

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----



# **BÁO CÁO**

## **THỰC TẬP CƠ SỞ**

**Giảng viên: Kim Ngọc Bách**

**Tuần 3: 29/3/2025**

**Sinh viên: Nguyễn Khắc Trường B22DCCN884**

# MỤC LỤC

MỤC LỤC.....	2
A. Kiến thức học được tuần này .....	1
1. Vue Router .....	1
2. Vue Store.....	1
3. Thư mục Apis.....	3
B. Tiến độ dự án.....	5
1. Hoàn thành ERD .....	5
2. Hoàn thành Data Flow Diagram .....	6
3. Hoàn thành Sequence Diagram.....	7
4. Hoàn thành 3 page login, register, profile bên phía FrontEnd .....	10

# A. Kiến thức học được tuần này

## 1. Vue Router

### Vue Router là gì?

Vue Router là thư viện quản lý định tuyến (routing) cho **Vue.js**, giúp tạo các ứng dụng **SPA (Single Page Application)** có nhiều trang mà không cần tải lại trang.

### Tính năng chính:

- **Định tuyến động:** Chuyển trang dựa trên URL mà không cần tải lại.
- **Lazy Loading:** Tải component theo yêu cầu để tối ưu hiệu suất.
- **Guard (Bảo vệ route):** Kiểm soát quyền truy cập vào các trang.
- **History Mode:** Hỗ trợ hash mode (#) hoặc history mode dùng API của trình duyệt.

Ví dụ áp dụng route để định tuyến các page:

```
import { createRouter, createWebHistory } from 'vue-router'
import Home from '@views/Home.vue'
import About from '@views/About.vue'

const routes = [
  { path: '/', component: Home },
  { path: '/about', component: About }
]

const router = createRouter({
  history: createWebHistory(),
  routes
})

export default router
```

## 2. Vue Store

**Store** trong Vue là một cơ chế quản lý trạng thái tập trung, giúp chia sẻ dữ liệu giữa các component một cách hiệu quả. Khi ứng dụng phát triển lớn, việc quản lý trạng thái trở nên phức tạp, và **Pinia** (hoặc Vuex - phiên bản cũ) là giải pháp giúp tổ chức dữ liệu tốt hơn.

Store thường được chia thành các module nhỏ hơn, Khi ứng dụng Vue lớn dần, quản lý trạng thái trong một store duy nhất trở nên khó khăn. Chia module giúp **tổ chức, tách biệt và tối ưu** việc quản lý trạng thái.

Ví dụ tách Store thành các modules:

```
import { defineStore } from 'pinia'

export const useAuthStore = defineStore('auth', {
  state: () => ({
    user: null
  }),
  actions: {
    login(user) {
      this.user = user
    },
    logout() {
      this.user = null
    }
  }
})
```

(module auth)

```
import { defineStore } from 'pinia'

export const useCartStore = defineStore('cart', {
  state: () => ({
    items: []
  }),
  actions: {
    addItem(item) {
      this.items.push(item)
    }
  }
})
```

(module cart)

### 3. Thư mục Apis

Trong một dự án Vue, thư mục apis thường được sử dụng để tổ chức và quản lý các API request, giúp code gọn gàng, dễ bảo trì và tái sử dụng.

- Lợi ích của thư mục apis:
  - + Tách biệt logic gọi API → Giúp component không bị rối.
  - + Dễ quản lý và bảo trì → Khi API thay đổi, chỉ cần chỉnh sửa ở một nơi.
  - + Dễ dàng tái sử dụng → Tránh lặp lại code gọi API.
- Cũng như store api cũng được chia ra thành các module nhỏ

```
src/
├─ apis/
│   ├─ auth.js
│   ├─ user.js
│   ├─ product.js
│   └─ cart.js
```

