

IERG4210 Assignment Phase 4

BOOTSTRAP

JQUERY

JS JAVASCRIPT

CSS3

PHP

Name: Lau Long Ching

SID: 1155127347

Please visit <http://3.13.126.10> or <https://secure.s31.ierg4210.ie.cuhk.edu.hk/> to mark my assignment, thank you!

Phase 3 marking checklist

1. No XSS Injection and Parameter Tampering Vulnerabilities in the whole website

complete

- [UI Enhancement Only] Proper and vigorous client-side input restrictions for all forms

complete

```
pattern="/^[a-zA-Z0-9_]+$"/></div>
<label class="mt-1" for="prod_name"> Na
<div> <input class="form-control" id="p
| | pattern="/^[a-zA-Z0-9_]+$"/></div>
<label class="mt-1" for="prod_price"> P
<div> <input class="form-control" id="p
| | pattern="/^[0-9]+$/"/></div>
```

- Proper and vigorous server-side input sanitizations and validations for all forms

complete

```
$catid = $_POST["catid"];
$name = htmlspecialchars($_POST["name"]);
$price = $_POST["price"];
$desc = htmlspecialchars($_POST["description"]);
$inv = $_POST["inventory"];
$filename = $name . ".jpg";
$sql = "UPDATE products SET catid= ? , name= ?, price= ?,
$q = $db->prepare($sql);
$q->bindParam(1, $catid);
```

- Proper and vigorous context-dependent output sanitizations

complete

```
<div class="brand"> Hey there,
['username']) {echo htmlspecialchars($_SESSION['username']);} else
document.getElementById('shoppingcarttable').innerHTML;
* localStorage.getItem(info.PID);
td>' + HtmlSanitizer.SanitizeHtml(info.NAME) + '</td><td
ById('shoppingcarttable').innerHTML = tablehtml;
```

2. Mitigate SQL Injection Vulnerabilities in the whole website

complete

- Apply parameterized SQL statements with the PDO library

complete

```
$admin_flag = 2;
$db = ierg4210_DB();
$q = $db->prepare("INSERT INTO account
$q->bindParam(1, $email);
$q->bindParam(2, $salt);
$q->bindParam(3, $password);
$q->bindParam(4, $admin_flag);
$q->execute();
header('Location: login.php', true, 301);
```

3. Mitigate CSRF Vulnerabilities in the whole website

complete

- Apply and validate secret nonces for every form

complete

```
</script>
<button type="submit" class="btn btn-primary mt-2">Add</button>
<input type="hidden" name="nonce" value="<?php echo csrf_getNonce('prod_insert'); ?>" />
</form>
</fieldset>
```

- ALL forms must defend against Traditional and Login CSRF

complete

```
<button class="btn btn-outline-light btn-lg px-5" type="submit">Login</button>
<input type="hidden" name="nonce" value="<?php echo csrf_getNonce('login'); ?>" />
</form>
```

4. Authentication for Admin Panel

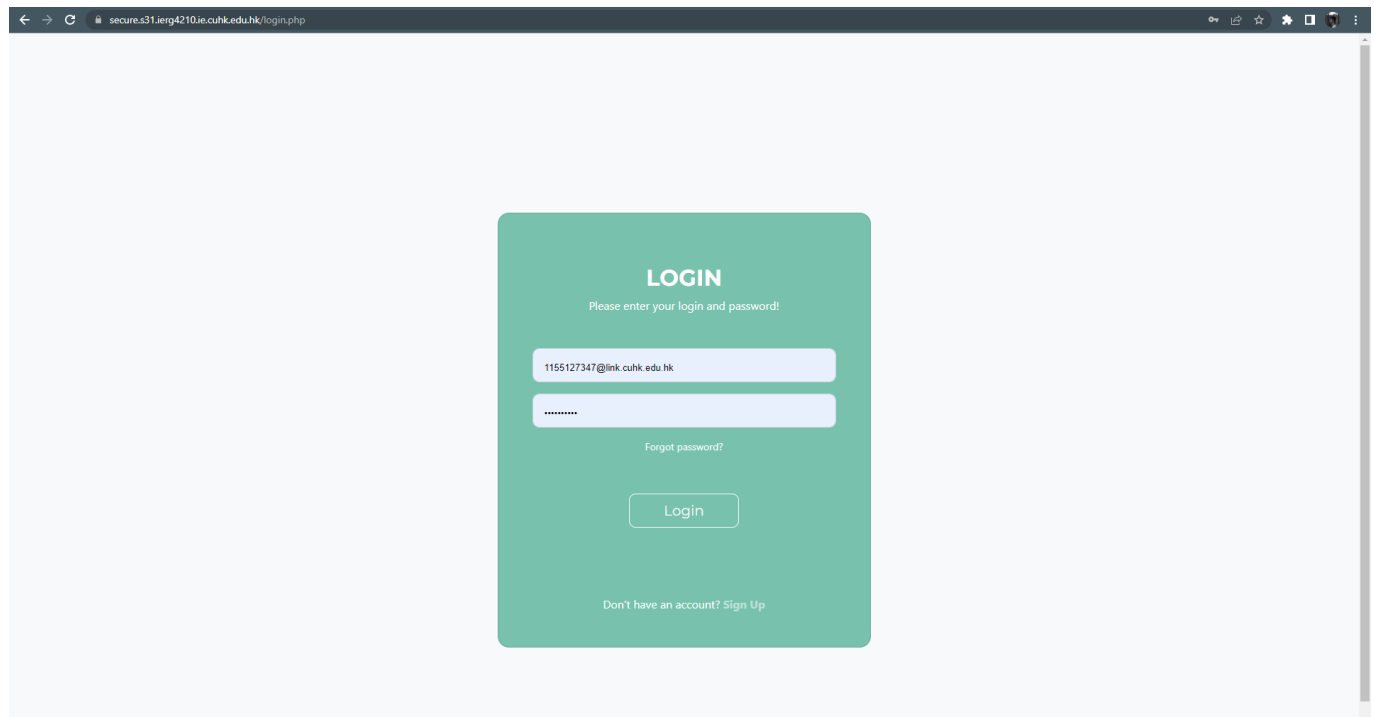
- Create a user table (or a separate DB with only one user table)
 - Required columns: userid (primary key), email, password
 - Data: at least 2 users of your choice, 1 admin and 1 normal user (using admin flag)
 - Security: Passwords must be properly salted and hashed before storage

