TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC

KHOA CÔNG NGHỆ THÔNG TIN



PRJ – REPORT  
**EXTENSIONS**

Giảng Viên: Phan Thanh Nhuần

**Nhóm D**:

1.Nguyễn Văn Thắng - 17211TT0480

2.Trương Khánh Hòa - 18211TT5272

3.Cao Đại Đạt - 18211TT2111

4.Nguyễn Văn Khải – 18211TT1987



**MỤC LỤC**

[I. HƯỚNG DẪN CÀI VÀ SỬ DỤNG EXTENSION VỚI CHROME 7](#_heading=h.1fob9te)

[1. Giới thiệu về extension 7](#_heading=h.3znysh7)

[2. Kiến trúc của extension 7](#_heading=h.2et92p0)

[3. Các extension hiện có 8](#_heading=h.1t3h5sf)

[II. HƯỚNG DẪN PHÁT TRIỂN EXTENSION AUTO LOGIN 10](#_heading=h.2s8eyo1)

[1. Quy Trình Phát Triển 10](#_heading=h.17dp8vu)

[1.1. Tạo project mới cho extension Auto Login 10](#_heading=h.3rdcrjn)

[1.2. Xây dựng file manifest.json 13](#_heading=h.z337ya)

[1.3. Xây dựng giao diện trong file popup.html 14](#_heading=h.3j2qqm3)

[1.4. Xây dựng chức năng trong file popup.js 16](#_heading=h.1y810tw)

[1.5. Xây dựng chức năng trong file content.js 20](#_heading=h.4i7ojhp)

[2. Hướng dẫn cài đặt extension Auto Login trên chrome 21](#_heading=h.2xcytpi)

[3. Áp dụng extension vào source code php-training 26](#_heading=h.1hmsyys)

[III. HƯỚNG DẪN PHÁT TRIỂN EXTENSION GET COOKIE 28](#_heading=h.vx1227)

[1. Quy trình phát triển 28](#_heading=h.3fwokq0)

[1.1. Tạo project mới cho extension Get Cookie: 28](#_heading=h.1v1yuxt)

[1.2. Xây dựng file mainifest.json 31](#_heading=h.37m2jsg)

[1.3. Xây dựng giao diện tại file popup.html 32](#_heading=h.1mrcu09)

[1.4. Xây dựng chức năng trong file popup.js 33](#_heading=h.46r0co2)

[2. Hướng dẫn cài đặt extension Get cookie trên chrome 35](#_heading=h.2lwamvv)

[3. Áp dụng extension vào source code php-training 43](#_heading=h.2r0uhxc)

[IV. HƯỚNG DẪN PHÁT TRIỂN EXTENSION MOUSE\_SELECT 45](#_heading=h.kgcv8k)

[1. Quy trình phát triển 45](#_heading=h.34g0dwd)

[1.1. Tạo project mới cho extension Mouse select 45](#_heading=h.1jlao46)

[1.2. Xây dựng file manifest.json 48](#_heading=h.2w5ecyt)

[1.3. Xây dựng giao diện tại file popup.html 49](#_heading=h.1baon6m)

[1.4. Xây dựng chức năng trong file popup.js 49](#_heading=h.3vac5uf)

[2. Hướng dẫn cài đặt extension Mouse select trên chrome 50](#_heading=h.2afmg28)

[3. Áp dụng extension vào source code php-training 56](#_heading=h.3ep43zb)

[V. HƯỚNG DẪN PHÁT TRIỂN EXTENSION ELEMENT CUSTOM 58](#_heading=h.184mhaj)

[1. Quy trình phát triển 58](#_heading=h.3s49zyc)

[1.1. Tạo project mới cho extension Element Custom 58](#_heading=h.279ka65)

[1.2. Xây dựng file manifest.json 62](#_heading=h.3jtnz0s)

[1.3. Xây dựng giao diện tại file popup.html 62](#_heading=h.1yyy98l)

[1.4. Xây dựng chức năng trong file popup.js 63](#_heading=h.4iylrwe)

[1.5. Xây dựng chức năng trong file content.js 66](#_heading=h.2y3w247)

[1.6. Xây dựng chức năng trong file option.html 67](#_heading=h.1d96cc0)

[1.7. Xây dựng chức năng trong file option.js 68](#_heading=h.3x8tuzt)

[2. Hướng dẫn cài đặt extension Element Custom trên chrome 71](#_heading=h.2ce457m)

**MỤC LỤC HÌNH**

[Hình 1. Kiến trúc Extension chứa background.html và popup.html (UI page). 8](#_heading=h.tyjcwt)

[Hình 2. Kiến trúc Extension chứa contentscript.js. 8](#_heading=h.3dy6vkm)

[Hình 3. Danh sách các extension 9](#_heading=h.4d34og8)

[Hình 4. Folder Auto\_Login. 10](#_heading=h.26in1rg)

[Hình 5. Phần mềm Visual Studio Code. 11](#_heading=h.lnxbz9)

[Hình 6. Chọn Open folder 11](#_heading=h.35nkun2)

[Hình 7. Chọn Folder Auto\_Login và nhấn select folder. 12](#_heading=h.1ksv4uv)

[Hình 8. Cây thư mục rỗng vừa được select vào. 12](#_heading=h.44sinio)

[Hình 9. Các file cho project. 13](#_heading=h.2jxsxqh)

[Hình 10. Giao diện nút tùy chỉnh và điều khiển chrome. 21](#_heading=h.1ci93xb)

[Hình 11. Giao diện công cụ khác. 21](#_heading=h.3whwml4)

[Hình 12. Giao diện khi nhấn vào tiện ích mở rộng. 22](#_heading=h.2bn6wsx)

[Hình 13. Tải tiện ích đã giải nén. 22](#_heading=h.qsh70q)

[Hình 14. Extension auto login đã được cài. 23](#_heading=h.3as4poj)

[Hình 15. Ghim extension vừa cài trong tiện ích. 23](#_heading=h.1pxezwc)

[Hình 16. Biểu tượng extension cạnh tiện ích. 24](#_heading=h.49x2ik5)

[Hình 17. Giao diện extension auto login. 24](#_heading=h.2p2csry)

[Hình 18. Giao diện trang el.tdc.edu.vn. 25](#_heading=h.147n2zr)

[Hình 19. Nhập thông tin vào extension auto login. 25](#_heading=h.3o7alnk)

[Hình 20. Log in trang el.tdc.edu.vn. 26](#_heading=h.23ckvvd)

[Hình 21. Extension tự điền tài khoản vào form. 26](#_heading=h.ihv636)

[Hình 22. Tài khoản đã được login thành công. 26](#_heading=h.32hioqz)

[Hình 23. Nhập username, password vào và lưu lại 27](#_heading=h.41mghml)

[Hình 24. Extension tự động đăng nhập vào trang admin 27](#_heading=h.2grqrue)

[Hình 25. Folder GET\_cookie 28](#_heading=h.4f1mdlm)

[Hình 26. Phần mềm Visual Studio Code 29](#_heading=h.2u6wntf)

[Hình 27. Chọn Open folder 29](#_heading=h.19c6y18)

[Hình 28. Chọn Folder Auto\_Login và nhấn select folder 30](#_heading=h.3tbugp1)

[Hình 29. Cây thư mục rỗng vừa được select vào 30](#_heading=h.28h4qwu)

[Hình 30. Cấu trúc thư mục extension get cookie 31](#_heading=h.nmf14n)

[Hình 31. Di chuyển vào danh sách extension đã được cài đặt. 36](#_heading=h.111kx3o)

[Hình 32. Danh sách extension đã cài đặt. 36](#_heading=h.3l18frh)

[Hình 33. Kích hoạt chế độ nhà phát triển. 37](#_heading=h.206ipza)

[Hình 34. Chức năng dành cho nhà phát triển. 37](#_heading=h.4k668n3)

[Hình 35. Chọn extension được định nghĩa qua chức năng dành cho nhà phát triển. 38](#_heading=h.2zbgiuw)

[Hình 36. Add thành công extension tự định nghĩa vào chrome. 39](#_heading=h.1egqt2p)

[Hình 37. Ghim extension vào thanh công cụ 40](#_heading=h.3ygebqi)

[Hình 38. Biểu tượng extension cạnh tiện ích 40](#_heading=h.2dlolyb)

[Hình 39. Giao diện extension GET\_cookie 41](#_heading=h.sqyw64)

[Hình 40. Giao diện trang el.tdc.edu.vn 42](#_heading=h.3cqmetx)

[Hình 41. Kết quả GET\_cookie của trang el.tdc.edu.vn 42](#_heading=h.1rvwp1q)

[Hình 42. So sánh cookie từ F12 và Extension. 42](#_heading=h.4bvk7pj)

[Hình 43. Đăng nhập vào trang quản trị bằng tài khoản admin 43](#_heading=h.1664s55)

[Hình 44. Extension đã lấy được phiên cookie hiện tại của trang web 43](#_heading=h.3q5sasy)

[Hình 45. Xem thông tin cookie của trang web bằng F12 kiểm tra source 44](#_heading=h.25b2l0r)

[Hình 46. Folder GET\_cookie 45](#_heading=h.43ky6rz)

[Hình 47. Phần mềm Visual Studio Code 46](#_heading=h.2iq8gzs)

[Hình 48. Chọn Open folder 46](#_heading=h.xvir7l)

[Hình 49. Chọn Folder mouse\_select và nhấn select folder 47](#_heading=h.3hv69ve)

[Hình 50. Cây thư mục rỗng vừa được select vào 47](#_heading=h.1x0gk37)

[Hình 51. Các file cho project 48](#_heading=h.4h042r0)

[Hình 52. Giao diện nút tùy chỉnh và điều khiển chrome 50](#_heading=h.pkwqa1)

[Hình 53. Giao diện công cụ khác 51](#_heading=h.39kk8xu)

[Hình 54. Giao diện khi nhấn vào tiện ích mở rộng 51](#_heading=h.1opuj5n)

[Hình 55. Chọn Folder mouse\_select và nhấn select folder 52](#_heading=h.48pi1tg)

[Hình 56. Extension mouse\_select đã được cài 52](#_heading=h.2nusc19)

[Hình 57. Ghim extension vừa cài trong tiện ích 53](#_heading=h.1302m92)

[Hình 58. Biểu tượng extension cạnh tiện ích 53](#_heading=h.3mzq4wv)

[Hình 59. Giao diện extension mouse\_select 54](#_heading=h.2250f4o)

[Hình 60. Giao diện trang el.tdc.edu.vn 54](#_heading=h.haapch)

[Hình 61. Nhấn Start để bắt đầu 55](#_heading=h.319y80a)

[Hình 62. Hover chuột vào phần muốn xóa 55](#_heading=h.1gf8i83)

[Hình 63. Phần có viền đỏ đã biến mất 55](#_heading=h.40ew0vw)

[Hình 64. Phần được chọn để select 55](#_heading=h.2fk6b3p)

[Hình 65. Kết quả sau khi click chuột 56](#_heading=h.upglbi)

[Hình 66. Giao diện trang login của php-training 56](#_heading=h.1tuee74)

[Hình 67. Nhấn vào button Start 57](#_heading=h.4du1wux)

[Hình 68. Element login đã bị xóa trên tab hiện tại 57](#_heading=h.2szc72q)

[Hình 69. Folder Element\_Custom 58](#_heading=h.meukdy)

[Hình 70. Phần mềm Visual Studio Code 59](#_heading=h.36ei31r)

[Hình 71. Chọn Open folder 59](#_heading=h.1ljsd9k)

[Hình 72. Chọn Folder Element\_Custom và nhấn select folder 60](#_heading=h.45jfvxd)

[Hình 73. Cây thư mục rỗng vừa được select vào 60](#_heading=h.2koq656)

[Hình 74. Các file cho project. 61](#_heading=h.zu0gcz)

[Hình 75. Giao diện nút tùy chỉnh và điều khiển chrome 71](#_heading=h.rjefff)

[Hình 76. Giao diện công cụ khác 71](#_heading=h.3bj1y38)

[Hình 77. Giao diện khi nhấn vào tiện ích mở rộng 72](#_heading=h.1qoc8b1)

[Hình 78. Chọn Folder Element\_Custom và nhấn select folder 72](#_heading=h.4anzqyu)

[Hình 79. Extension Element\_Custom đã được cài 73](#_heading=h.2pta16n)

[Hình 80. Ghim extension vừa cài trong tiện ích 73](#_heading=h.14ykbeg)

[Hình 81. Biểu tượng extension cạnh tiện ích 74](#_heading=h.3oy7u29)

[Hình 82. Giao diện extension Element\_Custom 74](#_heading=h.243i4a2)

[Hình 83. Giao diện trang el.tdc.edu.vn. 75](#_heading=h.j8sehv)

[Hình 84. Giao diện làm việc Element\_Custom. 75](#_heading=h.338fx5o)

[Hình 85. Nút thiết lập. 76](#_heading=h.1idq7dh)

[Hình 86. Giao diện thiết lập 76](#_heading=h.42ddq1a)

# **I. HƯỚNG DẪN CÀI VÀ SỬ DỤNG EXTENSION VỚI CHROME**

## **1. Giới thiệu về extension**

Extensions là những tiện ích mở rộng nhằm giúp bổ sung những tính năng hữu ích cho trình duyệt.

Extension sẽ tập các file như: HTML, CSS, Javascript, Images,….. được nén lại trong một thư mục.

Về cơ bản thì nó là những web pages, chúng có thể sử dụng tất cả các APIs mà trình duyệt cung cấp, từ XML Http Request tới JSON, HTML5.

**Mỗi extension về cơ bản có những file sau**:

+ Một file manifest

+ Một hoặc nhiều file HTML

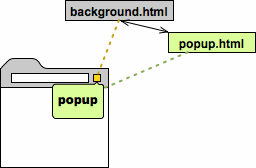
+ Optional: Một hoặc nhiều file Javascript

+ Optional: Bất kì file gì mà bạn cần, ví dụ: images,…

## **2. Kiến trúc của extension**

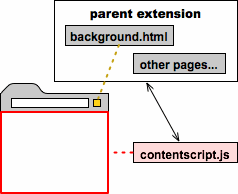
**Background pages**: được định nghĩa bởi **background.html** có thể bao gồm code Javascript để điều khiển các sự kiện của extension.

**UI page**: Extension có thể chứa những trang HTML thông thường nhằm hiển thị UI của extension.



**Hình 1. Kiến trúc Extension chứa background.html và popup.html (UI page).**

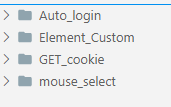
**Content script**: các bạn chỉ có thể lấy thông tin, thay đổi, thêm mới các element trên trang web với content script.



**Hình 2. Kiến trúc Extension chứa contentscript.js.**

## **3. Các extension hiện có**

* Auto login
* GET\_cookie
* Mouse select
* Element custom



**Hình 3. Danh sách các extension**

# **II. HƯỚNG DẪN PHÁT TRIỂN EXTENSION AUTO LOGIN**

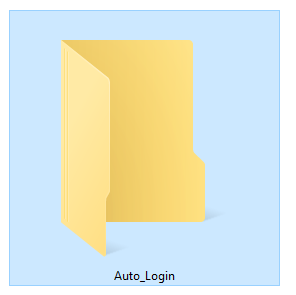
Auto Login là tiện ích hỗ trợ người dùng dễ dàng lập cấu hình chế độ đăng nhập tự động. Thay vì đợi một user (người dùng) nhập tên và mật khẩu, Extension Auto Login sẽ sử dụng tài khoản bạn nhập và lưu trước đó để sau này mỗi lần truy cập vào website đó sẽ đăng nhập cách tự động giúp làm tiết kiệm thời gian của người dùng.