- Ví dụ trong 1 file module api

```
import axios from '@utils/axios' // Import instance axios đã cấu hình

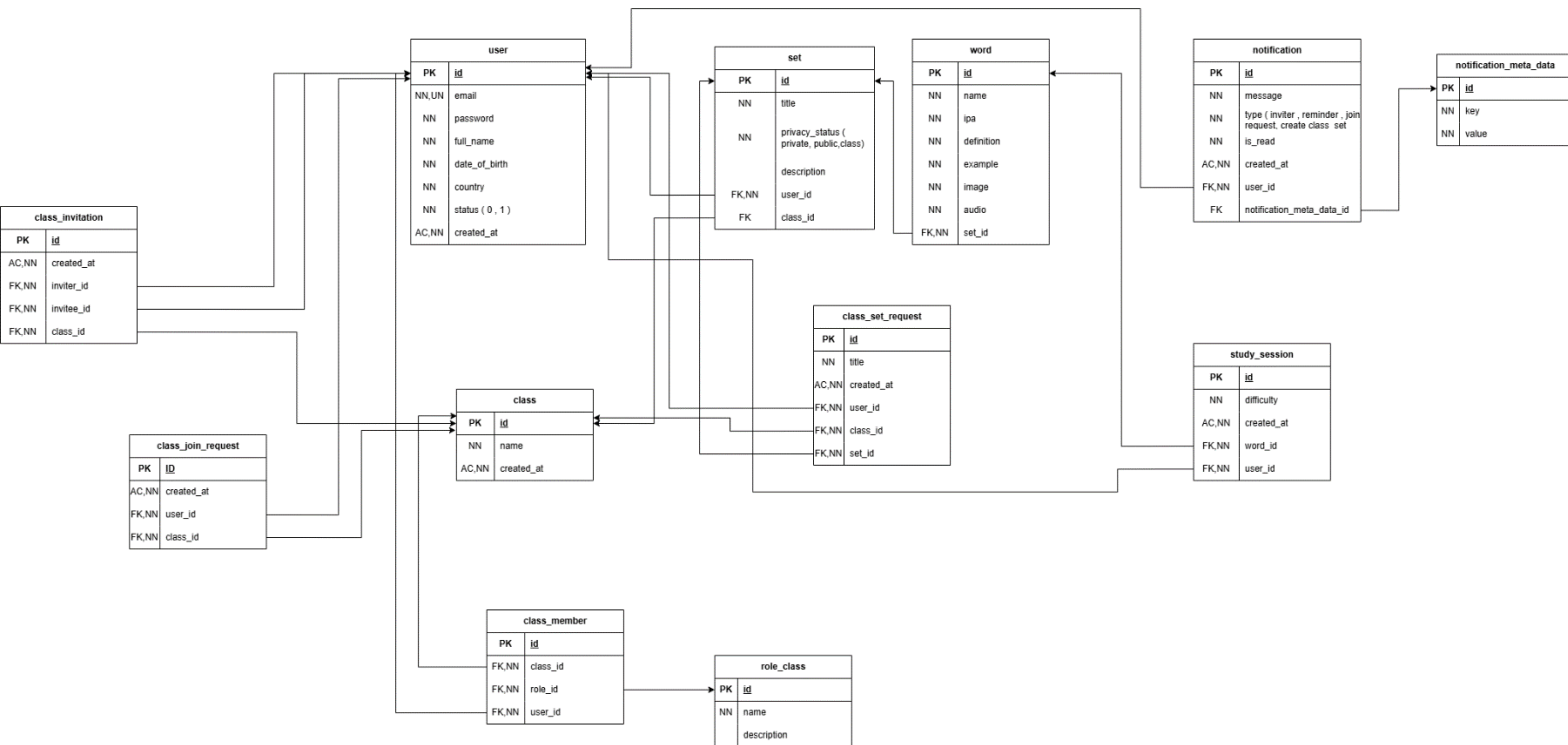
export const login = (credentials) => {
  return axios.post('/auth/login', credentials)
}

export const register = (userData) => {
  return axios.post('/auth/register', userData)
}

export const logout = () => {
  return axios.post('/auth/logout')
}
```