complete

```
FILENAME TEXT, FOREIGN KEY(CATID) REFERENCES CATEGORIES(CATID));
CREATE INDEX I1 ON PRODUCTS(CATID);
CREATE TABLE account ( userid integer primary key, email text, salt text, password text, admin_flag integer);
sqlite>
```

- Build a login page login.php that requests for email and password

complete



- Upon validated and authenticated, redirect the user to the admin panel or main page
- Indicate user name (or "guest" if not logged in) in your website

complete

complete

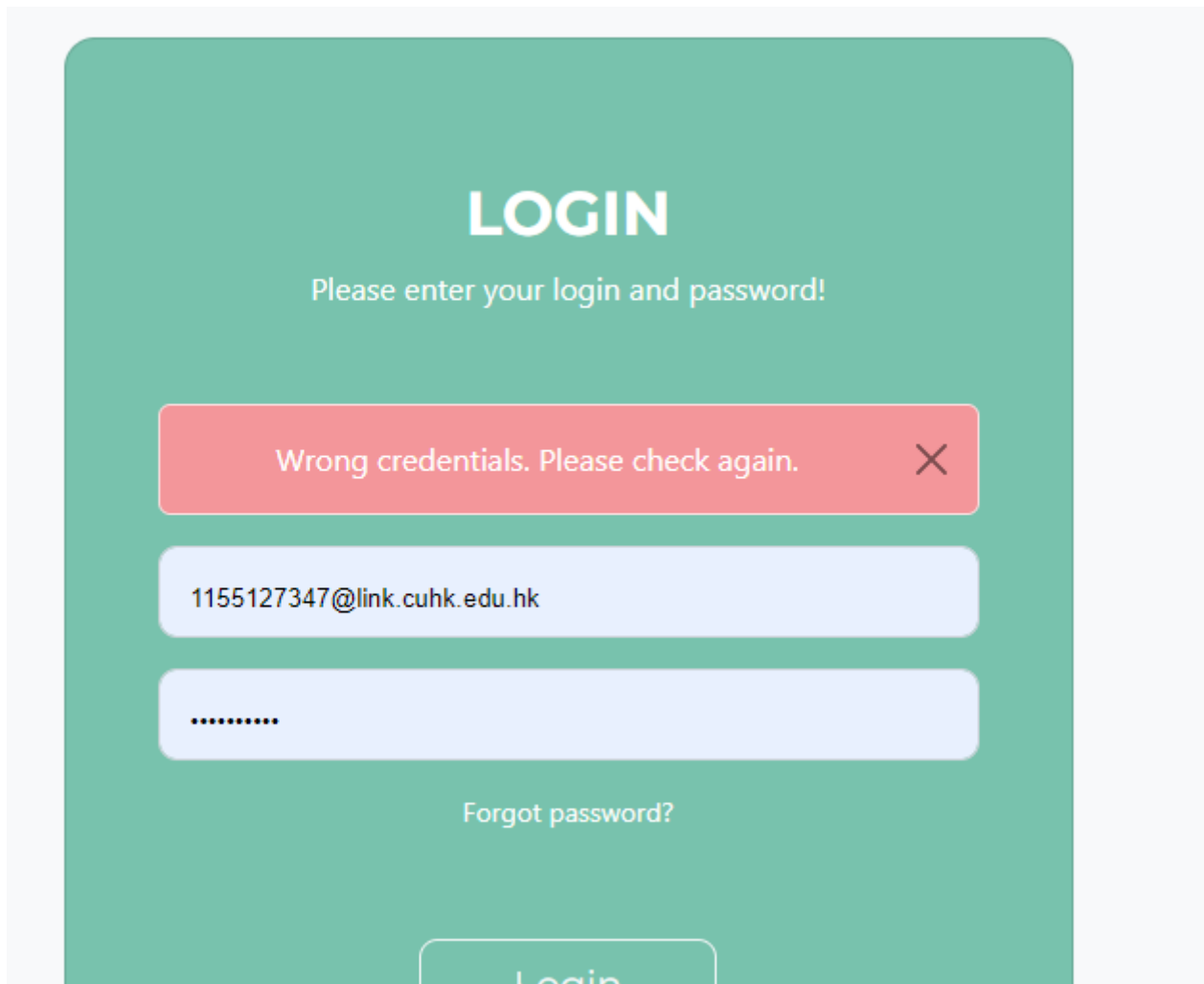
Hey there, 1155127347@link.cuhk.edu.hk!

