

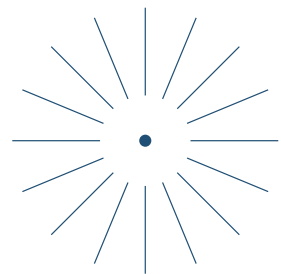
# ReactJS

---

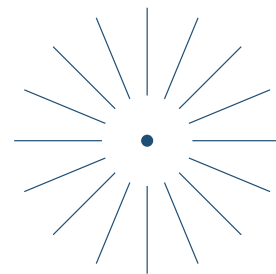


React

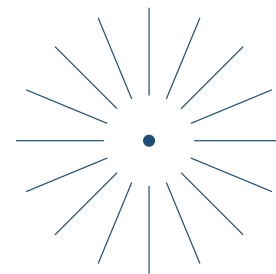
# Nội Dung



kiến thức cơ bản



Kiến thức nâng cao



sản phẩm



# 1, Kiến thức cơ bản



- **01** React Components
- **02** State và Hook
- **03** Routing
- **04** Form Handling
- **05** API Requests

# React Components

## Khái Niệm:

Components là các khối xây dựng cơ bản trong React, giúp tái sử dụng mã nguồn và tách biệt chức năng của ứng dụng.

## Chức Năng:

- Tạo giao diện người dùng linh hoạt, chia ứng dụng thành các phần nhỏ để quản lý.
- Nhận props (tham số) và trả về các thành phần hiển thị

```
import React, { useState } from "react";

const LoginForm = () => {
  const [formData, setFormData] = useState({
    email: "",
    password: "",
  });

  // ... (rest of the code)
};
```

# State và Hook

## Khái Niệm:

- State là trạng thái nội tại của một component, thay đổi khi dữ liệu hoặc sự kiện xảy ra.
- useState là hook giúp quản lý state trong functional components.

## Chức Năng:

- Lưu và cập nhật trạng thái của component.
- Trigger re-render khi state thay đổi.

```
// LoginForm.js
import React, { useState } from "react";

const LoginForm = () => {
  const [formData, setFormData] = useState({
    email: "",
    password: "",
  });

  // ... (rest of the code)
};
```

# Routing

## Khái Niệm:

- Định tuyến quyết định cách các component hiển thị dựa trên URL.
- react-router-dom cung cấp các thành phần giúp xây dựng hệ thống định tuyến.

## Chức Năng:

- Chuyển hướng giữa các trang mà không làm tải lại toàn bộ ứng dụng.
- Điều hướng dựa trên điều kiện của URL.
- 

```
// LoginForm.js
import React from "react";
import { useNavigate } from 'react-router-dom';

const LoginForm = () => {
  const navigate = useNavigate();

  // ... (rest of the code)

  const handleRegisterClick = () => {
    navigate("/register"); // Điều hướng tới trang R
  };
};
```

# Form Handling

## Khái Niệm:

- Xử lý sự kiện khi người dùng nhập liệu vào các form.
- Sử dụng các sự kiện như onChange và onSubmit để tương tác với dữ liệu form.

## Chức Năng:

- Nhận và xử lý dữ liệu nhập từ người dùng.
- Gửi yêu cầu khi form được gửi.

```
const LoginForm = () => {  
  const handleChange = (e) => {  
    setFormData({ ...formData, [e.target.name]: e.target.value }  
  );  
  
  const handleSubmit = async (e) => {  
    e.preventDefault();  
  
    try {  
      const response = await api.login(formData);  
      alert("Đăng nhập thành công!!!");  
      console.log(response.data);  
      navigate("/home");  
    } catch (error) {  
      console.error("Đăng nhập không thành công:", error.message);  
      alert("Đăng nhập không thành công!!!");  
    }  
  };  
  
  return (  
    // ... (rest of the code)  
  );  
};
```

# API Requests

## Khái Niệm:

- Gửi yêu cầu HTTP đến server để lấy hoặc gửi dữ liệu.
- Sử dụng thư viện như axios để thực hiện các yêu cầu.

## Chức Năng:

- Tương tác với backend server để lấy hoặc gửi dữ liệu.
- Xử lý kết quả từ server.

```
// api.js
import axios from 'axios';

const API_URL = 'http://localhost:3001/api';

const api = {
  // ... (rest of the code)

  login: async (userData) => {
    try {
      const response = await axios.post(`${API_URL}/login`, userData);
      return response.data;
    } catch (error) {
      throw error.response.data;
    }
  },
};
```



# Kiến Thức Nâng Cao

---



# Kiến Thức Nâng Cao

- State và Hook Nâng Cao
- Bảo Mật và Mã Hóa Mật Khẩu
- Toast Notifications
- CORS



# State và Hook Nâng Cao

## Sử dụng useEffect:

- useEffect là hook giúp xử lý các tác vụ sau khi component đã được render.