## B. Tiến độ dự án

### 1. Hoàn thành ERD



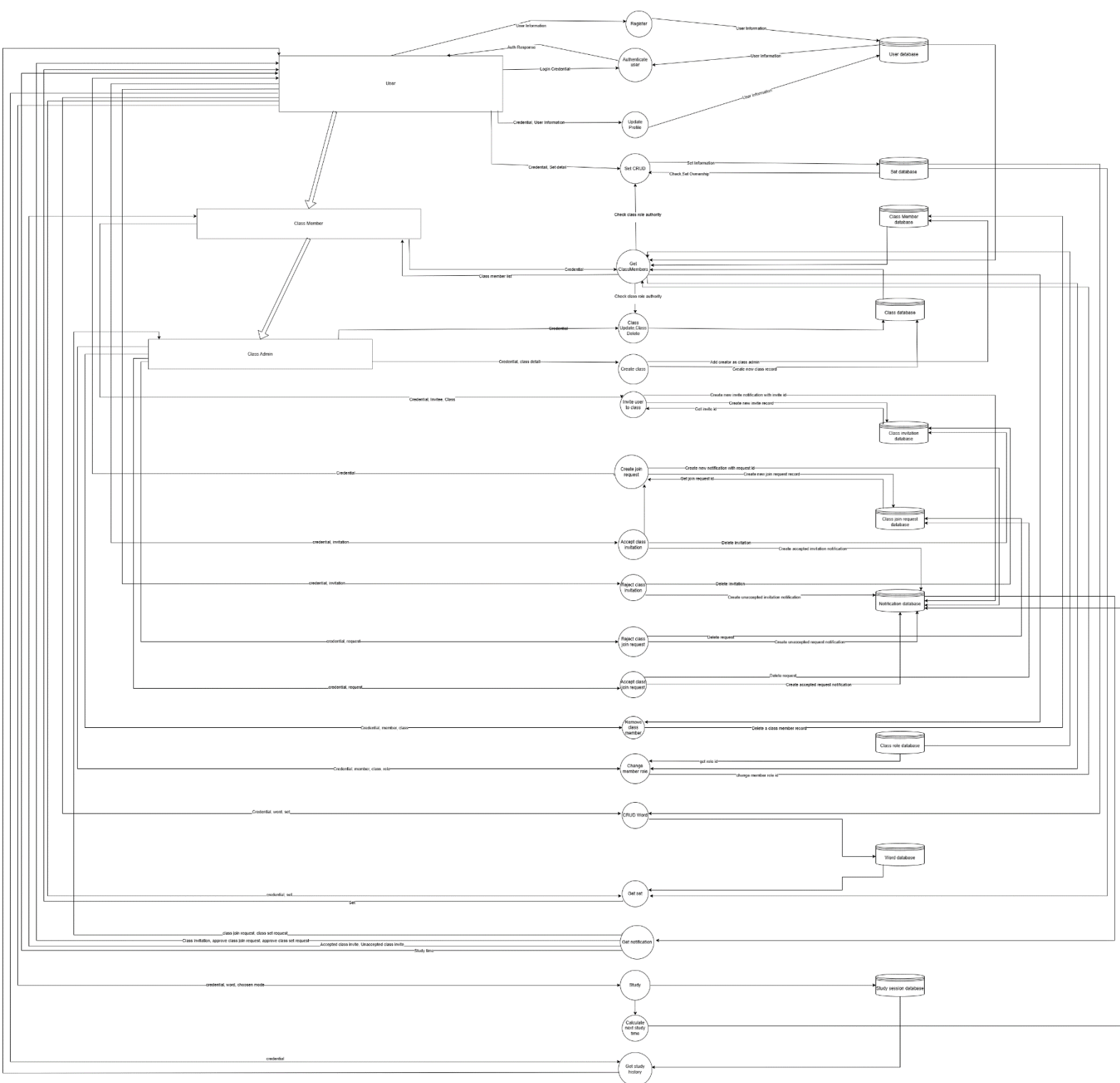
- Tổng có 12 bảng bao gồm
  - + user quản lý tài khoản người dùng.
  - + set quản lý các tập word.
  - + word quản lý các từ vựng mà người dùng học.
  - + class quản lý các lớp, nhóm.
  - + notification quản lý các thông báo đến các tài khoản người dùng.
  - + study\_session quản lý các lần học của người dùng, nó sẽ dùng để tính toán các lần thông báo học tiếp theo.
  - + class\_join\_request quản lý các yêu cầu gia nhập lớp.
  - + class\_invitation quản lý các lời mời gia nhập lớp.

