MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

Thực hành và củng cố hiểu biết của mình về Vue Router bằng cách triển khai các tính năng chính như Routes lồng nhau, Routes động, Bảo vệ Routes với xác thực giả lập, Tên Routes và Alias Routes.

NỘI DUNG

Yêu cầu chung:

Tạo một ứng dụng Vue.js đơn giản có tên là "MyBlog" nơi người dùng có thể điều hướng giữa các trang khác nhau, bao gồm danh sách blog, các bài viết blog riêng lẻ, và trang thông tin người dùng. Một số routes sẽ được bảo vệ bằng cách xác thực, và các routes khác sẽ hiển thị việc sử dụng routes lồng nhau, routes động, và alias routes.

PHẦN I: Thiết lập ứng dụng và tạo Component

Yêu cầu 1 (2 điểm): Thiết lập ứng dụng Vue.js cơ bản với Vue Router

- Khởi tạo một dự án Vue.js mới.
- Cài đặt Vue Router và thiết lập.

Yêu cầu 2 (3 điểm): Tạo các Component sau:

- **Home.vue**: Trang chủ đơn giản chào mừng người dùng đến "MyBlog."
- **BlogList.vue**: Trang hiển thị danh sách các bài viết blog.
- **BlogPost.vue**: Trang hiển thị chi tiết một bài viết blog (sử dụng routes động).
- **UserProfile.vue**: Trang thông tin người dùng hiển thị thông tin về người dùng đã đăng nhập.
- **Login.vue**: Trang đăng nhập nơi người dùng có thể đăng nhập để truy cập các routes được bảo vệ.
- **Dashboard.vue**: Route được bảo vệ chỉ có người dùng đã xác thực mới có thể truy cập, hiển thị bảng điều khiển đơn giản.

PHẦN II: Triển khai các Routers

Yêu cầu 3 (5 điểm): Thực hiện các yêu cầu sau:

a. ***Routes lồng nhau***:
- Tạo một route lồng nhau cho trang thông tin người dùng (/profile) với hai routes con:
- /profile/info: Hiển thị thông tin cơ bản của người dùng.
- /profile/settings: Hiển thị cài đặt người dùng.

b. ***Routes động:***
- Triển khai một route động cho các bài viết blog. Route sẽ là /blog/:id, trong đó :id là đoạn động đại diện cho ID của bài viết.
- Thành phần BlogPost.vue nên sử dụng tham số động id để hiển thị nội dung bài viết chính xác.

c. ***Bảo vệ Routes với xác thực giả lập:***
- Triển khai một hệ thống xác thực giả lập:
- Sử dụng một biến boolean (ví dụ: isAuthenticated) trong route guard để giả lập việc người dùng đã đăng nhập.
- Bảo vệ route /dashboard bằng một route guard. Nếu người dùng chưa được xác thực, chuyển hướng họ đến trang đăng nhập.

d. ***Tên Routes:***
- Gán tên cho các routes của bạn, chẳng hạn như Home, BlogList, BlogPost, UserProfile, Login, và Dashboard.
- Thực hiện điều hướng đến các routes này bằng cách lập trình sử dụng this.$router.push({ name: 'routeName' }).

e. ***Alias Routes:***
- Tạo một alias cho route /profile. Alias này sẽ là /me, do đó khi điều hướng đến /me cũng sẽ hiển thị thành phần UserProfile.

Hướng dẫn:

1. **Tạo các thành phần:**

   Tạo các thành phần cần thiết (Home.vue, BlogList.vue, BlogPost.vue, UserProfile.vue, Login.vue, và Dashboard.vue) trong thư mục src/views/.

2. **Cấu hình Router:**

   Trong src/router/index.js, định nghĩa các routes của bạn.

Xem mã code như hình sau:

index.js

teaching > vue > my-vue > src > router > index.js > ...

```
import { createRouter, createWebHistory } from "vue-router";
import Home from "../views/Home.vue";
import BlogList from "../views/BlogList.vue";
import BlogPost from "../views/BlogPost.vue";
import UserProfile from "../views/UserProfile.vue";
import UserProfileInfo from "../views/UserProfileInfo.vue";
import UserProfileSettings from "../views/UserProfileSettings.vue";
import Login from "../views/Login.vue";
import Dashboard from "../views/Dashboard.vue";

const isAuthenticated = false; // Mock authentication state

const routes = [
  { path: "/", name: "Home", component: Home },
  { path: "/blog", name: "BlogList", component: BlogList },
  { path: "/blog/:id", name: "BlogPost", component: BlogPost },
  {
    path: "/profile",
    name: "UserProfile",
    component: UserProfile,
    alias: "/me",
    children: [
      { path: "info", name: "UserProfileInfo", component: UserProfileInfo },
      {
        path: "settings",
        name: "UserProfileSettings",
        component: UserProfileSettings,
      },
    ],
  },
  { path: "/login", name: "Login", component: Login },
  {
    path: "/dashboard",
    name: "Dashboard",
    component: Dashboard,
    meta: { requiresAuth: true },
  },
];

const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes,
});

// Route Guard for Authentication
router.beforeEach((to, from, next) => {
  if (
    to.matched.some((record) => record.meta.requiresAuth) &&
    !isAuthenticated
  ) {
    next({ name: "Login" });
  } else {
    next();
  }
});

export default router;
```

3. **Implement Views:**
   - Trong BlogPost.vue, sử dụng tham số động id để hiển thị nội dung của một bài viết dựa trên id.
   - Trong UserProfile.vue, thiết lập thanh điều hướng hoặc liên kết để điều hướng giữa các routes info và settings.
   - Thực hiện logic đăng nhập trong Login.vue để cập nhật biến isAuthenticated và chuyển hướng người dùng đến Dashboard sau khi đăng nhập thành công (đơn giản là giả lập đăng nhập thành công).

4. **Liên kết điều hướng:**

   - Sử dụng các thành phần <router-link> để điều hướng giữa các routes trong thành phần Home.vue và các thành phần liên quan khác.
   - Thực hiện điều hướng lập trình bằng cách sử dụng: this.$router.push({ name: 'routeName' }).

*** Yêu cầu nộp bài:

SV nén file (*hoặc share thư mục google drive*) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---