## **1. Quy Trình Phát Triển**

### **1.1. Tạo project mới cho extension Auto Login**

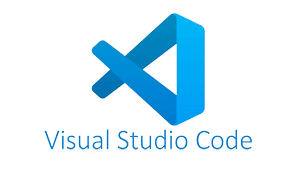
Tạo một cây thư mục và các file cần thiết của project.

Vào ổ đĩa bất kì của máy và tạo một folder có tên: **Auto\_Login**



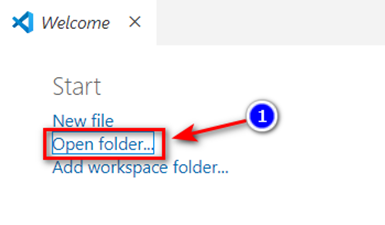
**Hình 4. Folder Auto\_Login.**

Sử dụng phần mền Visual Studio Code để tạo cây thư mục cho project.



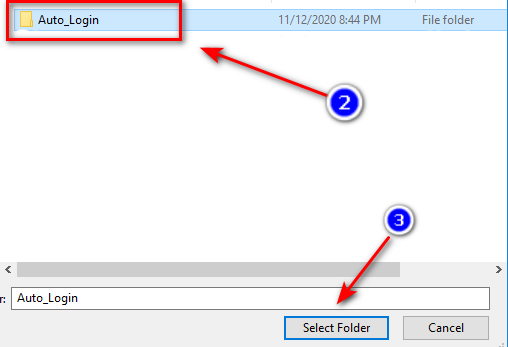
**Hình 5. Phần mềm Visual Studio Code.**

Sau khi khởi động phần mềm xong. Chọn vào mục **Open folder**.



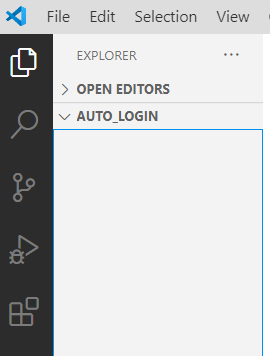
**Hình 6. Chọn Open folder**

Chọn đến folder **Auto\_Login** vừa mới tạo bên trên và nhấn **Select Folder**:



**Hình 7. Chọn Folder Auto\_Login và nhấn select folder.**

Sau khi select folder chúng ta sẽ có một cây thư mục rỗng như sau:



**Hình 8. Cây thư mục rỗng vừa được select vào.**

Các file dành cho project **Auto\_Login** bao gồm:

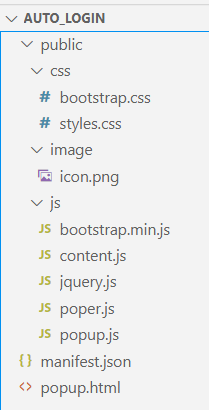
+ popup.html

+ manifest.json

+ folder public chứa các file css, images, js

+ popup.js

+ content.js



**Hình 9. Các file cho project.**

### **1.2. Xây dựng file manifest.json**

Đây là phần code dùng để đăng ký thông tin cho extension **Auto\_Login** trên chrome:

{

    "manifest\_version": 2,

    "name": "Auto Logged",

    "description": "Tự động đăng nhập",

    "version": "1.0",

    "permissions": ["tabs", "\*://\*/\*", "storage"],

    "content\_scripts": [{

        "matches": [

            "<all\_urls>"

        ],

        "js": ["public/js/jquery.js" , "public/js/content.js"]

    }],

    "browser\_action": {

        "default\_title": "Auto Logged",

        "default\_popup": "popup.html",

        "default\_icon": "public/image/icon.png"

    }

}

**Chú thích:**

|  |  |
| --- | --- |
| "manifest\_version" | Phiên bản hỗ trợ của chrome, hiện tại là 2.0. |
| "name" | Tên của ứng dụng. |
| "description" | Mô tả về ứng dụng. |
| "version" | Phiên bản của ứng dụng |
| "permissions" | Cấp quyền truy cập của ứng dụng đối với trình duyệt. |
| "options\_ui" | "page": Link dẫn tới trang thiết lập.  "open\_in\_tab": Mở tab mới nếu true. |
| "content\_scripts" | Mục Javascript. |
| "browser\_action" | Những mục được hiển thị và mở khi người dùng click vào biểu tượng của Extension. |

### **1.3. Xây dựng giao diện trong file popup.html**

Giao diện này sẽ được hiện ra khi ta click vào button của extension **Auto\_Login** trên chrome:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8">

    <link rel="stylesheet" type="text/css" href="public/css/bootstrap.css">

    <link rel="stylesheet" type="text/css" href="public/css/styles.css">

    <script src="public/js/jquery.js"></script>

  </head>

  <body>

    <header>

      <h3>Auto Logged</h3>

    </header>

    <div class="container">

      <div class="input-group mb-3 input-group-sm">

        <div class="input-group-prepend">

          <span class="input-group-text">Địa Chỉ</span>

        </div>

        <input id='CurrentCookieUrl' type="text" class="form-control">

      </div>

      <div class="input-group mb-3">

        <div class="input-group-prepend">

          <span class="input-group-text" id="username">Username</span>

        </div>

        <input type="text" id="inputuser" class="form-control" aria-label="Default"  aria-describedby="inputGroup-sizing-default">

      </div>

      <div class="input-group mb-3">

        <div class="input-group-prepend">

          <span class="input-group-text" id="password">Password&nbsp;</span>

        </div>

        <input type="password" id="inputpassword" class="form-control" aria-label="Default" aria-describedby="inputGroup-sizing-default">

      </div>

      <form>

        <div class="form-group">

          <label for="selectuser">List User</label>

          <select multiple class="form-control" id="selectuser">

          </select>

        </div>

      </form>

      <div class="custom-control custom-checkbox">

        <input type="checkbox" id="cbx-auto-login" class="custom-control-input">

        <label class="custom-control-label" for="cbx-auto-login">Tự động đăng nhập.</label>

      </div>

      <button id="save" class="btn btn-outline-success" type="button">Save</button>

      <button id="login" class="btn btn-outline-warning" type="button">Login</button>

      <button id="delete" class="btn btn-outline-danger" type="button" disabled="true">Delete</button>

    </div>

    <script src="public/js/popup.js"></script>

    <script src="public/js/poper.js"></script>

    <script src="public/js/bootstrap.min.js"></script>

  </body>

</html>

### **1.4. Xây dựng chức năng trong file popup.js**

var currentCookie="";

var currentUid = "";

$(document).ready(function () {

/\* xử lý sự kiện click vào mục cài đặt \*/

    $('#go-to-options').click(function() {

        if (chrome.runtime.openOptionsPage) {

        chrome.runtime.openOptionsPage();

        } else {

        window.open(chrome.runtime.getURL('options.html'));

        }

    });

/\*--------------\*/

/\* khởi tạo bộ lưu trữ nếu mở lần đầu\*/

    chrome.storage.sync.get(["userselected", "passselected"], function(active) {

        $('#inputuser').val(active["userselected"]);

        $('#inputpassword').val(active["passselected"]);

    });

    chrome.storage.sync.get(["active"], function(active) {

        if (!active['active']) {

            user = {"linkURL" : [],

                    "username": [],

                    "password": []};

            chrome.storage.sync.set({

            "list": user

            }, function() {

                console.log("active finish!");

            });

            chrome.storage.sync.set({

            "active": true

            });

        }

    });

/\*----------------\*/

/\* lấy danh sách user ở mỗi trang khác nhau \*/

get\_users\_page();

    function get\_users\_page() {

        $("#selectuser").children().remove();

        chrome.storage.sync.get(["list"], function(result) {

            var list = result['list'] || '';

            if (list != '') {

                var list\_url = list['linkURL'] || '';

                var list\_user = list['username'] || '';

                var url\_page = $("#CurrentCookieUrl").val();

                if (list\_user != '') {

                    for (var i = 0; i < list\_user.length; i++) {

                        if (list\_url[i] == url\_page) {

                    $("#selectuser").append('<option>'+list\_user[i]+'</option>');

                        }

                    }

                }

            }

        });

    }

/\*------------------\*/

/\* lấy thông tin user khi click vào từng option của form select.\*/

$("select#selectuser").change(function(){

      var selected = $(this).children("option:selected").val();

      var url\_page = $("#CurrentCookieUrl").val();

      $('#delete').removeAttr("disabled");

      // alert("You have selected - " + selected);

      chrome.storage.sync.get(["list"], function(result) {

            var list\_url = result['list']['linkURL'] || '';

            var list\_user = result['list']['username'] || '';

            var list\_pass = result['list']['password'] || '';

             if (list\_url != '') {

                for (var i = 0; i < list\_url.length; i++) {

                    if (url\_page == list\_url[i]) {

                        if (selected == list\_user[i]) {

                            $('#inputuser').val(list\_user[i]);

                            $('#inputpassword').val(list\_pass[i]);

                            chrome.storage.sync.set({

                            "userselected": list\_user[i],

                            "passselected": list\_pass[i]

                            }, function() {

                                console.log("save!");

                            });

                        }

                    }

                }

            }

        });

    });

/\*----------\*/

/\* Xử lý sự kiện click nút save lưu thông tin tài khoản.\*/

 $('#save').click(function () {

        var get\_input\_user = $("#inputuser").val();

        var get\_input\_pass = $("#inputpassword").val();

        var get\_input\_url = $("#CurrentCookieUrl").val();

        chrome.storage.sync.get(["list"], function(result) {

            var get\_list = result['list'];

            var check = true;

            var list\_url = get\_list['linkURL'] || '';

            var list\_user = get\_list['username']|| '';

            var list\_pass = get\_list['password'] || '';

            for (var i = 0; i < list\_user.length; i++) {

            if (list\_url[i] == get\_input\_url && list\_user[i] == get\_input\_user) {

                    check = false;

                    console.log(list\_pass[i]);

                    console.log(get\_input\_pass);

                    if (list\_pass[i] != get\_input\_pass) {

                        alert("Chức năng đổi pass chưa ra mắt!"+"\nPass cũ: "+ list\_pass[i] +"\nPass mới: " + get\_input\_pass + '\nTại vị trí: ' + i);

                    }

                }

            }

            if (check && get\_input\_user != '') {

                get\_list['linkURL'].push(get\_input\_url);

                get\_list['username'].push(get\_input\_user);

                get\_list['password'].push(get\_input\_pass);

                chrome.storage.sync.set({

                "list": get\_list

                }, function() {

                    console.log("save!");

                    get\_users\_page();

                });

            }

        });

    });

/\*---------------\*/

/\* sự kiện click đăng nhập \*/

$("#login").click(function () {

        chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {

            var activeTab = tabs[0];

            chrome.tabs.sendMessage(activeTab.id, {"message": "start"});

        });

    });

/\*--------------\*/

/\* chức năng auto login.\*/

    chrome.storage.sync.get(["auto-login"], function(checkbox) {

        if (checkbox["auto-login"]) {

            $('#cbx-auto-login').attr('checked', true);

            $('#login').attr('disabled', true);

            chrome.storage.sync.set({

                "auto-login": true

            });

        }

        else{

            $('#cbx-auto-login').attr('checked', false);

            $('#login').attr('disabled', false);

            chrome.storage.sync.set({

                "auto-login": false

            });

        }

    });

    $('#cbx-auto-login').click(function() {

        if ($(this).is(':checked')) {

            $('#login').attr('disabled', true);

            chrome.storage.sync.set({

                "auto-login": true

                }, function() {

                    console.log("auto login is on");

            });

        } else{

            $('#login').attr('disabled', false);

            chrome.storage.sync.set({

                "auto-login": false

                }, function() {

                    console.log("auto login is off");

            });

        }

    });

}); //end document ready

/\*---------------------\*/

/\* lấy url hiện tại và gọi function lọc url rồi add vô frontend.\*/

function loadCurrentCookie() {

    chrome.tabs.getSelected(null, function (tab) { //lấy cửa sổ làm việc hiện tại

        var currentUrl=tab.url;

        //add địa chỉ url đã lọc

        $('#CurrentCookieUrl').val(extractHostname(currentUrl));

    });

}

/\*-----------------\*/

/\* function lọc url.\*/

function extractHostname(url) {

    var hostname;

    //lọc url của web

    if (url.indexOf("://") > -1) {

        hostname = url.split('/')[2];

    }

    else {

        hostname = url.split('/')[0];

    }

    //tìm và xóa địa chỉ port

    hostname = hostname.split(':')[0];

    //tìm và xóa "?"

    hostname = hostname.split('?')[0];

    return hostname;

}

//gọi chức năng load loadCurrentCookie()

loadCurrentCookie();

/\*-----------\*/

### **1.5. Xây dựng chức năng trong file content.js**

Ta xây dựng hàm này để lấy dữ liệu của user và tự động add vào input field của trang sau đó thực hiện submit chức năng đăng nhập.

$(document).ready(function () {

    chrome.storage.sync.get(["auto-login"], function (checkbox) {

      if (checkbox["auto-login"]) {

        login();

      }

    });

    chrome.runtime.onMessage.addListener(function (request) {

      if (request.message === "start") {

        login();

      }

    });

  });

function login() {

    chrome.storage.sync.get(["userselected", "passselected"], function(result) {

        $('input[type=text]').val(result["userselected"]);

        $('input[type=password]').val(result["passselected"]);

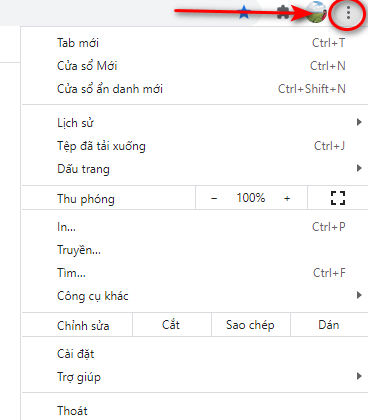
        $('button[type=submit]').click();

    });

};

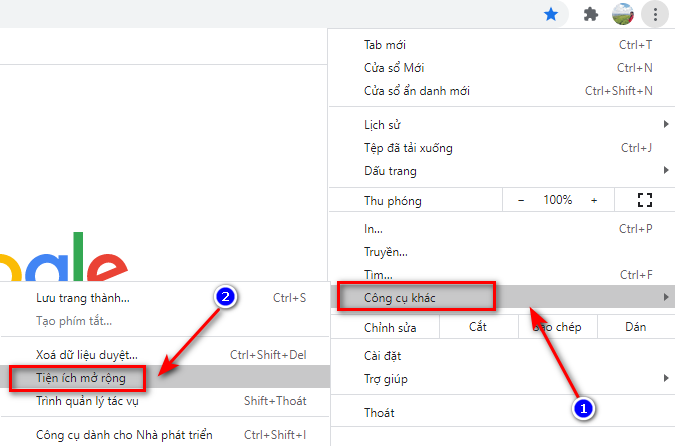
## **2. Hướng dẫn cài đặt extension Auto Login trên chrome**

**Bước 1.** Các bạn mở chrome và nhấn vào nút tùy chỉnh và điều khiển chrome bên góc phải màn hình:



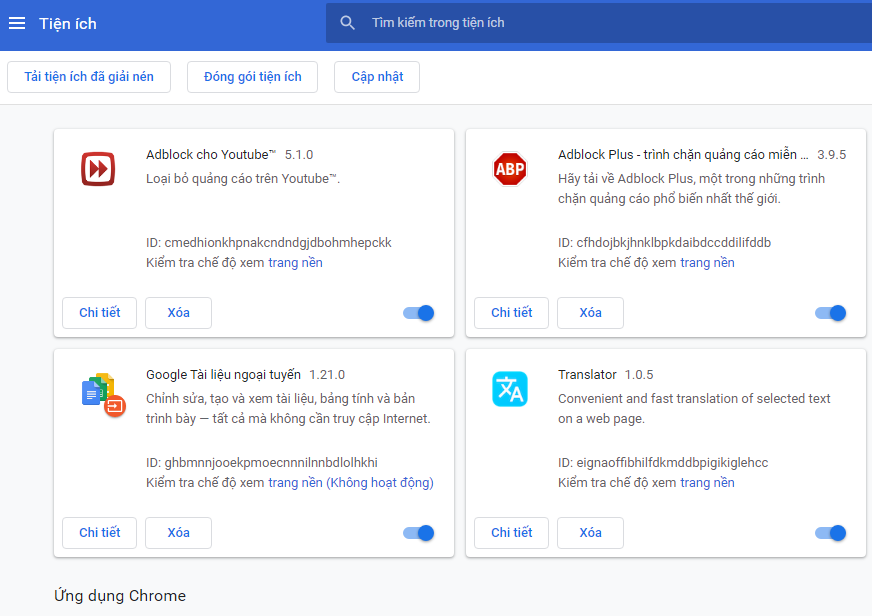
**Hình 10. Giao diện nút tùy chỉnh và điều khiển chrome.**