- + `class_member` quản lý các thành viên của nhóm, lớp.
- + `role_class` quản lý các vai trò trong nhóm.
- + `class_set_request` quản lý các set mà đang trong trạng thái chờ xét duyệt để được xuất hiện trong lớp, nhóm học.
- + `notification_meta_data` giúp việc truy xuất notification sẽ lấy thêm được các id, request,...

## **2. Hoàn thành Data Flow Diagram**

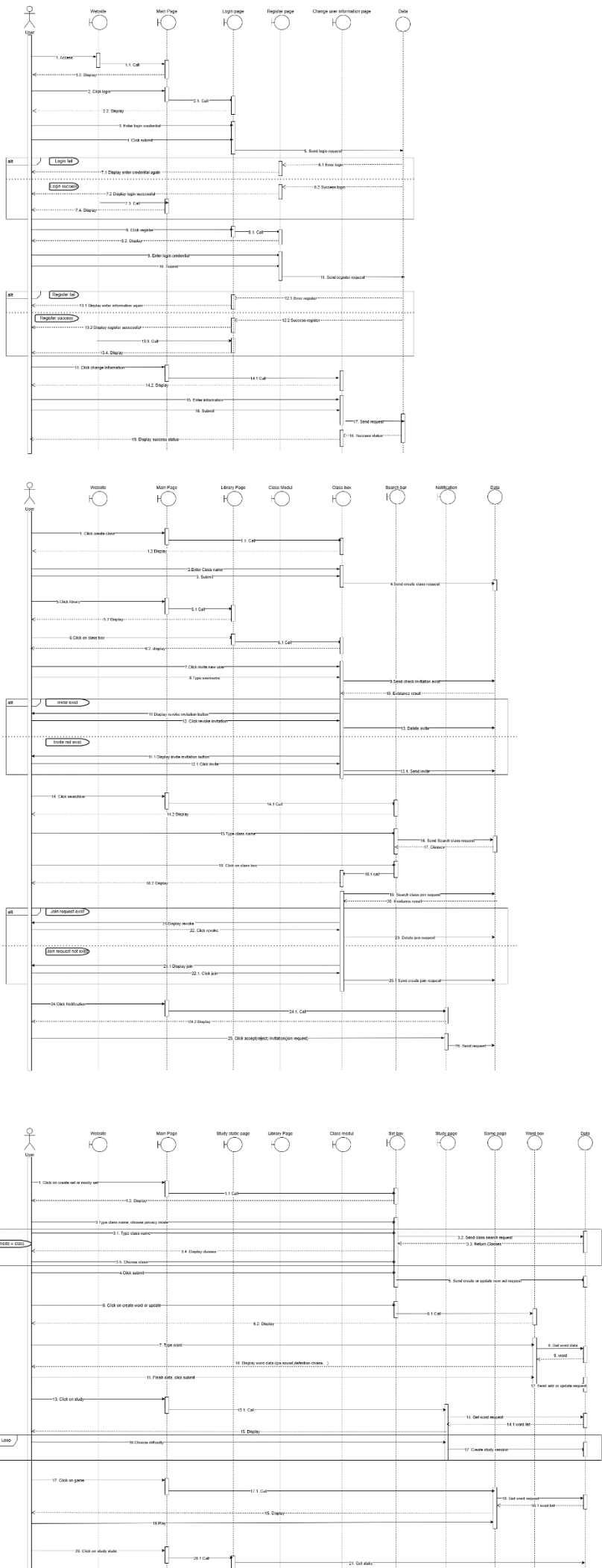
- DFD sẽ miêu tả luồng dữ liệu từ các hàm đến các database cũng như xem các hàm sẽ sử dụng các database nào.



