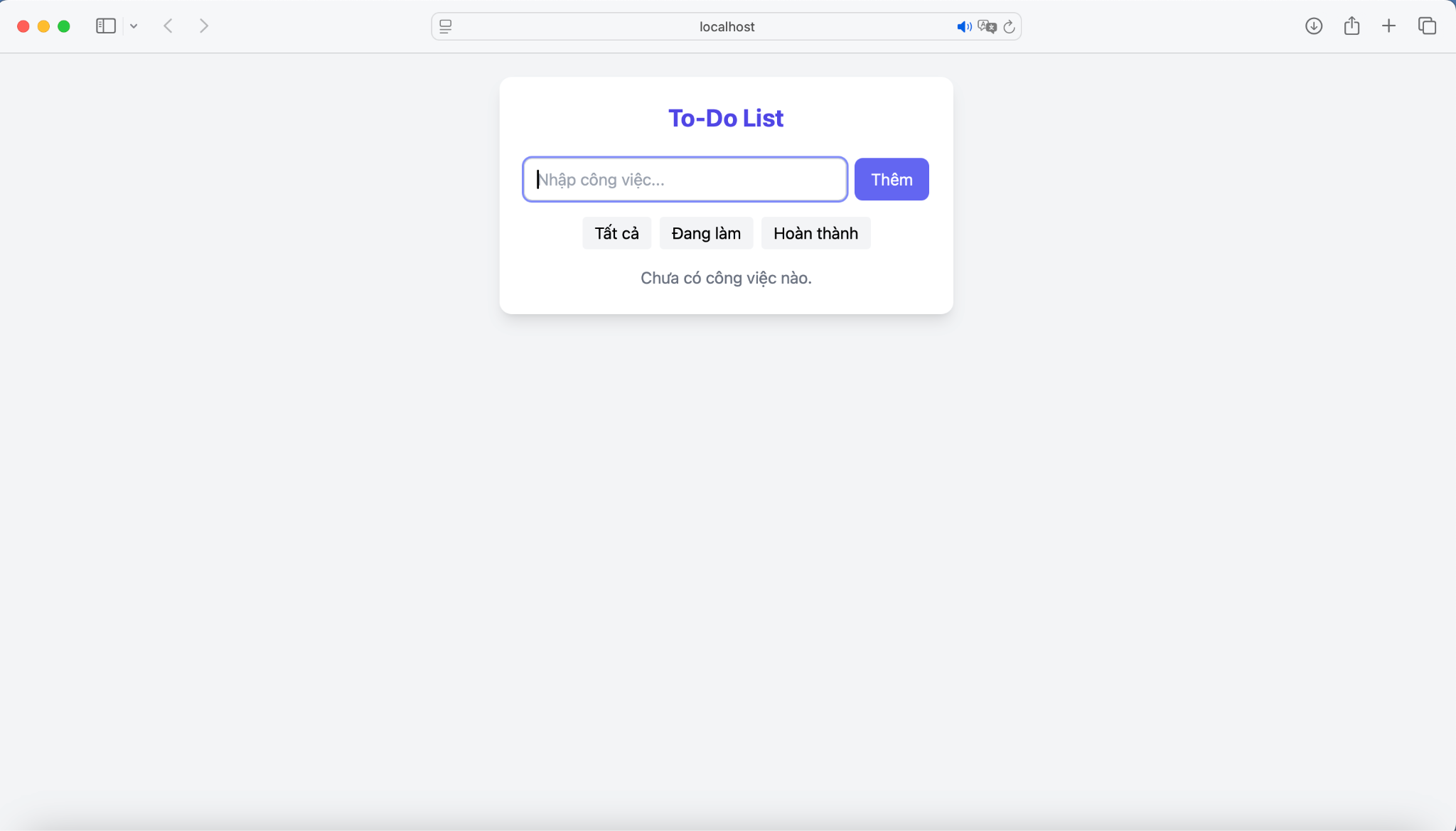
### **Tuần 7: Hoàn thiện & kiểm thử**

* Hoàn thiện chức năng lọc công việc (nếu có).
* Test trên nhiều trình duyệt.
* Viết báo cáo tổng kết.

## **5. Kết quả đạt được**

* Ứng dụng To-Do List chạy ổn định, đáp ứng yêu cầu CRUD.
* Dữ liệu lưu trữ an toàn bằng Local Storage.
* Giao diện thân thiện, hiện đại, có hiệu ứng tương tác và thông báo rõ ràng.
* Mã nguồn rõ ràng, tách biệt component.

## **6. Giao diện minh họa**

**

## **7. Đánh giá và hướng phát triển**

### **7.1. Đánh giá**

* **Ưu điểm:**
  + Đáp ứng đầy đủ chức năng cơ bản và một số nâng cao.
  + Áp dụng tốt React Hooks (useState, useEffect).
  + UI hiện đại nhờ Tailwind CSS.
* **Nhược điểm:**
  + Chưa có chức năng sửa công việc.
  + Bộ lọc công việc mới ở mức cơ bản.

### **7.2. Hướng phát triển**

* Thêm tính năng **Edit** công việc.
* Đồng bộ dữ liệu với server qua API.
* Đăng nhập và lưu công việc theo tài khoản.

## **8. Kết luận**

Qua quá trình thực hiện, em đã nắm được cách sử dụng React để xây dựng ứng dụng thực tế, hiểu rõ hơn về component, state, props, hook, cũng như tích hợp thư viện CSS và thông báo.  
Đề tài giúp rèn luyện kỹ năng lập trình front-end, tư duy thiết kế UI/UX và quản lý dữ liệu cục bộ.