**Bước 2.** Tiếp theo chọn **Công cụ khác** -> **Tiện ích mở rộng**



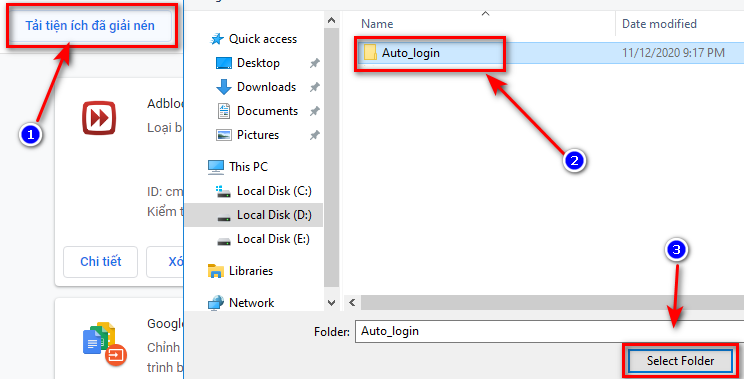
**Hình 11. Giao diện công cụ khác.**

Sau khi nhấn vào tiện ích mở rộng bạn sẽ được giao diện như sau:



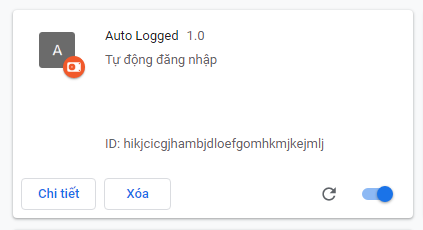
**Hình 12. Giao diện khi nhấn vào tiện ích mở rộng.**

**Bước 3.** Tiếp theo nhấn vào mục: **Tải tiện ích đã giải nén** -> chọn file **Auto\_login** -> chọn **Select Folder**



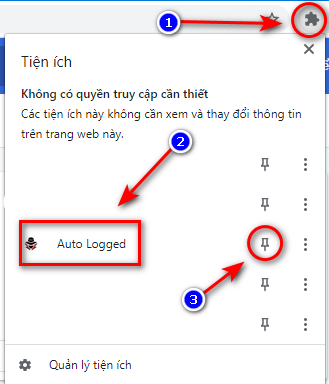
**Hình 13. Tải tiện ích đã giải nén.**

Sau khi tải lên xong ta sẽ có một extension như hình:



**Hình 14. Extension auto login đã được cài.**

**Bước 4.** Tiếp theo các bạn nhấn vào biểu tượng: **Tiện ích** -> chọn đến extension vừa được cài -> nhấn vào biểu tượng nút ghim.



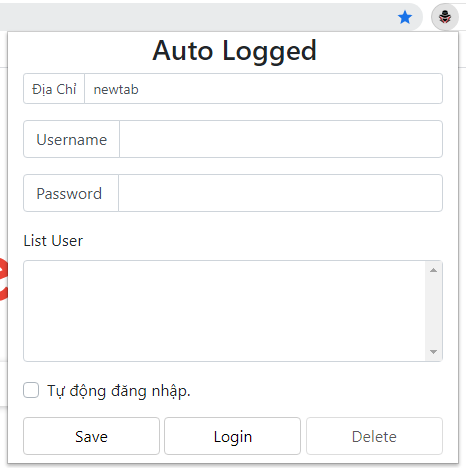
**Hình 15. Ghim extension vừa cài trong tiện ích.**

Sau khi ghim extension sẽ hiển thị bên ngoài kế bên tiện ích. Nhằm mục đích cho người dùng dễ sử dụng:



**Hình 16. Biểu tượng extension cạnh tiện ích.**

Giao diện làm việc của extension auto login:



**Hình 17. Giao diện extension auto login.**

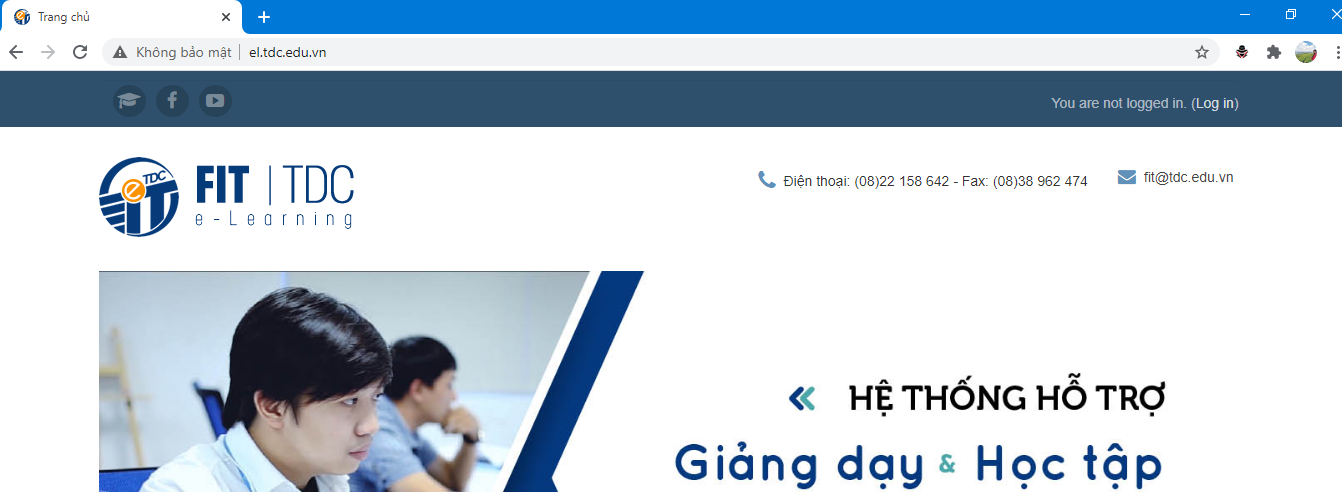
**Trong đó:**

* Địa chỉ: là địa chỉ URL của trang hiện tại.
* Username: là tài khoản user mà bạn muốn dùng đăng nhập.
* Password: là mật khẩu của tài khoản user bên trên.
* List user: là danh sách các tài khoản được lưu.

**Demo**: Áp dụng vào các trang web như: trang el.tdc.edu.vn, online.tdc.edu.vn,…

Áp dụng vào trang **el.tdc.edu.vn**

Các bạn truy cập vào trang <http://el.tdc.edu.vn/> sẽ thấy giao diện như sau:



**Hình 18. Giao diện trang el.tdc.edu.vn.**

Sau khi vào được giao diện như hình trên, tiếp theo nhấn vào icon extension **Auto\_Login** chúng ta đã cài và điền các thông tin đăng nhập vào giao diện của extension **Auto\_Login**, sau đó nhấn save và click vào nút “**Tự động đăng nhập**”.



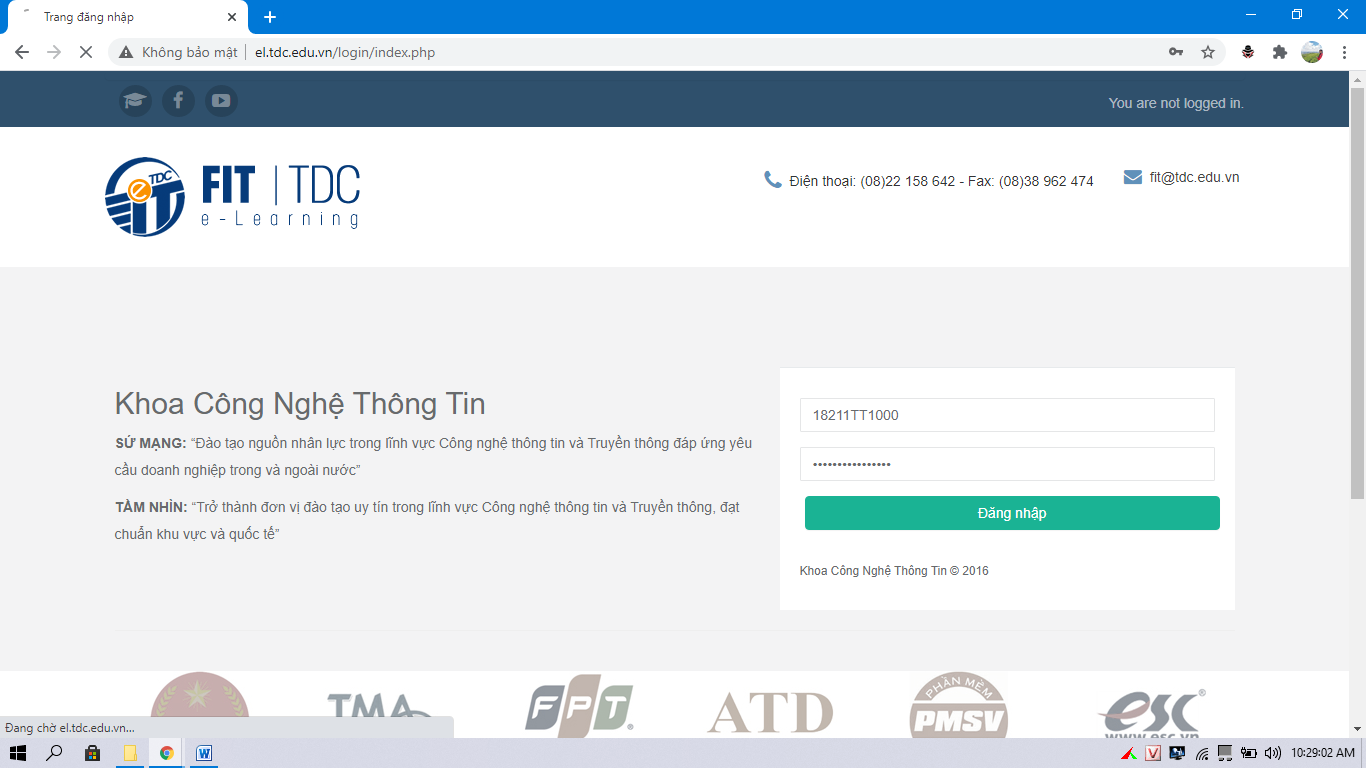
**Hình 19. Nhập thông tin vào extension auto login.**

Sau khi thực hiện các bước trên xong bạn chỉ cần nhấn vào Login ở giao diện của trang **el.tdc.edu.vn** và không cần nhập bất cứ gì vì extension **Auto\_Login** đã thực hiện việc tự động đăng nhập thông qua tài khoản chúng ta đã nhập và lưu trước đó.



**Hình 20. Log in trang el.tdc.edu.vn.**

Extension auto login sẽ tự điền các thông tin tài khoản mà bạn đã lưu khi nãy vào form đăng nhập.



**Hình 21. Extension tự điền tài khoản vào form.**

Và kết quả sau khi để extension **Auto\_Login** đăng nhập.



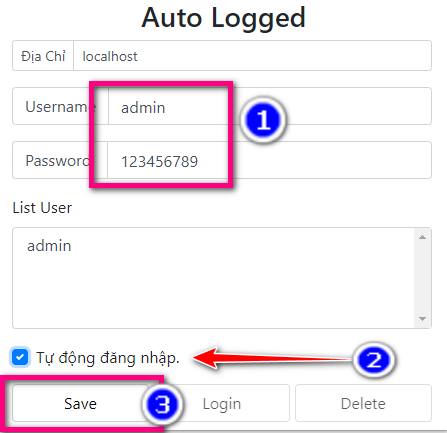
**Hình 22. Tài khoản đã được login thành công.**

## **3. Áp dụng extension vào source code php-training**

Truy cập vào địa chỉ <http://localhost/php-training/login.php>

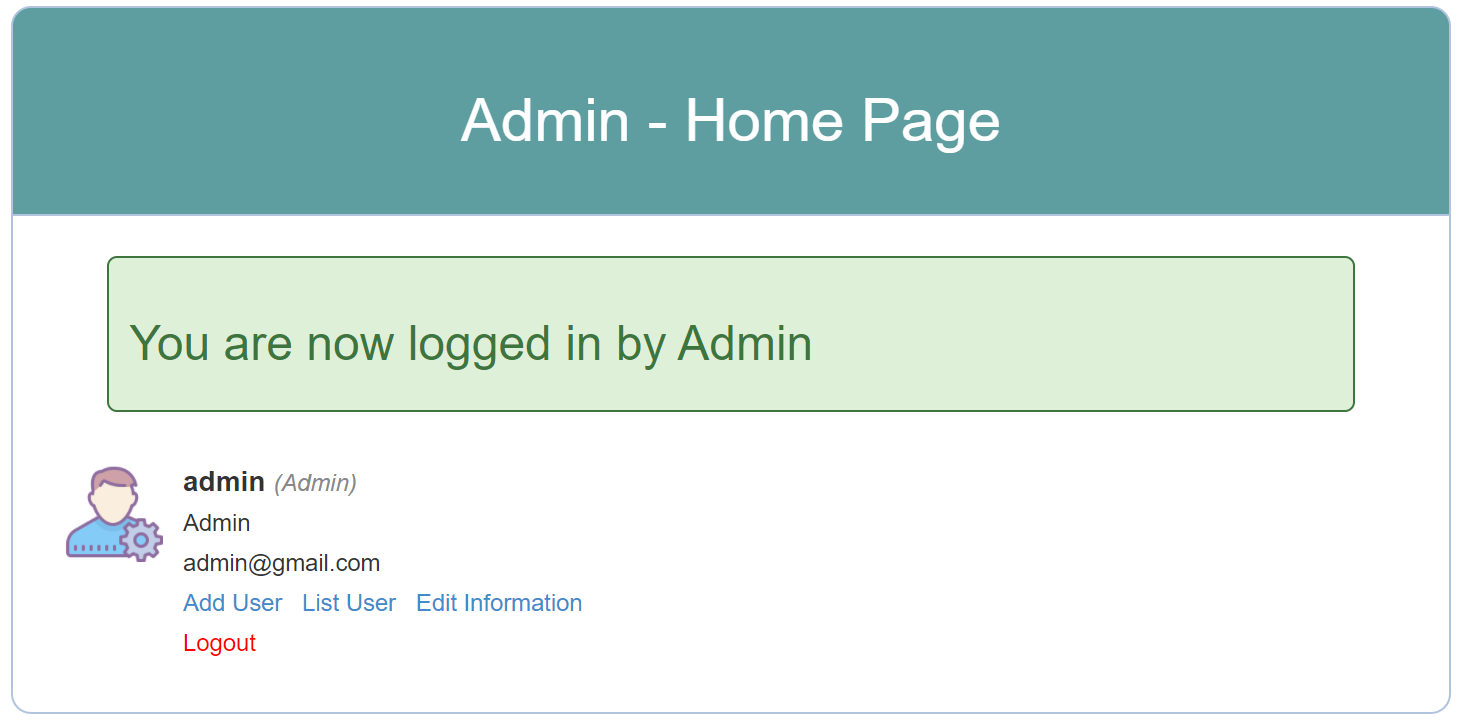
Sau đó ta click vào icon của extension **Auto Login** lên và nhập username, password đăng nhập vào ô và nhấn **Save**.