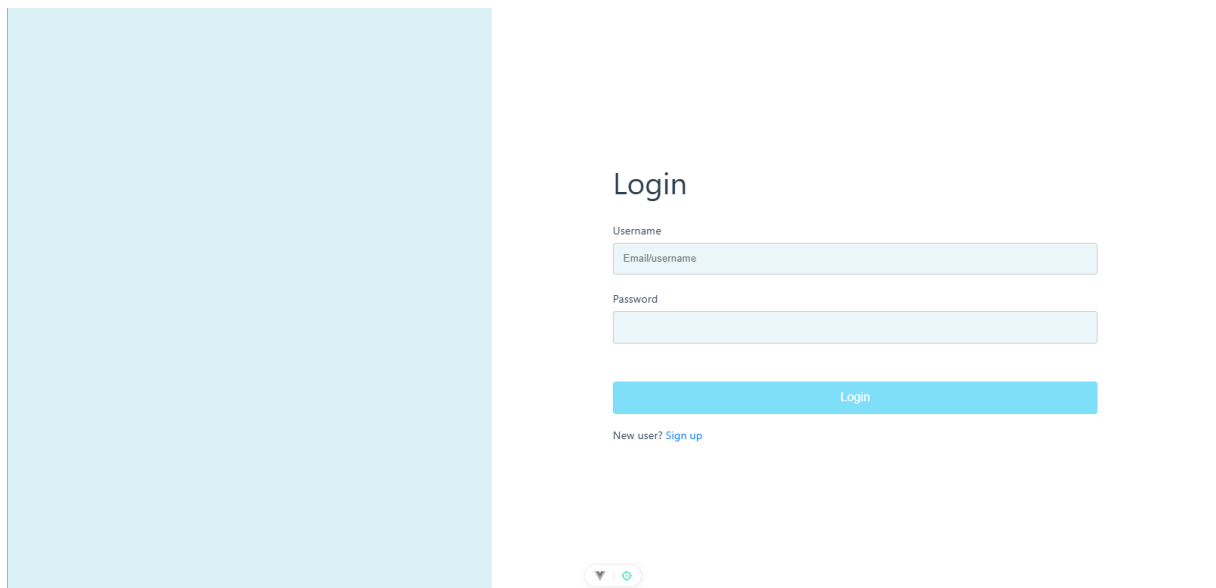


### 3. Hoàn thành Sequence Diagram

- Ở bên phải em miêu tả luồng các page frontend giao tiếp với nhau cũng như giao tiếp với người dùng.
- Ở bên trái em miêu tả luồng cả hệ thống tương tác với nhau