## Chức Năng:

- Thực hiện các tác vụ như gọi API, làm sạch trạng thái, hoặc đăng ký sự kiện.
- Quản lý các side effects trong functional components.

```
// AppointmentForm.js
import React, { useState, useEffect } from "react";

function AppointmentForm() {
  // ... (rest of the code)

  useEffect(() => {
    window.scrollTo({ top: 0, behavior: "smooth" });
  });

  // ... (rest of the code)
};
```

# Bảo Mật và Mã Hóa Mật Khẩu

## Khái niệm:

- bcrypt là thư viện mã hóa mật khẩu để bảo vệ thông tin người dùng.

## Chức Năng:

- Mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu.
- Bảo vệ thông tin người dùng khỏi tấn công từ bên ngoài.

```
// server.js
const bcrypt = require('bcrypt');

// ... (rest of the code)

app.post('/api/register', async (req, res) => {
  try {
    const { username, email, password } = req.body;

    // ... (rest of the code)

    // Mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu
    const hashedPassword = await bcrypt.hash(password, 10);

    // ... (rest of the code)
  } catch (error) {
    console.error('Đăng ký không thành công:', error);
    res.status(500).json({ message: 'Lỗi server' });
  }
});
```

# Toast Notifications

## Khái Niệm:

- react-toastify giúp hiển thị thông báo ngắn một cách dễ dàng.

## Chức Năng:

- Cung cấp phản hồi người dùng thông qua các thông báo ngắn.
- Hiển thị thông báo thành công hoặc lỗi một cách thân thiện.

```
// SubscribeNewsletter.js
import React, { useState } from "react";
import { ToastContainer, toast } from "react-toastify";

function SubscribeNewsletter() {
  // ... (rest of the code)

  const handleBookAppointmentClick = async () => {
    // ... (rest of the code)

    try {
      const response = await axios.post("http://localhost:3001/api", {
        email: inputEmail,
      });

      console.log(response.data.message);

      toast.success(response.data.message, {
        // ... (rest of the code)
      });
    } catch (error) {
      // ... (rest of the code)
    }
  };
}
```

# CORS

## Khái Niệm:

- Giải quyết vấn đề CORS để đảm bảo an toàn trong giao tiếp giữa frontend và backend.

## Chức Năng:

- Đảm bảo sự tin cậy và an toàn khi trao đổi dữ liệu giữa client và server.
- Xử lý vấn đề liên quan đến chính sách CORS.

```
// server.js
const cors = require('cors');

app.use(cors());
```



# Health-Plus

Chào mừng đến với sản phẩm của chúng tôi





# Các chức năng chính



1

Login & Register

2

Đặt Lịch Hẹn

3

Đăng Ký Nhận  
Bản Tin



# Login & Register

## Login Form

Email

newAccount123@gmail.com

Password

.....

LOGIN

REGISTER

### Login

- Sử dụng React để tạo giao diện đăng nhập.
- Kết nối với Express backend để xác thực người dùng.

## Register Form

Username

NewAccont

Email

newAccount123@gmail.com

Password

.....

REGISTER

LOGIN

### Register

- Tạo giao diện đăng ký và gửi yêu cầu đăng ký đến server Express.
- Mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu MongoDB.

# Đặt Lịch Hẹn

## Mô Tả:

- Sử dụng React để xây dựng giao diện đặt lịch hẹn.
- Kết nối với Express để gửi yêu cầu đặt lịch hẹn thông qua API.
- Xử lý kiểm tra và hiển thị kết quả đặt lịch hẹn.

## Why Choose Health

over the reasons to choose Health Plus for your healthcare needs. Experience expert venience, and personalized solutions, making your well-being our top priority. Join us journey to better health and a happier life.

✓ Best Professional Doctors

✓ Emergency Care

✓ 24/7 Support Live Chat

✓ Enrollment Easy and Quick

📅 Book Appointment

Health +

## Book Appointment Online

\* Patient Full Name:


\* Patient Phone Number:

\* Patient Gender:

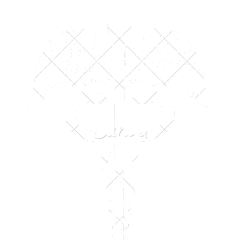
\* Preferred Appointment Time:

\* Preferred Mode:

Confirm Appointment



# Đăng Ký Nhận Bản Tin



## Mô Tả:

- Sử dụng React để tạo giao diện đăng ký nhận bản tin.
- Gửi yêu cầu đăng ký đến Express backend qua API.
- Xử lý kết quả và hiển thị thông báo đăng ký thành công hoặc lỗi.

## Health +

Talk to online doctors and get medical advice, online prescriptions, refills and medical notes within minutes. On-demand healthcare services at your fingertips.

**Đăng ký nhận bản tin của chúng tôi \***

Nhập địa chỉ email của bạn

Đăng ký



**Thank You  
For Your Time!**