Khi này, nếu bạn muốn tự động đăng nhập vào trang web mỗi khi mở lên thì hãy nhấn vào “**Tự động đăng nhập**” và lưu lại.



**Hình 23. Nhập username, password vào và lưu lại**

Sau khi đã lưu lại, Bạn hãy tải lại trang, lập tức extension sẽ tự động đăng nhập vào trang web quản trị của **php-training** với tài khoản bạn đã lưu trước đó.



**Hình 24. Extension tự động đăng nhập vào trang admin**

# **III. HƯỚNG DẪN PHÁT TRIỂN EXTENSION GET COOKIE**

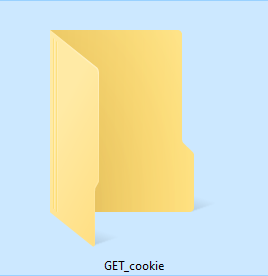
File tạm cookie sẽ được tự động tạo ra trong máy tính của bạn mỗi khi bạn truy cập một trang Web nào đó, nó sẽ lưu những thông tin cá nhân của bạn như thiết bị bạn đang sử dụng, tài khoản cá nhân, phiên làm việc hiện tại, v.v…

Extension **Get Cookie** là một tiện ích hỗ trợ giúp người dùng lấy được thông tin phiên cookie hiện tại của người dùng.

## **1. Quy trình phát triển**

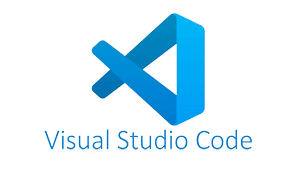
### **1.1. Tạo project mới cho extension Get Cookie:**

Vào ổ đĩa D của máy và tạo một folder có tên: **GET\_cookie**



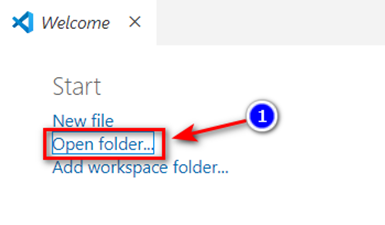
**Hình 25. Folder GET\_cookie**

Sử dụng phần mềm Visual Studio Code để tạo cây thư mục cho project.



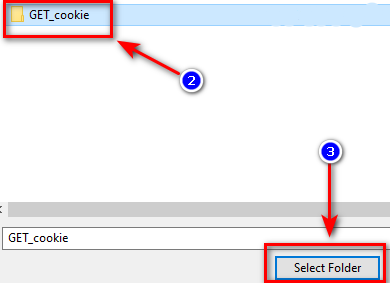
**Hình 26. Phần mềm Visual Studio Code**

Sao khi khởi động phần mềm xong. Chọn vào mục **Open folder**



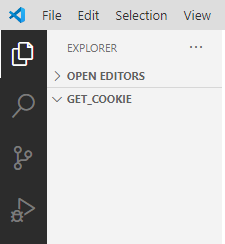
**Hình 27. Chọn Open folder**

Chọn đến folder **GET\_cookie** vừa mới tạo bên trên và nhấn **Select Folder**:



**Hình 28. Chọn Folder Auto\_Login và nhấn select folder**

Sau khi select folder chúng ta sẽ có một cây thư mục rỗng như sau:



**Hình 29. Cây thư mục rỗng vừa được select vào**

Các file dành cho project **GET\_cookie** bao gồm:

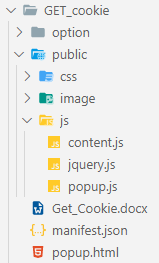
+ popup.html

+ manifest.json

+ folder public chứa các file css, images, js

+ popup.js

+ content.js



**Hình 30. Cấu trúc thư mục extension get cookie**

Sau khi đã tạo cấu trúc như hình trên sẽ tiến hành cấu hình cho file **mainifest.json.**

### **1.2. Xây dựng file mainifest.json**

Đây là phần code dùng để đăng ký thông tin cho extension **Auto\_Login** trên chrome:

{

    "manifest\_version": 2,

    "name": "Cookie",

    "description": "Get Cookie",

    "version": "1.0",

    "options\_ui": {

        "page": "option/options.html",

        "open\_in\_tab": true

    },

    "permissions": ["cookies", "tabs", "\*://\*/\*", "storage"],

    "content\_scripts": [{

        "matches": [

            "<all\_urls>"

        ],

        "js": ["public/js/jquery.js" , "public/js/content.js"]

    }],

    "browser\_action": {

        "default\_title": "Get Cookie",

        "default\_popup": "popup.html",

        "default\_icon": "public/image/icon.png"

    }

}

**Giải thích:**

|  |  |
| --- | --- |
| "manifest\_version" | Phiên bản hỗ trợ của chrome, hiện tại là 2.0. |
| "name" | Tên của ứng dụng. |
| "description" | Mô tả về ứng dụng. |
| "version" | Phiên bản của ứng dụng(dùng để cập nhật ứng dụng khi đưa lên kho ứng dụng tiện ích của google). |
| "permissions" | Cấp quyền truy cập của ứng dụng đối với trình duyệt. |
| "options\_ui" | "page": Link dẫn tới trang thiết lập.  "open\_in\_tab": Mở tab mới nếu true. |
| "content\_scripts" | Mục Javascript. |
| "browser\_action" | Những mục được hiển thị và mở khi người dùng click vào biểu tượng của Extension. |

### **1.3. Xây dựng giao diện tại file popup.html**

Giao diện này sẽ được hiện ra khi ta click vào button của extension **Get cookie** trên chrome:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="public/css/bootstrap.css">

  <link rel="stylesheet" type="text/css" href="public/css/styles.css">

  <script src="public/js/jquery.js"></script>

</head>

<body>

  <header>

    <h3>Get Cookie</h3>

    <button id="go-to-options" class="btn">⚙</button>

  </header>

  <div class="container">

    <div class="input-group mb-3 input-group-sm">

      <div class="input-group-prepend">

        <span class="input-group-text">Địa Chỉ</span>

      </div>

    <input id='CurrentCookieUrl' type="text" class="form-control">

    </div>

    <textarea id="cookieresult" class = "form-control" rows = "10" placeholder = "Không có gì để hiển thị!"></textarea>

  </div>

<script src="public/js/popup.js"></script>

</body>

</html>

### **1.4. Xây dựng chức năng trong file popup.js**

Đoạn code này sẽ truy cập trang cài đặt và kiểm tra tạo vùng nhớ:

var currentCookie = "";

var currentUid = "";

$(document).ready(function () {

  $("#go-to-options").click(function () {

    if (chrome.runtime.openOptionsPage) {

      chrome.runtime.openOptionsPage();

    } else {

      window.open(chrome.runtime.getURL("options.html"));

    }

  });

  //check active

  chrome.storage.sync.get(["active"], function (active) {

    if (!active["active"]) {

      urls = { urls: [], cookie: [] };

      chrome.storage.sync.set({

          list\_urls: urls,

        },

        function () {

          console.log("active finish!");

        }

      );

      chrome.storage.sync.set({

        active: true,

      });

    }

  });

});

Tiếp theo là đoạn code lấy cookie ở trang hiện tại:

function loadCurrentCookie() {

  chrome.tabs.getSelected(null, function (tab) {

    //lấy cửa sổ làm việc hiện tại

    var currentUrl = tab.url;

    //add địa chỉ url đã lọc

    $("#CurrentCookieUrl").val(extractHostname(currentUrl));

    // lấy cookie đã lưu tùy theo url hiện tại

    chrome.cookies.getAll({ url: currentUrl }, function (cookie) {

      var result = "";

      for (var i = 0; i < cookie.length; i++) {

        result += cookie[i].name + "=" + cookie[i].value + ";\n";

        if (cookie[i].name == "c\_user") {

          currentUid = cookie[i].value;

        }

      }

      $("#cookieresult").val(result);

    });

  }

Và cuối cùng là đoạn code lọc ra url của trang web:

function extractHostname(url) {

  var hostname;

  //lọc url của web

  if (url.indexOf("://") > -1) {

    hostname = url.split("/")[2];

  } else {

    hostname = url.split("/")[0];  }

  //tìm và xóa địa chỉ port

  hostname = hostname.split(":")[0];

  //tìm và xóa "?"

  hostname = hostname.split("?")[0];

  return hostname; }

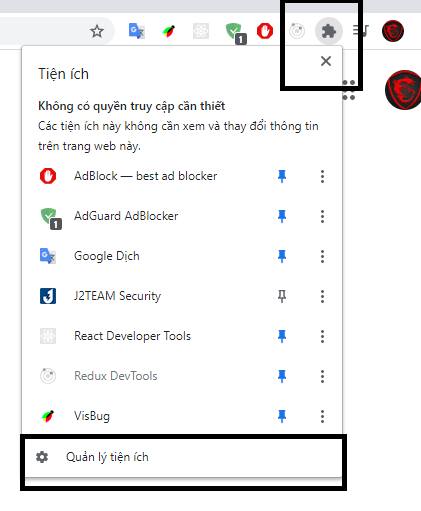
//gọi chức năng load loadCurrentCookie()

loadCurrentCookie();

## **2. Hướng dẫn cài đặt extension Get cookie trên chrome**

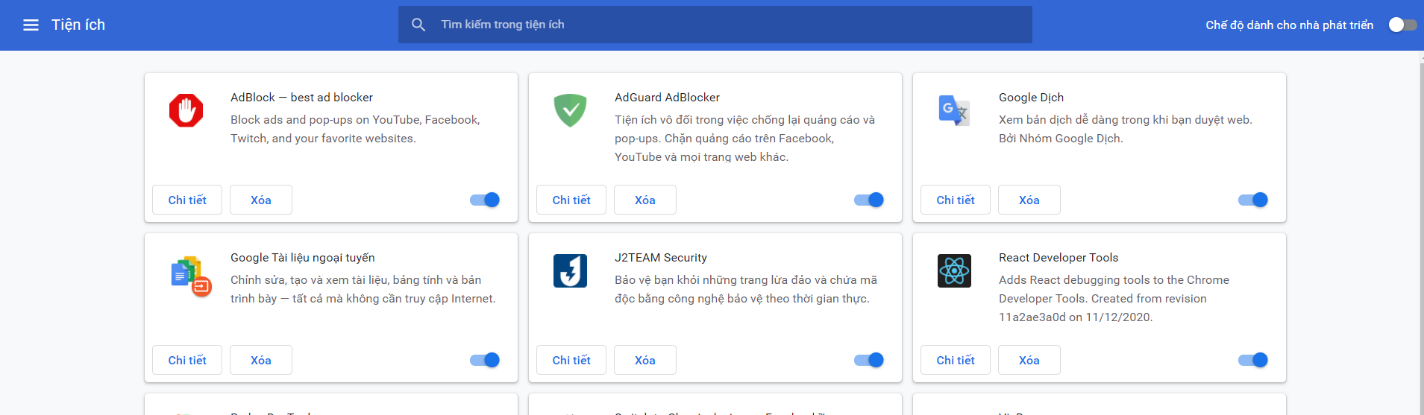
**Bước 1:** Khởi động trình duyệt chrome ấn vào icon extension như hình dưới đây mở danh sách extension đã được cài đặt vào chrome hoặc truy cập liên kết:

**chrome://extensions/**



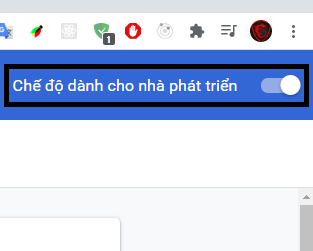
**Hình 31. Di chuyển vào danh sách extension đã được cài đặt.**

**Bước 2:** Sau khi truy cập vào danh sách extension đã được cài đặt sẽ nhận được giao diện với các extension được cài đặt như hình dưới đây.



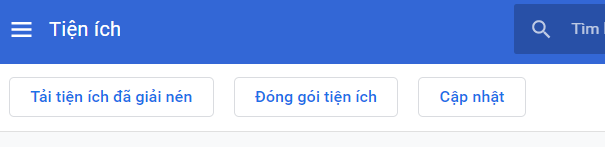
**Hình 32. Danh sách extension đã cài đặt.**

**Bước 3:** Để có thể thêm một extension tự định nghĩa vào chrome thì việc kích hoạt “**Chế độ dành cho nhà phát triển**” như hình dưới đây là điều bắt buộc.



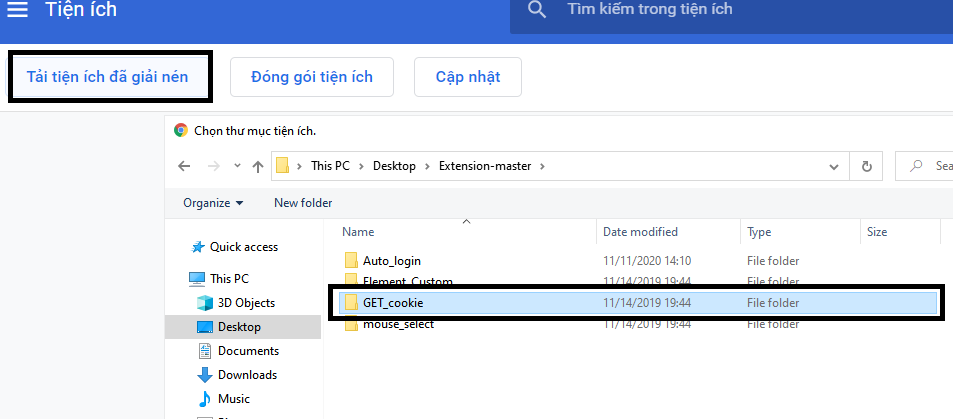
**Hình 33. Kích hoạt chế độ nhà phát triển.**

**Bước 4:** Sau khi “**Chế độ dành cho nhà phát triển**” được kích hoạt, ta sẽ nhận được ba nút chức năng như hình:



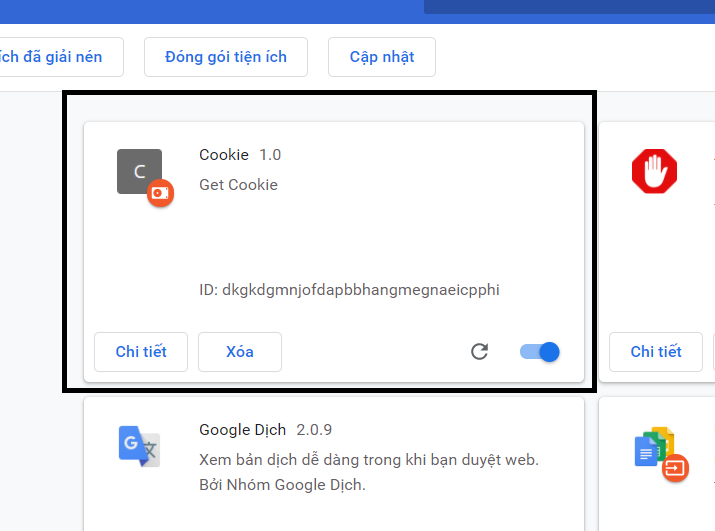
**Hình 34. Chức năng dành cho nhà phát triển.**

**Bước 5:** Click vào nút “**Tải tiện ích đã giải nén**” dành cho nhà phát triển và chọn extension đã được định nghĩa để add vào chrome.