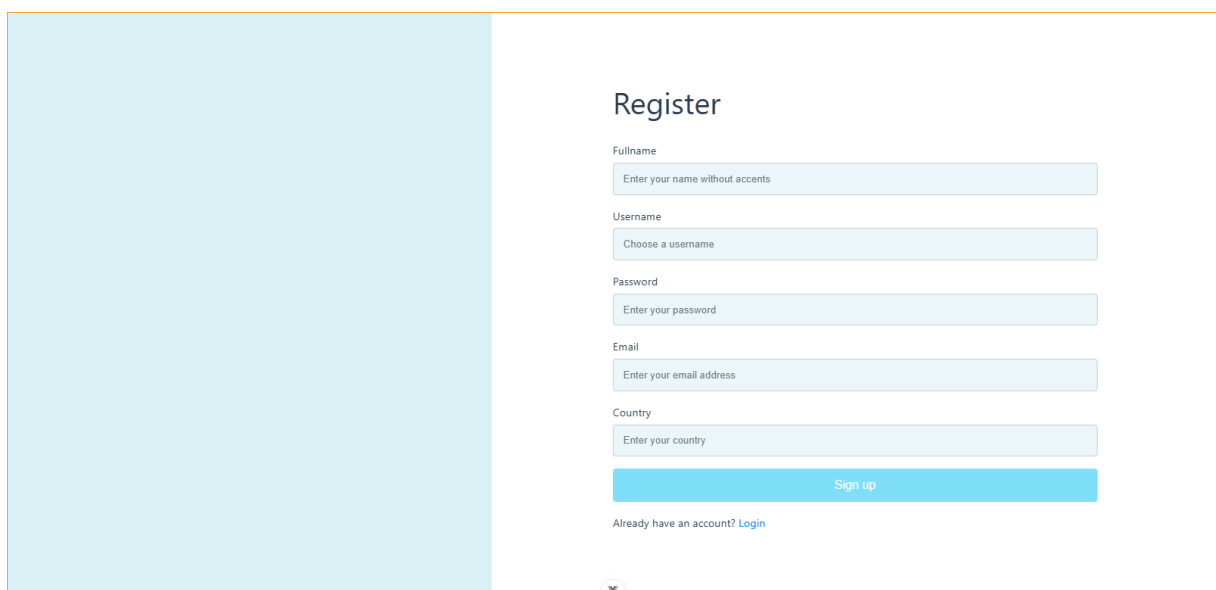


#### 4. Hoàn thành 3 page login, register, profile bên phía FrontEnd



The image shows a login form on a light gray background. On the left, there is a large, solid light blue rectangular area. The form itself is on the right and includes the following elements: a title 'Login' in a dark gray font; a label 'Username' above a text input field with the placeholder text 'Email/username'; a label 'Password' above another text input field; a solid blue button with the text 'Login' in white; and a link 'New user? Sign up' in blue text below the button. At the bottom center of the form area, there is a small, light gray rounded rectangle containing a downward arrow and a green checkmark icon.

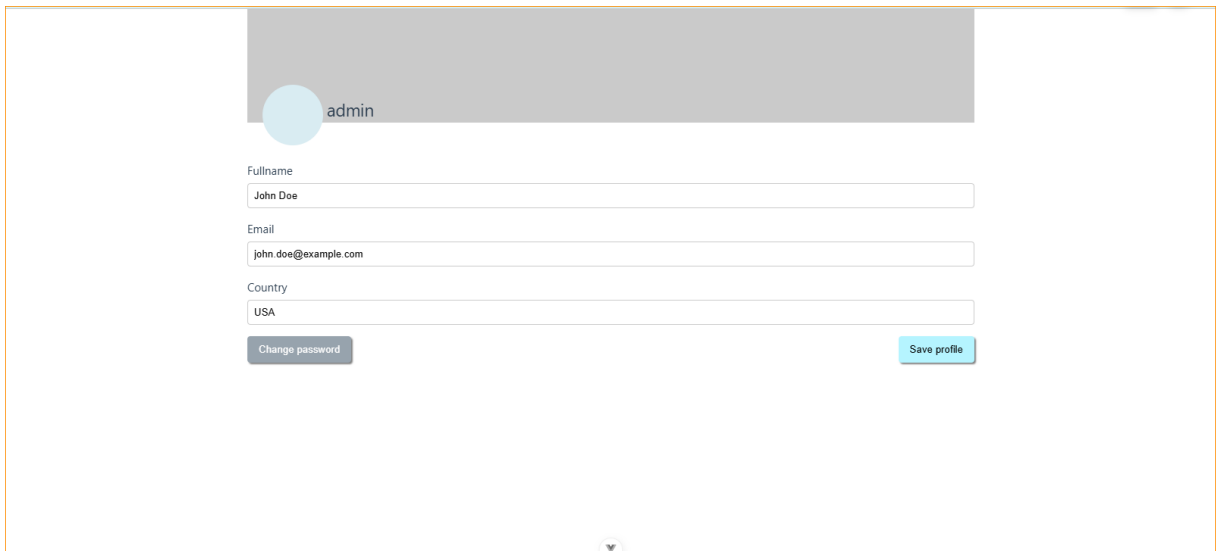
(/login)



