

Báo cáo Thực tập Doanh nghiệp

Đề tài: Xây dựng ứng dụng web sử dụng công nghệ React, NextJs, Golang, NodeJs, Docker

Hanoi, 9/2023

Sinh viên thực hiện: Nguyễn Duy Cường

MSSV: 20191714 – Lớp: CLC HTTT K64

Doanh nghiệp: **Trung tâm Công nghệ Thông tin MobiFone**

- Chi nhánh Tổng Công ty Viễn thông MobiFone

I- Nội dung thực tập được giao:

- Tìm hiểu ngôn ngữ JavaScript và framework ReactJs, NodeJs và ExpressJs:
 - Cú pháp đơn giản
 - DOM
 - RESTful API
 - HTTP request
- Xây dựng Web Todo List hoàn chỉnh sử dụng ReactJs cho client side và NodeJs cho Server side.
 - Thiết kế Giao diện Người dùng
 - Xây dựng Back-end
 - Tích hợp và triển khai Front-end và Back-end

II- Kết quả thực hiện:

(Source code [available Here](#))

1. Tìm hiểu ngôn ngữ JavaScript và framework ReactJs, NodeJs và ExpressJs.

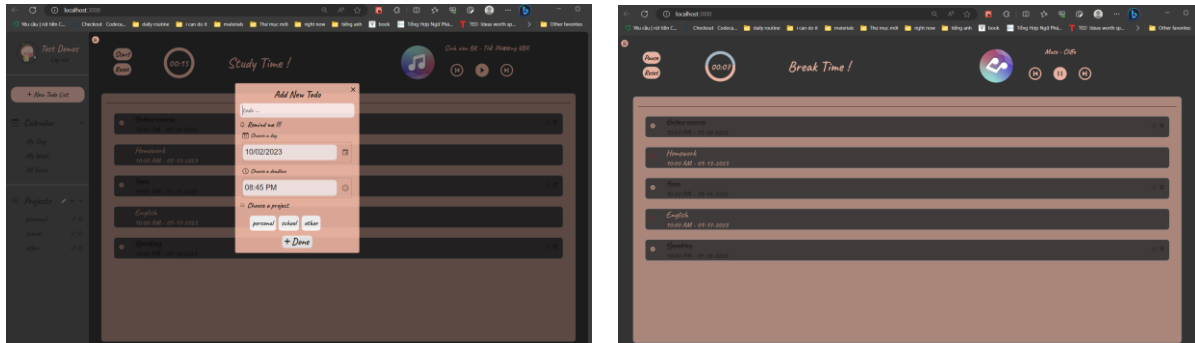
- JS là một ngôn ngữ lập trình phía máy khách cho phép tạo trải nghiệm tương tác trên trình duyệt, thay đổi nội dung trang và xử lý sự kiện người dùng.
- React.js sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện và tạo hiệu suất tốt cho ứng dụng web.

- DOM là một biểu diễn cấu trúc cây của trang web HTML hoặc XML, cho phép JavaScript tương tác với và thay đổi nội dung và cấu trúc của trang web.
- JavaScript có thể sử dụng DOM để thêm, sửa đổi và xóa phần tử trên trang
- Node.js là một môi trường thực thi mã JavaScript phía máy chủ, cho phép viết mã JS để xây dựng ứng dụng máy chủ và xử lý yêu cầu từ phía máy khách.
 - HTTP Request là cách máy khách (trình duyệt) gửi yêu cầu đến máy chủ web để nhận dữ liệu hoặc thực hiện hành động như lấy trang web, gửi dữ liệu hoặc tải tệp.
 - Có nhiều loại yêu cầu HTTP như GET, POST, PUT, DELETE, và nhiều loại yêu cầu khác.
- Express.js là một framework Node.js phía máy chủ, giúp xây dựng các ứng dụng web và API một cách dễ dàng và nhanh chóng, cung cấp cơ chế định tuyến và middleware để quản lý yêu cầu HTTP.
 - API là một tập hợp các định tuyến và xử lý yêu cầu HTTP mà đã được xác định trong ứng dụng Express.
 - RESTful API là một kiến trúc thiết kế dựa trên các nguyên tắc và quy tắc chuẩn để tạo ra các dịch vụ web có thể truy cập thông qua giao thức HTTP.

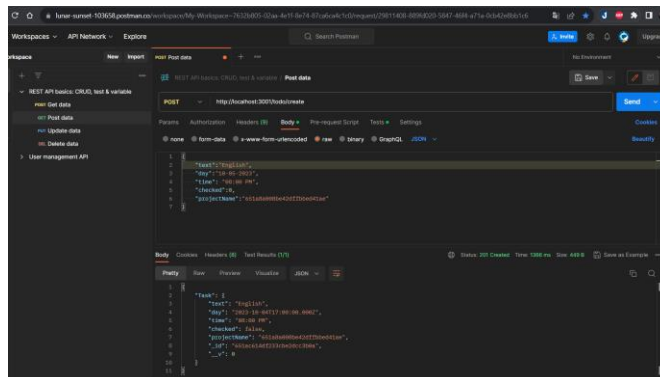
2. Xây dựng Web Todo

- Giao diện Web:
 - Xây dựng giao diện trang chính cho To-Do List.
 - Hiển thị danh sách công việc cần làm.
 - Cho phép thêm công việc mới.
 - Cho phép xóa công việc và đánh dấu công việc đã hoàn thành.
 - Tạo các thành phần React cần thiết.
 - Xây dựng các function cho nhiệm vụ CRUD và hỗ trợ phía Server.
- Back-end:
 - Thiết lập dự án Node.js và Express.js.
 - Khởi tạo dự án Node.js và cài đặt Express.js.
 - Thiết lập cơ bản cho API và xử lý yêu cầu.
 - Kết nối với Cơ sở dữ liệu MongoDB.
 - Cấu hình kết nối với MongoDB.
 - Xây dựng Schema và mô hình dữ liệu cho danh sách công việc.

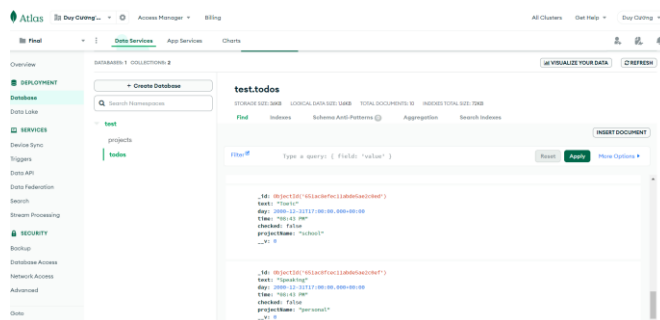
- Xây dựng API cho To-Do List.
 - Tạo các endpoint API cho CRUD tasks.
 - Xử lý yêu cầu từ phía client và tương tác với cơ sở dữ liệu.
 - Tích hợp:
 - Tích hợp giao diện người dùng React với API của Express.js.
- ### 3. DEMO
- Giao diện:



Postman:



Database:



III- Tự đánh giá:

- Ưu điểm:
 - Thực hiện thực tập đầy đủ số buổi đã đăng ký trước.
 - Hoàn thành được nhiệm vụ được giao.
- Nhược điểm:
 - Chưa có sự quyết liệt trong quá trình thực tập, không đẩy nhanh được tiến độ học tập.
 - Chưa chủ động trong quá trình học kiến thức mới.