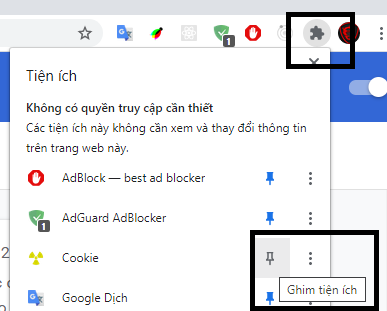
**Hình 35. Chọn extension được định nghĩa qua chức năng dành cho nhà phát triển.**

**Bước 6:** Sau khi add thành công sẽ nhận được giao diện với extension đã được thêm như hình dưới đây.



**Hình 36. Add thành công extension tự định nghĩa vào chrome.**

**Bước 7:** Sau khi add xong, ta cần phải vào lại icon biểu tượng extension và tiến hành sử dụng extension để get cookie của các trang web khác, để có thể thuận tiện trong việc dùng extension cần phải thêm một thao tác ghim extension như hình dưới đây.



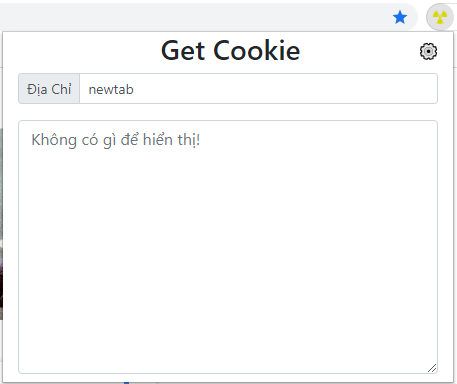
**Hình 37. Ghim extension vào thanh công cụ**

Sau khi ghim extension sẽ hiển thị bên ngoài kế bên tiện ích. Nhằm mục đích cho người dùng dễ sử dụng:



**Hình 38. Biểu tượng extension cạnh tiện ích**

Giao diện hiển thị của extension **GET\_cookie**:



**Hình 39. Giao diện extension GET\_cookie**

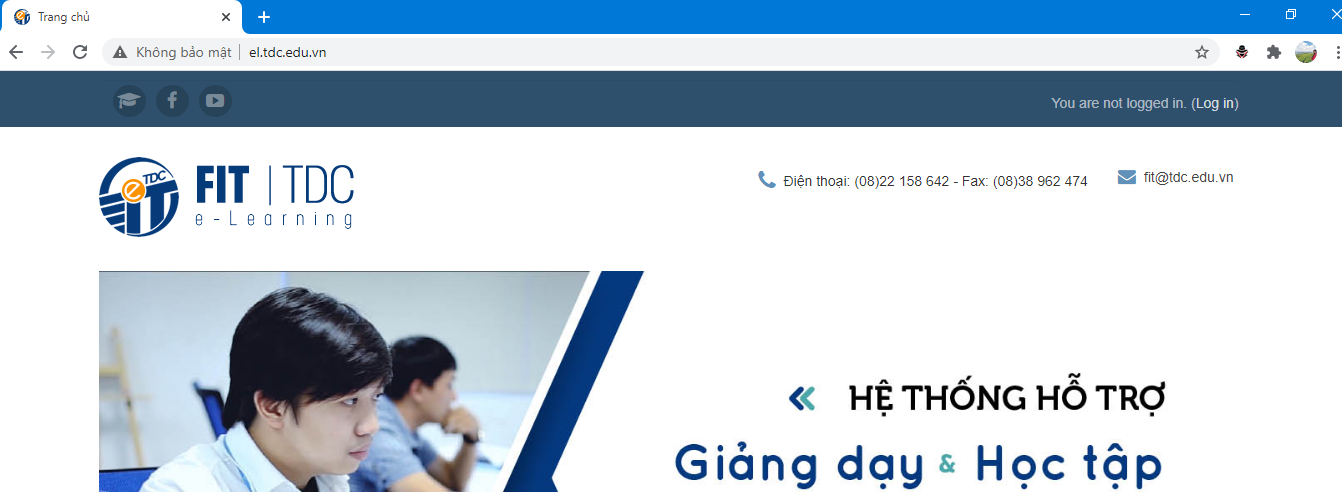
**Trong đó:**

+ Địa chỉ: là địa chỉ URL hiện tại.

+ Textarea: hiển thị cookie trang hiện tại.

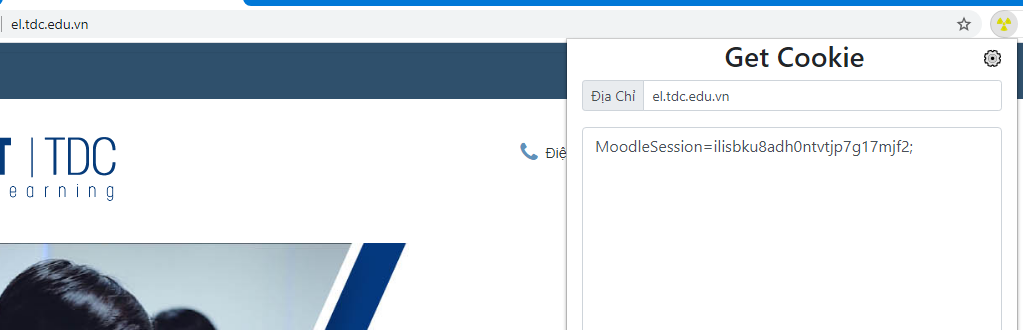
**Demo**: Áp dụng vào trang **el.tdc.edu.vn**

Các bạn truy cập vào trang <http://el.tdc.edu.vn/> sẽ thấy giao diện như sau:



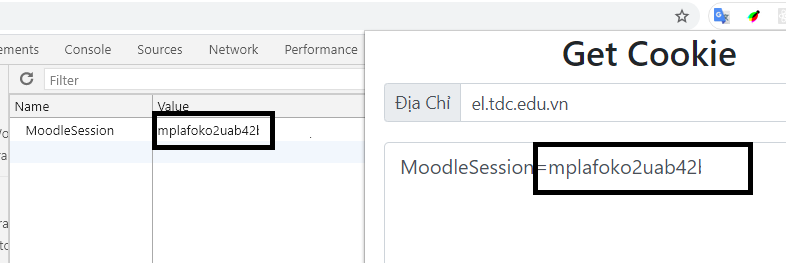
**Hình 40. Giao diện trang el.tdc.edu.vn**

Sau khi vào được giao diện như hình trên, ta nhấn vào icon extension **GET\_cookie** chúng ta đã cài và kết quả sẽ trả về như sau:



**Hình 41. Kết quả GET\_cookie của trang el.tdc.edu.vn**

Đây là ví dụ thử kiểm tra và đối chiếu cookie từ trình duyệt thông qua F12 và extension Get Cookie lấy được trên trang web **el.tdc.edu.vn**

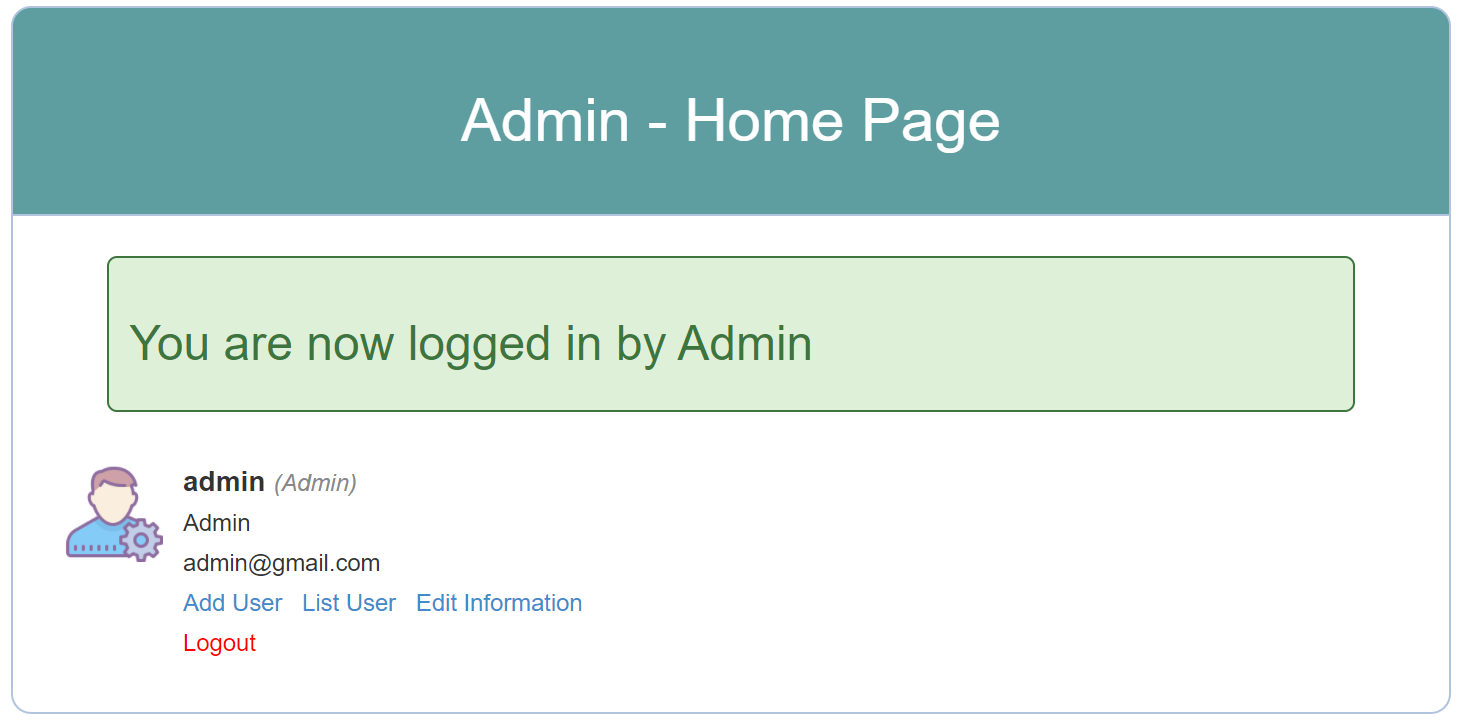


**Hình 42. So sánh cookie từ F12 và Extension.**

## **3. Áp dụng extension vào source code php-training**

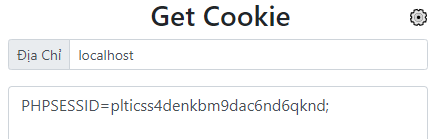
Truy cập vào địa chỉ <http://localhost/php-training/login.php>

Sau đó bạn đăng nhập vào tài khoản admin quản trị viên.



**Hình 43. Đăng nhập vào trang quản trị bằng tài khoản admin**

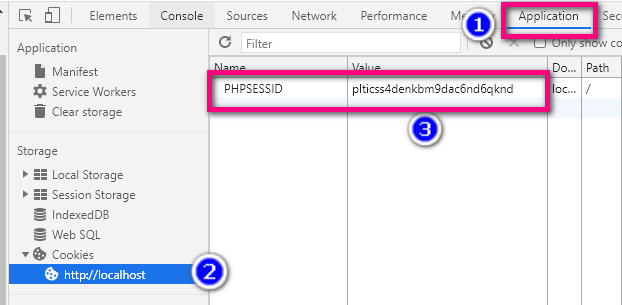
Tiếp theo chúng ta click vào icon của extension **Get Cookie** lên. Chúng ta sẽ thấy được thông tin cookie của phiên hiện tại.



**Hình 44. Extension đã lấy được phiên cookie hiện tại của trang web**

**Chúng ta sẽ bật F12 lên để kiểm tra xem đúng cookie không.**

Sau khi bật F12 kiểm tra source code lên, ta chọn vào tab **Application** -> **Cookie** -> [**http://localhost**](http://localhost) như hình dưới:



**Hình 45. Xem thông tin cookie của trang web bằng F12 kiểm tra source**

Chúng ta thấy rằng qua 2 hình trên thì thông tin của phiên cookie hiện tại là giống nhau. Chứng minh được extension này đã hoạt động tốt và đã lấy ra được thông tin về phiên cookie hiện tại của trang web **php-training**.

# **IV. HƯỚNG DẪN PHÁT TRIỂN EXTENSION MOUSE\_SELECT**

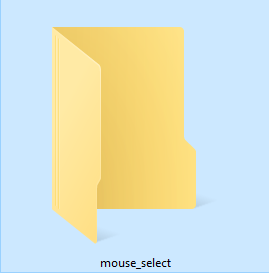
Element custom giúp người dùng có khả năng xóa các element html trực tiếp trên tab hiện hành. Ví dụ có thể sử dụng nó để có một số banner quảng cáo hiện tràn lan trên màn hình. Giúp người dùng có trải nghiệm tốt hơn khi truy cập Internet.

## **1. Quy trình phát triển**

### **1.1. Tạo project mới cho extension Mouse select**

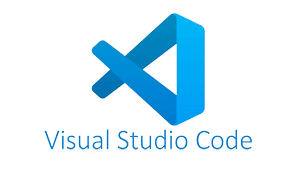
Trước tiên ta cần tạo một cây thư mục và các file cần thiết cho project. Vào ổ đĩa bất kỳ với tên thư mục **mouse\_select**. Tiến hành cài đặt thư mục và bố trí sẽ được cấu trúc thư mục.

Vào ổ đĩa D của máy và tạo một folder có tên: **mouse\_select**



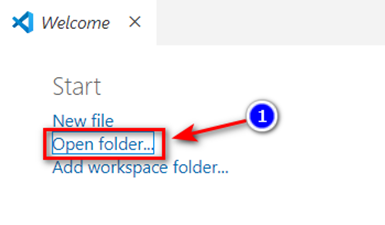
**Hình 46. Folder GET\_cookie**

Sử dụng phần mềm Visual Studio Code để tạo cây thư mục cho project.



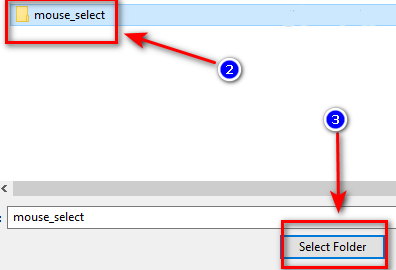
**Hình 47. Phần mềm Visual Studio Code**

Sau khi khởi động phần mềm xong. Chọn vào mục **Open folder**



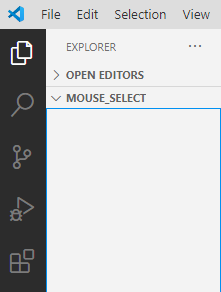
**Hình 48. Chọn Open folder**

Chọn đến folder **mouse\_select** vừa mới tạo bên trên và nhấn Select Folder:



**Hình 49. Chọn Folder mouse\_select và nhấn select folder**

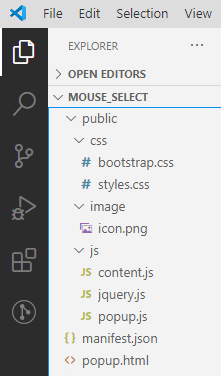
Sau khi select folder chúng ta sẽ có một cây thư mục rỗng như sau:



**Hình 50. Cây thư mục rỗng vừa được select vào**

Các file dành cho project **mouse\_select** bao gồm:

* popup.html
* manifest.json
* Folder public chứa các file css, images, js
  + popup.js
  + content.js



**Hình 51. Các file cho project**

Sau khi đã tạo cấu trúc như hình trên sẽ tiến hành cấu hình cho file **mainifest.json**

### **1.2. Xây dựng file manifest.json**

{

    "manifest\_version": 2,

    "name": "Select Elements",

    "description": "Select and Delete Elements",

    "version": "1.2",

    "permissions": ["tabs", "\*://\*/\*", "storage"],

    "content\_scripts": [{

    "matches": ["<all\_urls>"],

    "js": ["public/js/jquery.js" , "public/js/content.js"]}],

    "browser\_action": {

    "default\_title": "Scope",

    "default\_popup": "popup.html",

    "default\_icon": "public/image/icon.png"

