**BÁO CÁO BÀI TẬP**

**PHÁT TRIỂN HỆ THỐNG WEB AN TOÀN**

**ĐỀ TÀI:**

Tìm hiểu xây dựng extension trên các trình duyệt Firefox, Chrome, IE, Safari và trình duyệt di động

**Họ và tên : Đỗ Văn Tuấn - 20144863**

1. **Xây dựng extension trên chrome**

Thư mục chứa một tiện ích mở rộng trên chrome gồm một file config tên là manifest.json. Bên trong file này là toàn bộ thông tin mà Chrome cần biết để tải các thành phần bên trong tiện ích mở rộng như : tiêu đề, mô tả, version , permission và các icon v.v…

Ví dụ một file manifest:

{

"name": "Demo",

"version": "1.2.0",

"description": "demo",

"homepage\_url": "https://soict.hust.edu.vn/",

"background":

{

"scripts": [ "main.js" ]

},

"content\_scripts": [

{

"matches": ["https://soict.hust.edu.vn/"],

"js": ["content.js"]

}

],

"browser\_button": {

"caption": "Demo",

"tooltipText": "Demo",

"icon": "icons/button.png"

},

"manifest\_version": 2

}

Giải thích chi tiết một số trường:

**Name:** Tên của tiện ích mở rộng

**Icons:** chứa danh sách các icon có sẵn theo từng kích cỡ. Chrome sẽ tự động tìm kiếm các kích cỡ biểu tượng cụ thể để sử dụng trong các trường hợp khác nhau

**Browser\_action:**Khi chỉ định một browser\_action nghĩa là ta muốn thông báo với chrome rằng ta muốn đặt một biểu tượng ở góc phải ở thanh địa chỉ của trình duyệt và khi bấm vào biểu tượng nó thì tiện ích mở rộng sẽ được chạy. Chỉ nên thiết lập một browser\_action khi mà tiện ích mở rộng được áp dụng cho tất cả các trang web. Nếu chỉ muốn áp dụng nó trên một số trang hãy chỉ định một page\_action. Chú ý là không thể sử dụng browser\_action và page\_action cùng lúc.

**Permissions:** Trường này liệt kê các quyền hạn mà tiện ích mở rộng yêu cầu với trình duyệt. Có rất nhiều quyền hạn để lựa chọn tuy nhiên khi xây dựng một tiện ích mở rộng ta chỉ nên yêu cầu các quyền hạn mà mình cần.

Mọi chức năng chính của một tiện ích mở rộng được định nghĩa trong một file javascript đơn lẻ. File này sẽ bao gộm các hàm và các đối tượng mà tiện ích sử dụng.

Ngoài ra một tiện ích mở rộng còn có một trang nền để theo dõi trạng thái truy cập của các tab trong một trình duyệt đối với các chương trình tiện ích kiểm tra số lần truy cập một website hay chương trình chặn những website trong một blacklist. Lưu ý rằng ta chỉ được sử dụng duy nhất một trang nền.

Một tiện ích mở rộng có thể gọi ra một cửa sổ popup khi ta bấm vào biểu tượng (điểm bắt đầu của tiện ích mở rộng). Cửa sổ này hiển thị những thông tin cần thiết giúp người dùng dễ dàng theo dõi hoạt động của chương trình.

Tiếp theo là trang tùy chọn, đây là nơi bạn có thể thiết lập các giá trị, thông số cấu hình cho chương trình tiện ích mở rộng để chương trình chạy phù hợp với mục đích của từng người dùng cụ thể.

Để thử nghiệm một tiện ích mở rộng hãy nạp tất cả các file của tiện ích mà ta vừa xây dựng nhưng ko đóng gói chúng. Để nạp các tiện ích mở rộng đầu tiên chọn ô Developer Mode rồi chọn Load unpacked extension rồi chỉ tới thư mục chứa các file của tiện ích mở rộng để nạp nó. Nếu file tiện ích bị lỗi chrome sẽ thông báo và ko chạy tiện ích của bạn.

Trong bài tập này em sẽ xây dựng một tiện ích mở rộng đơn giản hiển thị lời chào tới người dùng.

Tiện ích mở rộng bao gồm 4 file đơn giản:

+ icon.png: file ảnh làm phần khởi đầu cho tiện ích

+ manifest.json: file mô tả tiện ích, khai báo các thành phần của tiện ích mở rộng

+ popup.html: phần hiển thị khi bấm vào nút khởi đầu của tiện ích

+ popup.js: phần xử lí chính của tiện ích.

1. **So sánh xây dựng tiện ích mở rộng trên chrome với firefox và safari.**
   1. *Chrome*

Đối với chrome, để trình duyệt nhận ra sự có mặt của tiện ích mở rộng khá dễ dàng. Ta chỉ cần chỉ định một mục browser\_action trong file khai báo và cung cấp cho nó một biểu tượng.

Trong chrome, bạn có quyền truy cập tới localStorage một cách ổn định xuyên suốt các mỗi lần mở ứng dụng này và điều đó giúp cho việc giữ ổn định dữ liệu giữa các phiên làm việc của trình duyệt trở nên đơn giản.

Tất cả các trang tạo nên tiện ích của chúng ta có thể liên lạc với trang nền thông qua chrome.extension.getBackgroundPage(), và một API truyền tin cho phép các tiện ích mở rộng liên lạc với nhau khi cần thiết.

* 1. *Firefox*

Không hề khó khăn hơn Chrome khi bạn muốn sự hiện diện của một thanh tiện ích (Add-on Bar) ở dưới cùng của trình duyệt. Bạn có thể làm được với việc tạo ra một Widget trong tệp main.js.

Có một chú ý nhỏ là để lưu trữ dữ liệu qua localStorage thì trên firefox cung cấp API Firefox-specific simple-storage. Muốn có một lớp Storage Manager để làm việc trên cả Chrome và Firefox bạn cần thực hiện một số tính năng phát hiện.

Để các phần khác nhau của tiện ích mở rộng trao đổi với nhau cần tạo ra một phương thức cho việc giao tiếp giữa các port và thiết lập các bộ thu (listeners) và phát (emitters).

* 1. *Safari*

Đối với safari thậm chí còn dễ dàng hơn so với chrome và firefox vì Extension Builder của Safari thực hiện nhiều phỏng đoán bên ngoài quy trình. Việc sử dụng localStorage giống với khi bạn sử dụng cho trình duyệt Chrome, điều đó giúp cho việc giữ ổn định dữ liệu giữa các phiên làm việc của trình duyệt trở nên đơn giản.

Bằng việc truy cập trực tiếp vào trang toàn cục, bạn có thể dễ dàng lấy thông tin đến và đi từ các phần khác nhau của trình mở rộng.