The image shows a register form on a light gray background. On the left, there is a large, solid light blue rectangular area. The form is on the right and includes the following elements: a title 'Register' in a dark gray font; a label 'Fullname' above a text input field with the placeholder text 'Enter your name without accents'; a label 'Username' above a text input field with the placeholder text 'Choose a username'; a label 'Password' above a text input field with the placeholder text 'Enter your password'; a label 'Email' above a text input field with the placeholder text 'Enter your email address'; a label 'Country' above a text input field with the placeholder text 'Enter your country'; a solid blue button with the text 'Sign up' in white; and a link 'Already have an account? Login' in blue text below the button. At the bottom center of the form area, there is a small, light gray rounded rectangle containing a downward arrow icon.

(/register)

- Sau khi đăng ký xong sẽ chuyển sang trang login

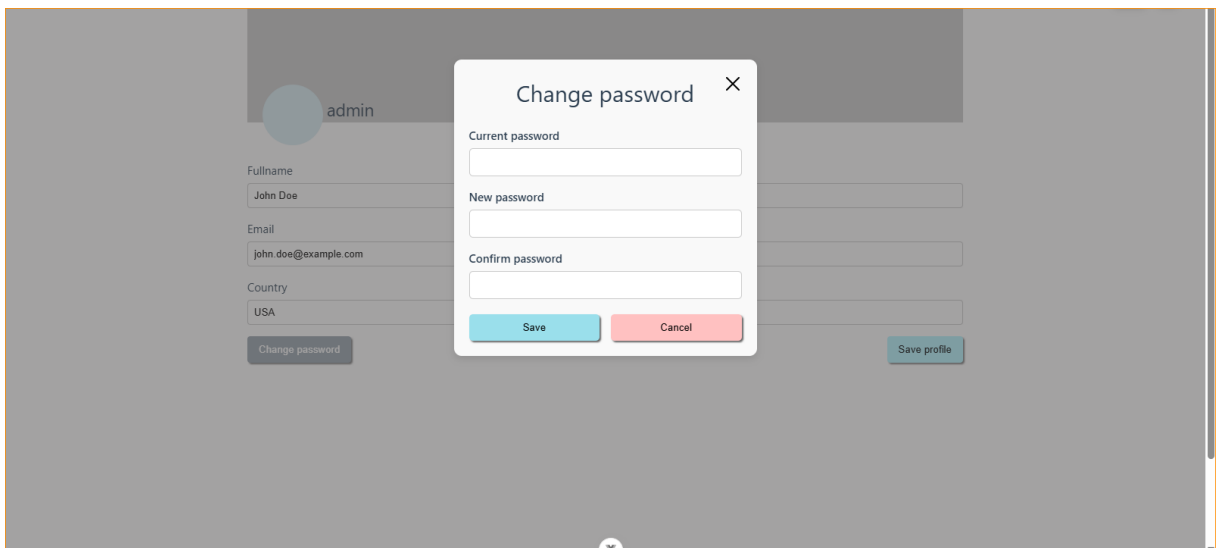


A user profile form with a header bar containing a blue circular profile picture and the text 'admin'. Below the header, there are four input fields: 'Fullname' with the value 'John Doe', 'Email' with the value 'john.doe@example.com', and 'Country' with the value 'USA'. At the bottom of the form, there are two buttons: 'Change password' on the left and 'Save profile' on the right.

(/profile)

- Ở trang profile sẽ có đổi thông tin cá nhân hoặc đổi mật khẩu

-



A modal dialog titled 'Change password' with a close button (X) in the top right corner. The modal contains three input fields: 'Current password', 'New password', and 'Confirm password'. At the bottom of the modal, there are two buttons: 'Save' (blue) and 'Cancel' (red). The modal is overlaid on a blurred background of the user profile form.

(pop up change password)