    }

}

**Chú thích :**

|  |  |
| --- | --- |
| "manifest\_version" | Phiên bản hỗ trợ của chrome, hiện tại là 2.0. |
| "name" | Tên của ứng dụng. |
| "description" | Mô tả về ứng dụng. |
| "version" | Phiên bản của ứng dụng |
| "permissions" | Cấp quyền truy cập của ứng dụng đối với trình duyệt. |
| "options\_ui" | "page": Link dẫn tới trang thiết lập.  "open\_in\_tab": Mở tab mới nếu true. |
| "content\_scripts" | Mục Javascript. |
| "browser\_action" | Những mục được hiển thị và mở khi người dùng click vào biểu tượng của Extension. |

### **1.3. Xây dựng giao diện tại file popup.html**

Giao diện này sẽ được hiện ra khi ta click vào button của extension **Mouse select** trên chrome:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="public/css/bootstrap.css">

  <script src="public/js/jquery.js"></script>

  <script src="public/js/popup.js"></script>

</head>

<body style="width: 200px; height: 100px;">

  <div align="center">

    <input type="button" id="btn-start" class="btn btn-primary" value="Start" style="margin-top: 30px">

  </div>

  Sau khi Start -> chọn đối tượng và ấn phím R để xóa :)

</body>

</html>

### **1.4. Xây dựng chức năng trong file popup.js**

//Xử lý sự kiện click

$(document).ready(function () {

    $('#btn-start').click(function(){

        if ($('#btn-start').val() == 'Start') {

            $('#btn-start').val('Started');

            $('#btn-start').attr('disabled', "true");

        }

        //-----------

//gửi sự kiện click tới content.js lúc này content.js sẽ nhận được sự kiện và thực hiện chức năng của nó.

        chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {

            chrome.tabs.sendMessage(tabs[0].id, {"message": "start"});

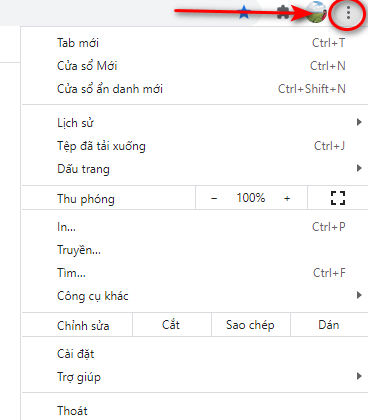
        });

    });

});

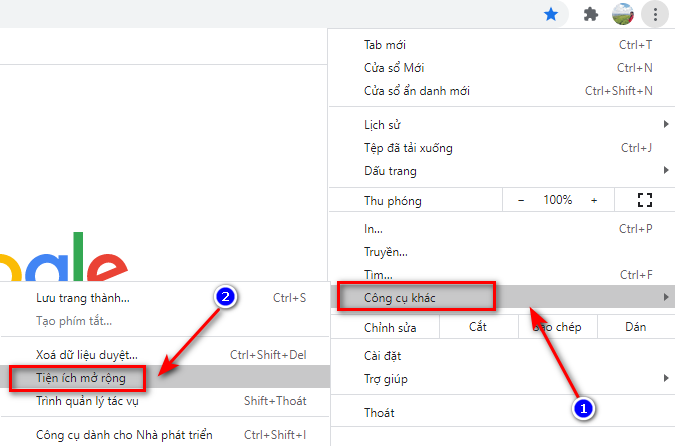
## **2. Hướng dẫn cài đặt extension Mouse select trên chrome**

**Bước 1:** Các bạn mở chrome và nhấn vào nút tùy chỉnh và điều khiển chrome bên góc phải màn hình.



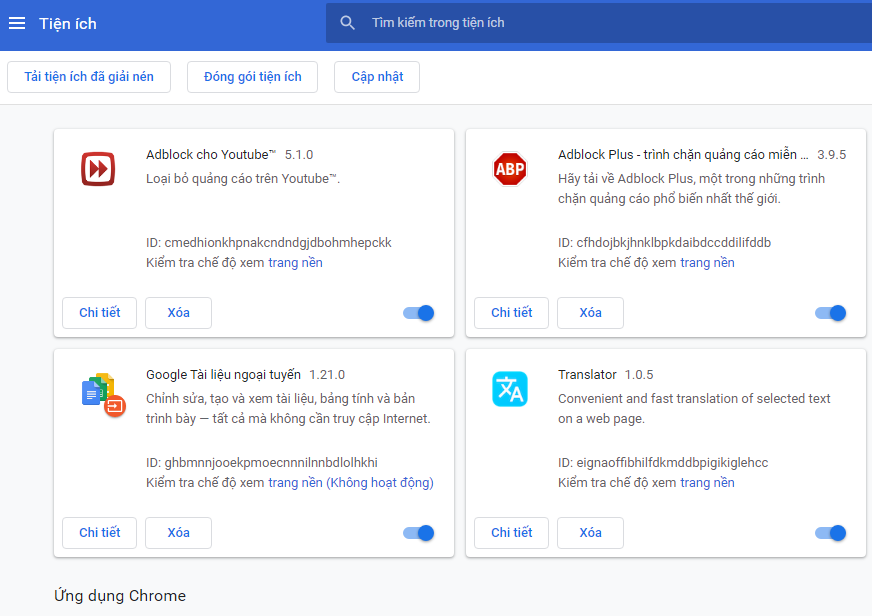
**Hình 52. Giao diện nút tùy chỉnh và điều khiển chrome**

Tiếp theo chọn **Công cụ khác** -> **Tiện ích mở rộng**.



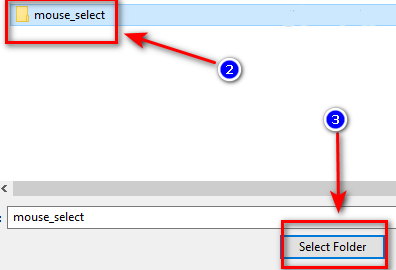
**Hình 53. Giao diện công cụ khác**

**Bước 2:** Sau khi nhấn vào tiện ích mở rộng bạn sẽ được giao diện như sau:



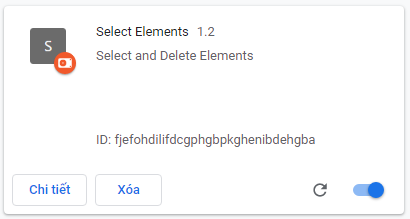
**Hình 54. Giao diện khi nhấn vào tiện ích mở rộng**

Tiếp theo nhấn vào mục: **Tải tiện ích đã giải nén** -> chọn file **mouse\_select** -> chọn **Select Folder**.



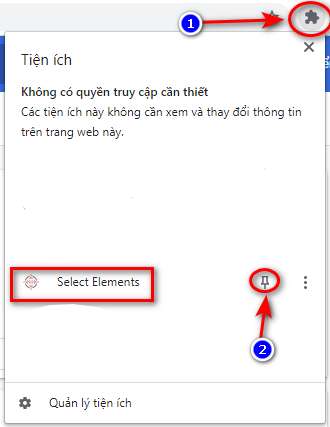
**Hình 55. Chọn Folder mouse\_select và nhấn select folder**

**Bước 3:** Sau khi tải lên xong ta sẽ có một extension như hình:



**Hình 56. Extension mouse\_select đã được cài**

Tiếp theo các bạn nhấn vào biểu tượng: Tiện ích -> chọn đến extension vừa được cài -> nhấn vào biểu tượng nút ghim.



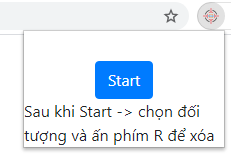
**Hình 57. Ghim extension vừa cài trong tiện ích**

**Bước 4:** Sau khi ghim extension sẽ hiển thị bên ngoài kế bên tiện ích. Nhằm mục đích cho người dùng dễ sử dụng:



**Hình 58. Biểu tượng extension cạnh tiện ích**

Giao diện làm việc của extension **mouse\_select**:



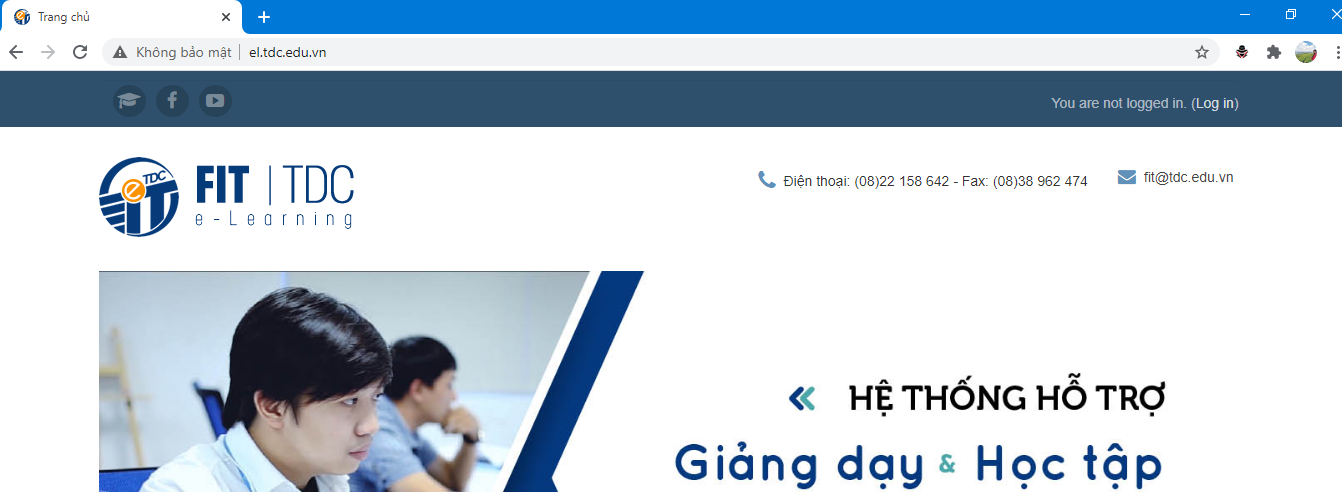
**Hình 59. Giao diện extension mouse\_select**

**Trong đó:**

+ **Start**: nút nhấn bắt đầu.

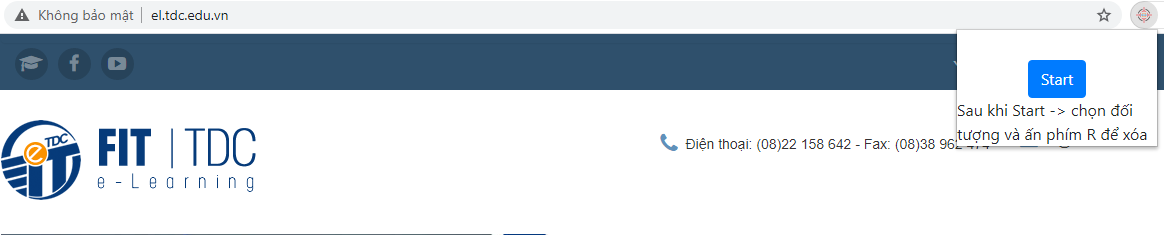
**Demo**: Áp dụng vào trang **el.tdc.edu.vn**.

Các bạn truy cập vào trang <http://el.tdc.edu.vn/> sẽ thấy giao diện như sau:



**Hình 60. Giao diện trang el.tdc.edu.vn**

Sau khi vào được giao diện như hình trên, tiếp theo nhấn vào icon extension **mouse\_select** chúng ta đã cài và và nhấn vào nút **Start**:



**Hình 61. Nhấn Start để bắt đầu**

Sau khi nhấn **Start** chúng ta chọn các phần muốn select trên trang ví dụ: header và footer.

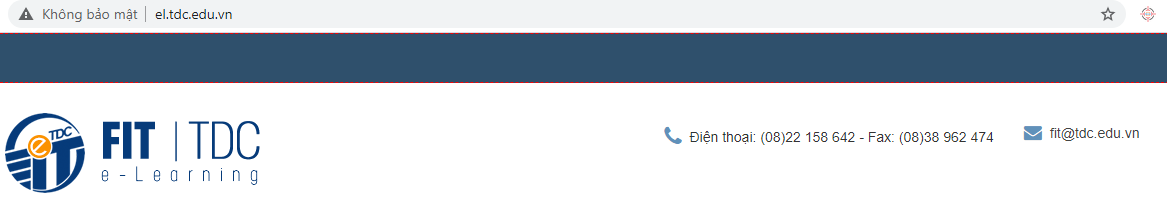
**Áp dụng với header:**



**Hình 62. Hover chuột vào phần muốn xóa**

Khi đưa chuột vào phần muốn xóa, phần đó sẽ có viền đỏ bao quanh giống như hình trên.

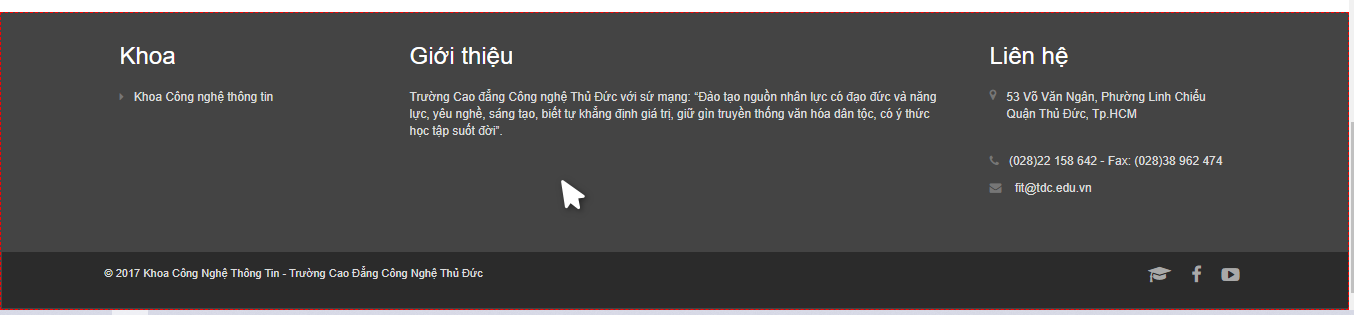
Và sau khi click vào và nhấn chữ R trên bàn phím thì phần được tô viền đỏ sẽ biến mất.



**Hình 63. Phần có viền đỏ đã biến mất**

**Áp dụng vào footer**:

Cũng tương tự như header, chúng ta đưa chuột vào phần muốn select.



**Hình 64. Phần được chọn để select**

Và click chuột.

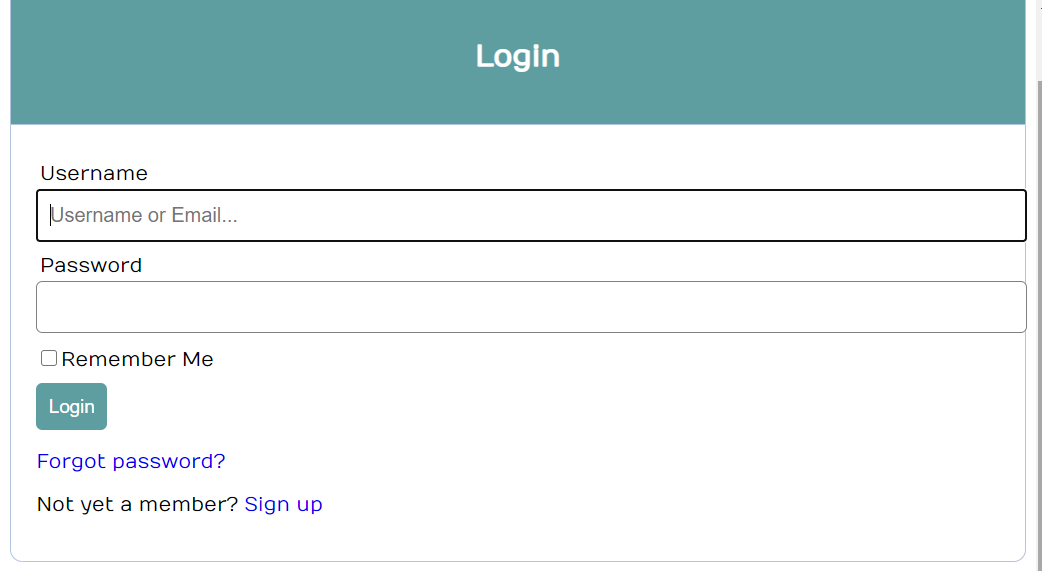
Kết quả: Mất đi phần footer chúng ta đã chọn.