You are now at: [Home](#)

Categories:

- Otherwise, prompt for errors (i.e. either email or password is incorrect)

complete

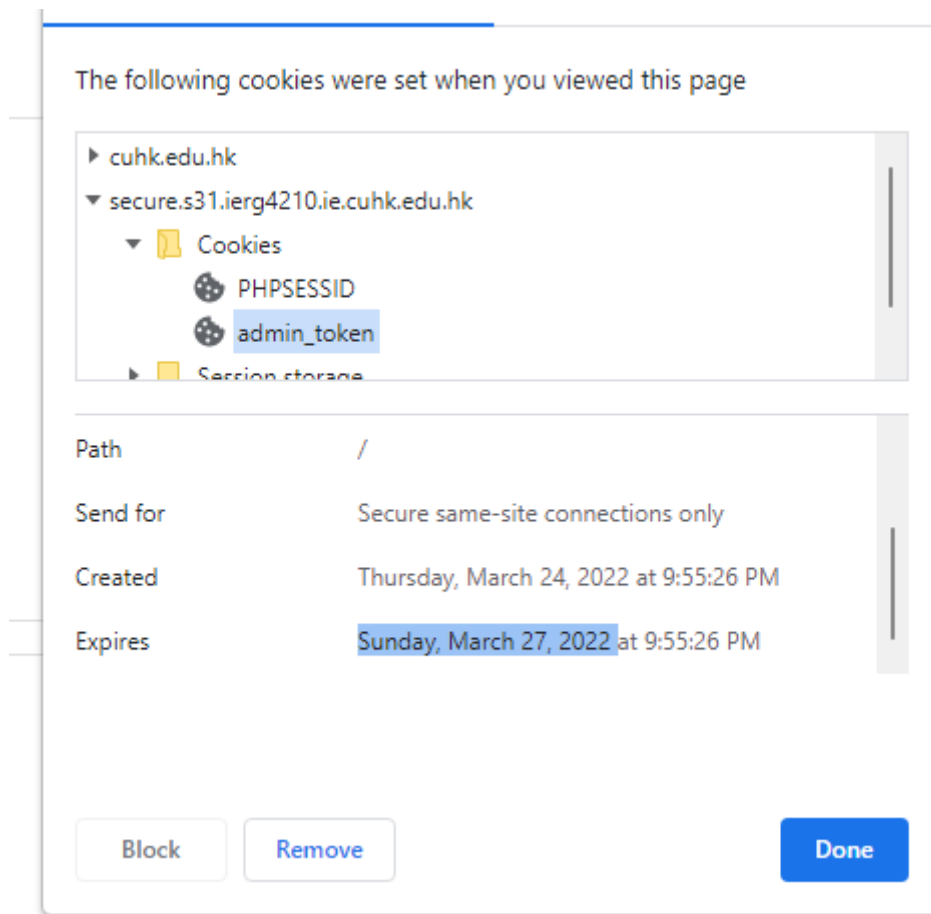


- A separated normal user login page is not compulsory

complete

- Maintain an authentication token using Cookies (with httpOnly)
 - Cookie name: auth; value: a hashed token; property: httpOnly
 - Cookies persist after browser restart (i.e. $0 < \text{expires} < 3$ days)

complete



- No Session Fixation Vulnerabilities (rotate session id upon successful login)

complete

```
$saltedPwd = hash_hmac('sha256', $pwd,  
if($saltedPwd == $r['password']){  
    $exp = time() + 3600*24*3;  
    $token = array(  
        'em' => $r['email'],  
        'exp' => $exp,  
    );  
}
```