**Hình 65. Kết quả sau khi click chuột**

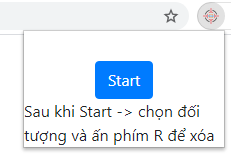
## **3. Áp dụng extension vào source code php-training**

Truy cập vào địa chỉ <http://localhost/php-training/login.php>



**Hình 66. Giao diện trang login của php-training**

Sau đó ta click vào icon của extension **Mouse select** lên và click vào button **Start**.

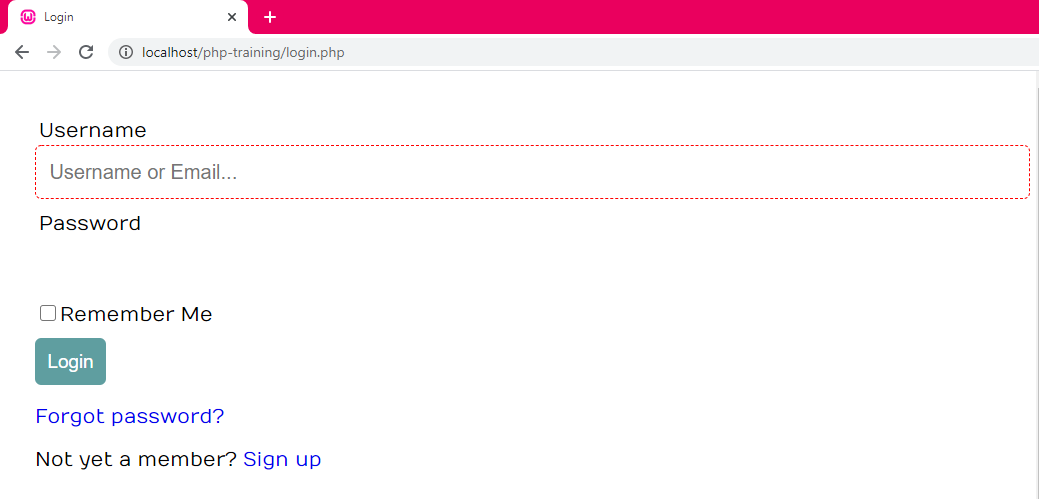


**Hình 67. Nhấn vào button Start**

Mỗi lần rê chuột vào một element thuộc trang login của **php-training** thì element đó sẽ được bao viền đỏ lên, nếu muốn xóa element đó đi ta chỉ cần nhấn nút **R** trên bàn phím.

**Ví dụ:** Chúng ta sẽ chọn vào element có hiện chữ Login và sau đó nhấn **R**.

Lập tức element sẽ bị xóa trên trình duyệt như hình bên dưới:



**Hình 68. Element login đã bị xóa trên tab hiện tại**

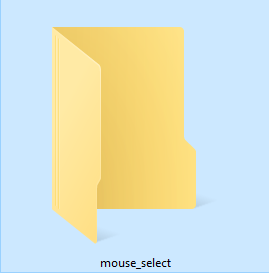
# **V. HƯỚNG DẪN PHÁT TRIỂN EXTENSION ELEMENT CUSTOM**

## **1. Quy trình phát triển**

### **1.1. Tạo project mới cho extension Element Custom**

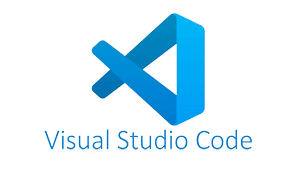
Tạo một cây thư mục và các file cần thiết của project. Vào ổ đĩa bất kỳ với tên thư mục Element\_Custom. Tiến hành cài đặt thư mục và bố trí sẽ được cấu trúc thư mục.

Vào ổ đĩa D của máy và tạo một folder có tên: **Element\_Custom.**



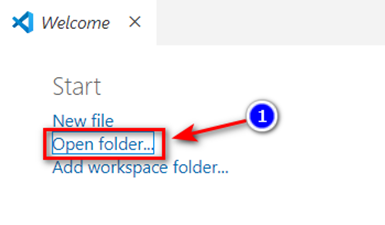
**Hình 69. Folder Element\_Custom**

Sử dụng phần mềm Visual Studio Code để tạo cây thư mục cho project.



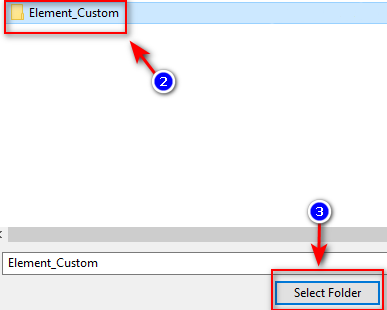
**Hình 70. Phần mềm Visual Studio Code**

Sau khi khởi động phần mềm xong. Chọn vào mục **Open folder.**



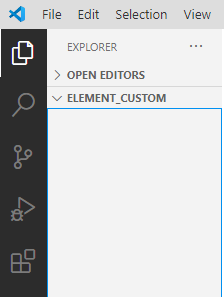
**Hình 71. Chọn Open folder**

Chọn đến folder **Element\_Custom** vừa mới tạo bên trên và nhấn **Select Folder**:



**Hình 72. Chọn Folder Element\_Custom và nhấn select folder**

Sau khi select folder chúng ta sẽ có một cây thư mục rỗng như sau:



**Hình 73. Cây thư mục rỗng vừa được select vào**

Các file dành cho project **mouse\_select** bao gồm:

+ popup.html

+ manifest.json

+ folder public chứa các file css, images, js

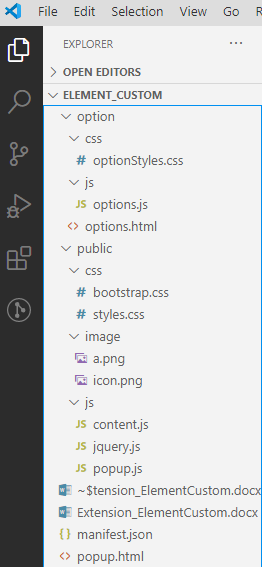
- popup.js

- content.js

+ Thư mục option

- option.html

- option.js



**Hình 74. Các file cho project.**

Sau khi đã tạo cấu trúc như hình trên sẽ tiến hành cấu hình cho file **mainifest.json**

### **1.2. Xây dựng file manifest.json**

{

    "manifest\_version": 2,

    "name": "Select Elements",

    "description": "Select and Delete Elements",

    "version": "1.2",

    "permissions": ["tabs", "\*://\*/\*", "storage"],

    "content\_scripts": [{

    "matches": ["<all\_urls>"],

    "js": ["public/js/jquery.js" , "public/js/content.js"]}],

    "browser\_action": {

    "default\_title": "Scope",

    "default\_popup": "popup.html",

    "default\_icon": "public/image/icon.png"

    }

}

**Chú thích:**

|  |  |
| --- | --- |
| "manifest\_version" | Phiên bản hỗ trợ của chrome, hiện tại là 2.0. |
| "name" | Tên của ứng dụng. |
| "description" | Mô tả về ứng dụng. |
| "version" | Phiên bản của ứng dụng |
| "permissions" | Cấp quyền truy cập của ứng dụng đối với trình duyệt. |
| "options\_ui" | "page": Link dẫn tới trang thiết lập.  "open\_in\_tab": Mở tab mới nếu true. |
| "content\_scripts" | Mục JavaScript. |
| "browser\_action" | Những mục được hiển thị và mở khi người dùng click vào biểu tượng của Extension. |

### **1.3. Xây dựng giao diện tại file popup.html**

Giao diện này sẽ được hiện ra khi ta click vào button của extension **Element Custom** trên chrome:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="public/css/bootstrap.css">

  <link rel="stylesheet" type="text/css" href="public/css/styles.css">

  <script src="public/js/jquery.js"></script>

</head>

<body>

  <header>

    <button id="go-to-options" class="btn btn-outline-primary">⚙</button>

    <h3 align="center">Element Custom</h3>

  </header>

  <div class="container">

    <h3>Urls</h3>

    <div class="input-group mb-3">

      <input id="production-urls" type="text" class="form-control" placeholder="Link" aria-label="Link" aria-describedby="basic-addon2">

      <div class="input-group-append">

        <button id="saveUrl" class="btn btn-outline-success" type="button">Save</button>

      </div>

    </div>

    <h3>Element</h3>

    <div class="input-group mb-3">

      <input id="elements" type="text" class="form-control" placeholder="Element" aria-label="Element" aria-describedby="basic-addon2">

      <div class="input-group-append">

        <select id="select-type" class="browser-default">

          <option value=".">CLASS</option>

          <option value="#">ID</option>

          <option value="">TAG</option>

        </select>

        <button id="save" class="btn btn-outline-success" type="button">Add</button>

      </div>

    </div>

    <div class="custom-control custom-checkbox">

     <input type="checkbox" id="cbx-auto-delete" class="custom-control-input">

<label class="custom-control-label" for="cbx-auto-delete">Tự động xóa.</label>

    </div>

    <button type="button" id="delete-list-element" class="btn btn-danger btn-sm btn-block">Xóa</button>

  </div>

    <script src="public/js/popup.js"></script>

</body>

</html>

### **1.4. Xây dựng chức năng trong file popup.js**

$(document).ready(function() {

    //check active when first use

    //khởi tạo bộ lưu trữ nếu mở lần đầu

    chrome.storage.sync.get(["active"], function(active) {

        if (!active['active']) {

            urls = {"urls" : [],

                    "elements": []};

            chrome.storage.sync.set({

            "list\_urls": urls

            }, function() {

                console.log("active finish!");

            });

            chrome.storage.sync.set({

            "active": true

            });

        }

    });

    //----------------

    //event button. Go to Setting

    //xử lý sự kiện click vào mục cài đặt

    $('#go-to-options').click(function() {

      if (chrome.runtime.openOptionsPage) {

        chrome.runtime.openOptionsPage();

      } else {

        window.open(chrome.runtime.getURL('options.html'));

      }

    });

    // ----------------------

    //save location of page

    //Lấy link ở page hiện tại trong bộ lưu trữ

    chrome.storage.sync.get(["location-href"], function(url) {

        $("#production-urls").val(url["location-href"]);

    });

//-------------------

//xử lý sự kiện lưu trang hiện tại vào bộ nhớ.

    $("#saveUrl").click(function() {

        var url = $("#production-urls").val();

        chrome.storage.sync.get(["list\_urls"], function(items) {

            var urls = items["list\_urls"]["urls"] || "";

            var elements = items["list\_urls"]["elements"] || "";

            urls.push(url);

            temp\_urls = {"urls" : urls, "elements": elements};

            chrome.storage.sync.set({

            "list\_urls": temp\_urls

            }, function() {

                alert("Đã lưu!");

            });

        });

    });

//------------------------

//xử lý sự kiện lưu trữ các Element.

    $("#save").click(function() {

        var element = $("#select-type").val() + $("#elements").val();

        chrome.storage.sync.get(["list\_urls"], function(items) {

            var urls = items["list\_urls"]["urls"] || "";

            var elements = items["list\_urls"]["elements"] || "";

            elements.push(element);

            temp\_element = {"urls" : urls, "elements": elements};

            chrome.storage.sync.set({

            "list\_urls": temp\_element

            }, function() {

                alert("Đã lưu!");

            });

        });

    });

    //-------------

//set màu đã lưu trong bộ nhớ cho giao diện App

    chrome.storage.sync.get(["border-color", "color", "text-color"], function(colors) {

        $("body").css("background", colors["border-color"]);

        $(".container").css("background", colors["color"]);

        $("footer>div").css("color", colors["text-color"]);

        $("h3").css("color", colors["text-color"]);

    });

    //------

    //xử lý sự kiện khi click vào nút xóa

    $("#delete-list-element").attr('checked', true);

    $("#delete-list-element").click(function() {

        chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {

            var activeTab = tabs[0];

            chrome.tabs.sendMessage(activeTab.id, {"message": "start"});

        });

    });

    //--------

    //checkbox auto xóa.

    chrome.storage.sync.get(["auto-delete"], function(checkbox) {

        if (checkbox["auto-delete"]) {

            $('.custom-control-input').attr('checked', true);

            $('#delete-list-element').attr('disabled', true);

            chrome.storage.sync.set({

                "auto-delete": true

            });

        }

        else{

            $('.custom-control-input').attr('checked', false);

            $('#delete-list-element').attr('disabled', false);

            chrome.storage.sync.set({

                "auto-delete": false

            });

        }

    });

    //------------------------

    //xử lý sự kiện checkbox auto xóa

    $('#cbx-auto-delete').click(function() {

        if ($(this).is(':checked')) {

            $('#delete-list-element').attr('disabled', true);

            chrome.storage.sync.set({

                "auto-delete": true

                }, function() {

                    console.log("saved");

            });

        } else{

            $('#delete-list-element').attr('disabled', false);

            chrome.storage.sync.set({

                "auto-delete": false

                }, function() {

                    console.log("saved");

            });

        }

    });

});

### **1.5. Xây dựng chức năng trong file content.js**

//gọi chức năng xóa Element.

$(document).ready(function() {

     //Lấy link page hiện tại (content) gửi qua popup

    chrome.storage.sync.set({

    "location-href": location.href

    });

    chrome.storage.sync.get(["auto-delete"], function(checkbox) {

        if (checkbox["auto-delete"]) {

            deleteElements();

        }

    });

     //-----

    chrome.runtime.onMessage.addListener(

        function(request, sender, sendResponse) {

            if( request.message === "start" ) {

                deleteElements();

            }

        }

    );

});

// ----------

//Xóa Element tại trang hiện tại.

function deleteElements() {

    chrome.storage.sync.get(["list\_urls"], function(items) {

        var urls = items["list\_urls"]["urls"];

        var elements = items["list\_urls"]["elements"];

        if (urls.length > 0) {

            if (elements.length > 0) {

                for (var i = 0; i < urls.length; i++) {

                    if (urls[i] == location.href) {

                        for (var i = 0; i < elements.length; i++) {

                            $(elements[i]).remove();

                        }

                    }

                }

            }

        }

    });

}

### **1.6. Xây dựng chức năng trong file option.html**

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Cài Đặt</title>

    <link rel="stylesheet" type="text/css" href="../public/css/bootstrap.css">

    <link rel="stylesheet" type="text/css" href="css/optionStyles.css">

    <script src="../public/js/jquery.js"></script>

    <script src="js/options.js"></script>

</head>

<body>

    <header>

        <h1 align="center">Thiết Lập</h1>

    </header>

    <div id="status" style="position: absolute; font-size: 36px; top: 40%; left: 40%;"></div>

    <div class="container">

        <div class="row">

            <div class="col-md-4" style="border: 2px solid #ccc; border-radius: 10px; padding: 20px;margin-bottom: 100px;">

                <h1 align="center"> Setting</h1>

                <div align="left">Border Color</div>

                <select id="border-color" class="form-control form-control-lg">

                    <option value="">Default</option>

                </select>

                <div align="left">Background Color</div>

                <select id="color" class="form-control form-control-lg">

                    <option value="">Default</option>

                </select>

                <div align="left">Text Color</div>

                <select id="text-color" class="form-control form-control-lg">

                    <option value="">Default</option>

                </select>

                <button id="save" type="button" class="btn btn-primary btn-lg btn-block" style="margin-top: 10px;">Save</button>

            </div>

            <div class="col-md-4" id="list-urls" style="border: 2px solid #ccc; border-radius: 10px; padding: 20px;margin-bottom: 100px;">

                <h1 align="center"> List Urls</h1>

            </div>

            <div class="col-md-4" id="list-element" style="border: 2px solid #ccc; border-radius: 10px; padding: 20px;margin-bottom: 100px;">

                <h1 align="center"> List Element</h1>

            </div>

        </div>

    </div>

    <div class="loader" style="margin-top: 50px;"><b style="position: relative; top: 35%; left: 11px;">Xin Chờ!</b></div>

</body>

</html>

### **1.7. Xây dựng chức năng trong file option.js**

$(document).ready(function() {

    chrome.storage.sync.get(["list\_urls"], function(items) {

        setTimeout(function() {

            var urls = items["list\_urls"]["urls"];

            var elements = items["list\_urls"]["elements"];

            $(".loader").remove();

            for (var i = 0; i < urls.length; i++) {

                $("#list-urls").append('<input type="button" id = "url" class="btn btn-outline-primary btn-lg btn-block" value="'+urls[i]+'"></input>');

            }

            for (var i = 0; i < elements.length; i++) {

                $("#list-element").append('<input type="button" id = "element" class="btn btn-outline-primary btn-lg btn-block" value="'+elements[i]+'"></input>');

            }

            //Lấy danh sách link Url và element từ bộ nhớ.

            $("input[type=button]").click(function() {

                var deleteE = $(this).val();

                var getIDE = $(this).attr("id");

                var tempArr = [];

                var j = 0;

                if (getIDE == "url") {

                    for (var i = 0; i < urls.length; i++) {

                        if (urls[i] != deleteE) {

                            tempArr[j] = urls[i];

                            j++;

                        }

                    }

                    urls = {"urls":tempArr, "elements":elements};

                    chrome.storage.sync.set({

                    "list\_urls": urls

                    }, function() {

                        location.reload();

                    });

                } else{

                    for (var i = 0; i < elements.length; i++) {

                        if (elements[i] != deleteE) {

                            tempArr[j] = elements[i];

                            j++;

                        }

                    }

                    urls = {"urls":urls, "elements":tempArr};

                    chrome.storage.sync.set({

                    "list\_urls": urls

                    }, function() {

                        location.reload();

                    });

                }

            });

        }, 750);

    });

            //-------------------

    //Xử lý sự kiện khi click vào mục nào thì sẽ xóa mục đó.

    var color = ['#e6194b', '#3cb44b', '#ffe119', '#4363d8', '#f58231', '#911eb4', '#46f0f0', '#f032e6', '#bcf60c', '#fabebe', '#008080', '#e6beff', '#9a6324', '#fffac8', '#800000', '#aaffc3', '#808000', '#ffd8b1', '#000075', '#808080', '#ffffff', '#000000'];

    for (var i = 0; i < color.length; i++) {

        $("#border-color").append('<option style = "color: '+color[i]+';" value="'+color[i]+'">'+color[i]+'</option>');

        $("#color").append('<option style = "color: '+color[i]+';" value="'+color[i]+'">'+color[i]+'</option>');

        $("#text-color").append('<option style = "color: '+color[i]+';" value="'+color[i]+'">'+color[i]+'</option>');

    }

    $("#border-color").change(function() {

        $("#border-color").css("color",$("#border-color").val());

    });

    $("#color").change(function() {

        $("#color").css("color",$("#color").val());

    });

    $("#text-color").change(function() {

        $("#text-color").css("color",$("#text-color").val());

    });

    $("#save").click(function() {

        var color1 = $("#border-color").val();

        var color2 = $("#color").val();

        var color3 = $("#text-color").val();

        chrome.storage.sync.set({

        "border-color": color1,

        "color": color2,

        "text-color": color3

        }, function() {

            $("#save").html("Save Finish!");

            $("#save").attr('disabled', true);

            setTimeout(function() {

                $("#save").html("Save");

                $("#save").attr('disabled', false);

            }, 2000);

        });

    });

    chrome.storage.sync.get(["border-color", "color", "text-color"], function(colors) {

        $("#border-color").val(colors["border-color"]);

        $("#color").val(colors["color"]);

        $("#text-color").val(colors["text-color"]);

        $("#border-color").css("color",colors["border-color"]);

        $("#color").css("color",colors["color"]);

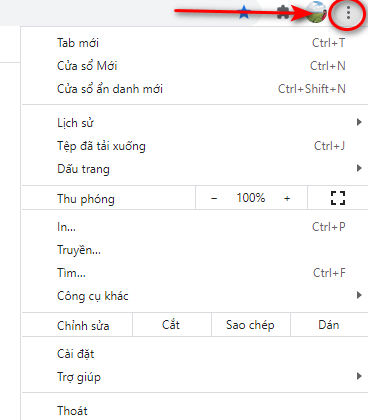
        $("#text-color").css("color",colors["text-color"]);

    });

});

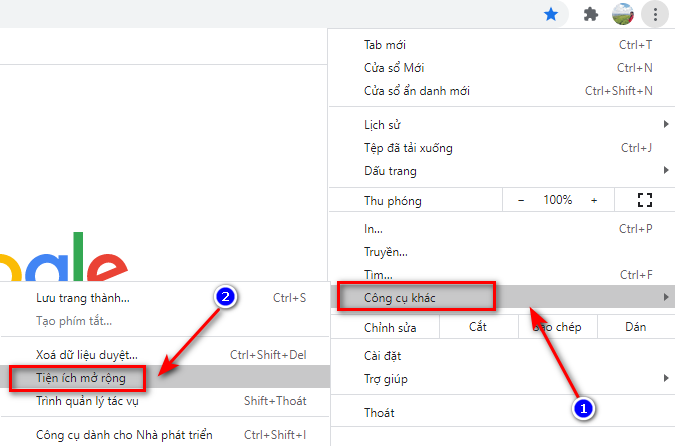
## **2. Hướng dẫn cài đặt extension Element Custom trên chrome**

**Bước 1**: Các bạn mở chrome và nhấn vào nút tùy chỉnh và điều khiển chrome bên góc phải màn hình.



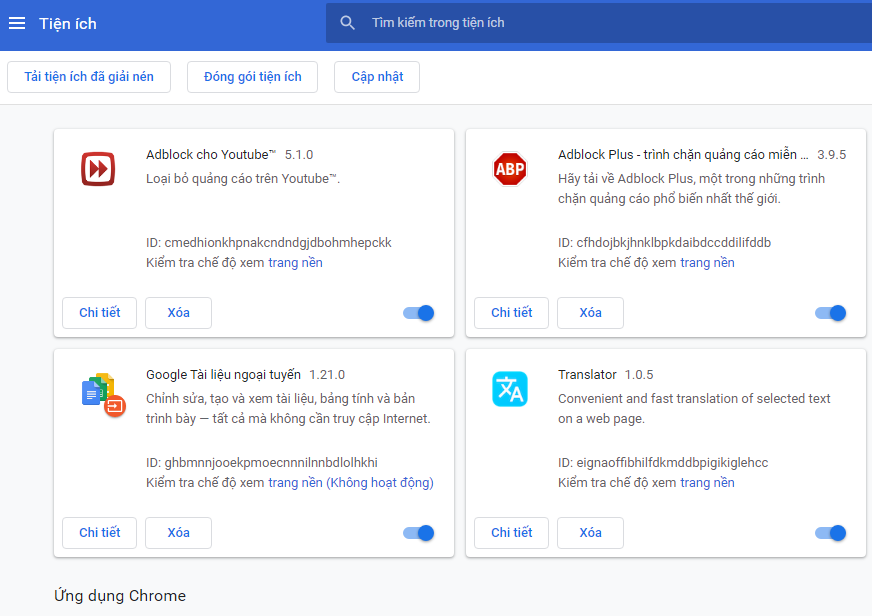
**Hình 75. Giao diện nút tùy chỉnh và điều khiển chrome**

Tiếp theo chọn **Công cụ khác** -> **Tiện ích mở rộng**.



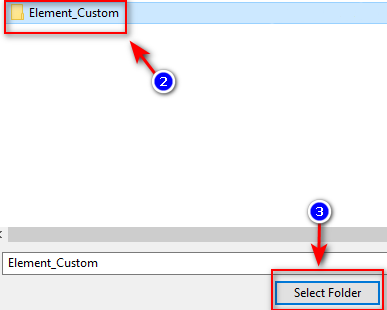
**Hình 76. Giao diện công cụ khác**

**Bước 2:** Sau khi nhấn vào tiện ích mở rộng bạn sẽ được giao diện như sau:



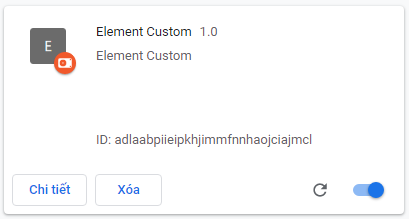
**Hình 77. Giao diện khi nhấn vào tiện ích mở rộng**

Tiếp theo nhấn vào mục: **Tải tiện ích đã giải nén** -> chọn file **Element\_Custom** -> chọn **Select Folder**.



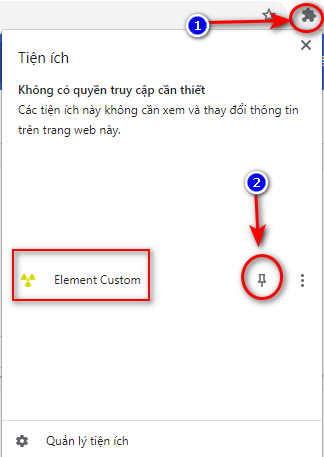
**Hình 78. Chọn Folder Element\_Custom và nhấn select folder**

**Bước 3:** Sau khi tải lên xong ta sẽ có một extension như hình:



**Hình 79. Extension Element\_Custom đã được cài**

Tiếp theo các bạn nhấn vào biểu tượng: Tiện ích -> chọn đến extension vừa được cài -> nhấn vào biểu tượng nút ghim.



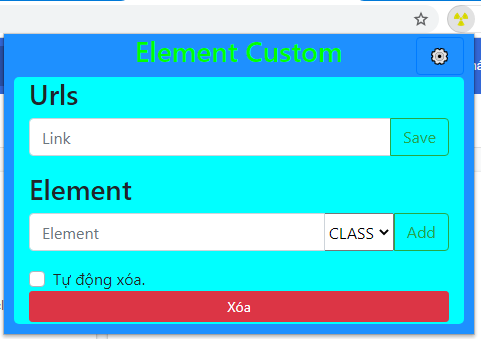
**Hình 80. Ghim extension vừa cài trong tiện ích**

**Bước 4:** Sau khi ghim extension sẽ hiển thị bên ngoài kế bên tiện ích. Nhằm mục đích cho người dùng dễ sử dụng:



**Hình 81. Biểu tượng extension cạnh tiện ích**

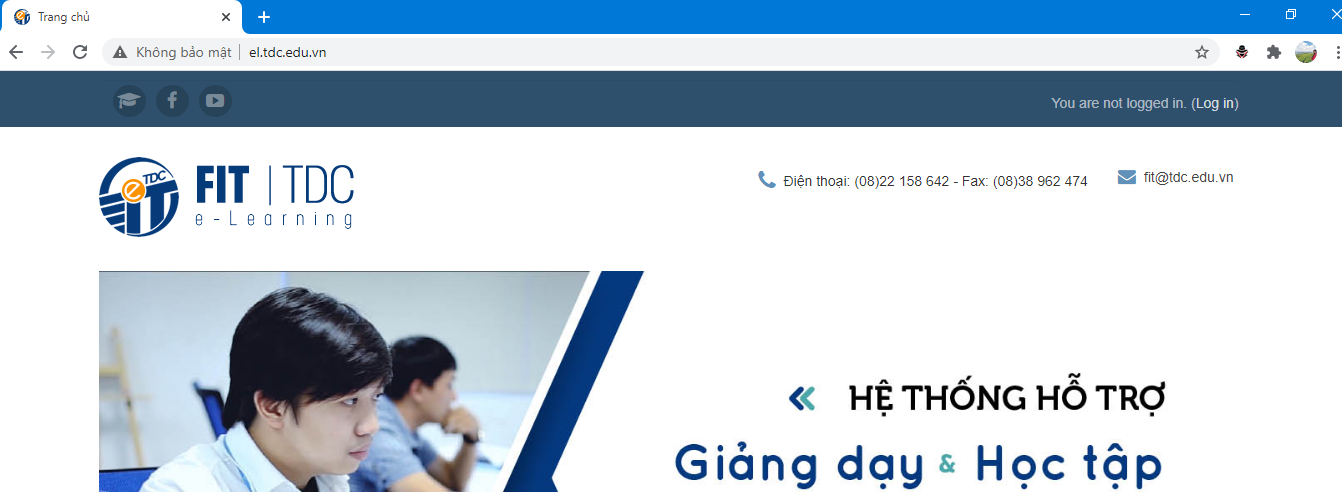
Giao diện làm việc của extension **Element\_Custom**:



**Hình 82. Giao diện extension Element\_Custom**

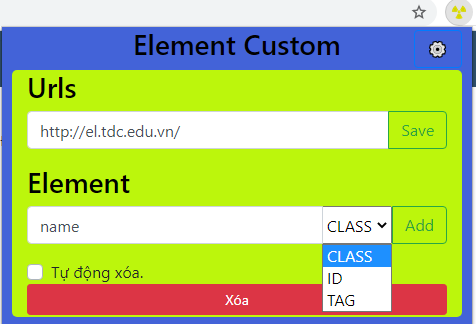
**Demo:** Áp dụng vào trang el.tdc.edu.vn.

Các bạn truy cập vào trang <http://el.tdc.edu.vn/> sẽ thấy giao diện như sau:



**Hình 83. Giao diện trang el.tdc.edu.vn.**

**Bước 5:** Sau khi vào được giao diện như hình trên, tiếp theo nhấn vào icon extension **Element\_Custom** chúng ta đã cài:

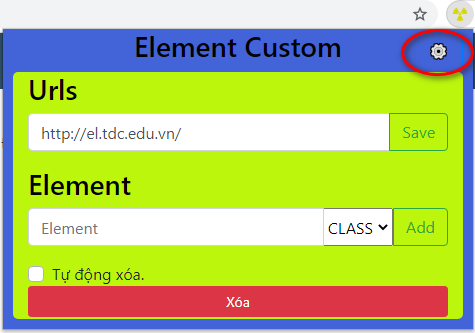


**Hình 84. Giao diện làm việc Element\_Custom.**

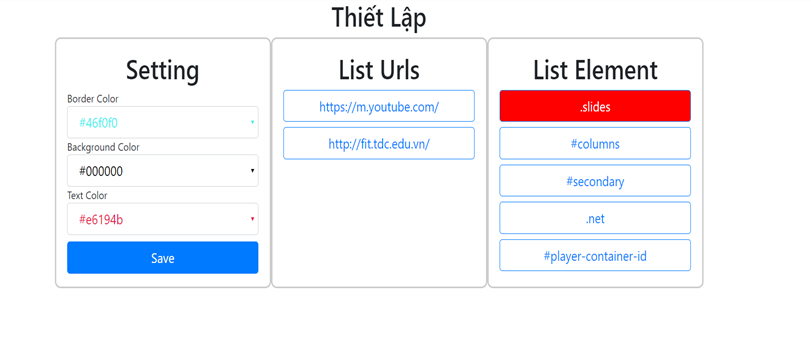
**Trong đó :**

* **Urls:** là địa chỉ trang hiện tại.
* **Element:** có thể chọn class, id hoặc tag.

Có thể thiết kế giao diện Element tại thiết lập bằng cách nhấn vào biểu tượng cài đặt trong giao diện của extension **Element\_Custom**



**Hình 85. Nút thiết lập.**



**Hình 86. Giao diện thiết lập**

Sau khi thiết lập xong chỉ cần nhấn **Save** là hoàn tất lưu thiết lập.