- Configure all authentication cookies to use the Secure and HttpOnly flags

complete

```
if($r['admin_flag'] == 1){  
    // create the cookie, make it HTTP only  
    setcookie('admin_token', json_encode($token), $exp, '', '', true, true);  
    // put it also in the session  
    $_SESSION['admin_token'] = $token;  
    $_SESSION['username'] = $email;  
}
```

- Validate the authentication token before revealing and executing admin features
 - If successful, let admin users access the admin panel and execute admin features
 - Otherwise (e.g. empty or tampered token), redirect back to the login page or main page
 - Security: both admin.html and admin-process.php must validate the auth. token

complete

```

include_once( '../lib/CSRF.php' );

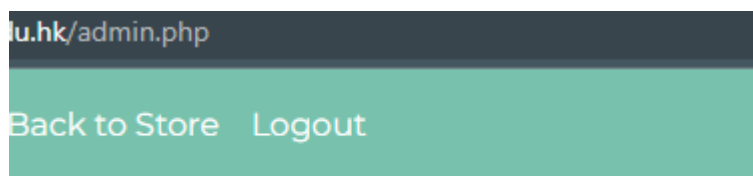
if (!auth_admin()){
    header('Location: ../login.php');
    exit();
}

function auth_admin(){
    if (!empty($_SESSION['admin_token']))
        return $_SESSION['admin_token']['em'];
    if (!empty($_COOKIE['admin_token'])){
        if ($t = json_decode(stripslashes($_COOKIE['admin_token']), true)){
            if (time() > $t['exp'])
                return false;
            global $db;
            $db = ierg4210_DB();
            $q = $db->prepare('SELECT * FROM account WHERE email = ? ');
            $q ->execute(array($t['email']));
            if ($r=$q->fetch()){
                $realk = hash_hmac('sha1', $t['exp'].$r['password'].$r['salt']);
                if($realk == $t['k']){
                    $_SESSION['admin_token'] = $t;
                    return $t['em'];
                }
            }
        }
    }
}

```

- PHP & SQL: Provide a logout feature that clears the authentication token

complete



- Supporting Change of Password
 - Must validate the current password first
 - Logout user after the password is changed

complete

The screenshot shows a web browser window with the URL 'secure.s31.ierg4210ie.cuhk.edu.hk/changepw.php'. The page has a green header bar with navigation links: 'IERG4210 Store', 'Home', 'Categories', 'Shopping Cart', and 'Logout'. There is a search bar on the right. The main content area is light gray and contains a white box titled 'Change Password'. Inside the box, it says 'Please enter your login, old and new password!'. There are two input fields: 'Current Password' and 'New Password'. Below the fields is a green 'Change' button.

5. All generated session IDs and nonces are not guessable throughout the whole assign.

complete

- e.g., the login token must not reveal the original password in plaintext

complete

```
$saltedPw = hash_hmac('sha256', $pwd, $r['salt']);
if($saltedPw == $r['password']){
    $exp = time() + 3600*24*3;
    $token = array(
        'em' => $r['email'],
        'exp' => $exp,
        'k' => hash_hmac('sha256', $exp.$r['password'], $r['salt'])
    );
```

- e.g., the CSRF nonce when applied in a hidden field must be random

complete

```
function csrf_getNonce($action){
    //Generate a nonce with mt_rand()
    $nonce = mt_rand() . mt_rand();
    // With regard to $action, save the
    if (!isset($_SESSION['csrf_nonce']))
```

6. Apply SSL certificate for secure.s[1-80].ierg4210.ie.cuhk.edu.hk

complete

- Certificate Application
 - When generating a CSR, use CUHK as Organization Name
 - Apply for a 90-day free certificate at <https://www.ssl.com/certificates/free/buy/> or <https://letsencrypt.org/> (or others)

complete



- Certificate Installation

complete

- Install the issued certificate and apply security configurations in Apache
- Apply strong algorithms and secure cipher suites▪ Host admin panel at [https://secure.s\[1-80\].ierg4210.ie.cuhk.edu.hk/admin.php](https://secure.s[1-80].ierg4210.ie.cuhk.edu.hk/admin.php)
- In the .htaccess (other ways are also OK), redirect users to https website if come from [http://\[secure...\]](http://[secure...]) or [http://\[...\]/admin.php](http://[...]/admin.php)

```
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://secure.s31.ieng4210.ie.cuhk.edu.hk/$1 [R,L]
```